H2020 5Growth Project
Grant No. 856709

D2.1: Initial Design of 5G End-to-End Service Platform

Abstract

The main goal of 5Growth is to validate the ability of core 5G technologies to accommodate multiple advanced vertical services over common infrastructure and, importantly, to extend baseline 5G platforms and architecture proposals with innovations that fill the gaps towards meeting the requirements of the vertical use cases described in WP1.

This document summarizes a gap analysis over such 5G state-of-the-art baseline platforms undertaken within the project, with particular focus on 5G-TRANSFORMER, and a list of concrete innovations that are being (will be) designed and implemented during the scope of 5Growth project, which will contribute to meet the stringent requirements of 5Growth vertical use cases.

Document properties

Document number	D2.1
Document title	Initial Design of 5G End-to-End Service Platform
Document responsible	Andres Garcia Saavedra (NEC)
Document editor	Andres Garcia Saavedra (NEC)
Authors	Andres Garcia Saavedra (NEC), Xi Li (NEC), Josep Xavier Salvat (NEC), Ioannis Stavrakakis (NKUA), Nancy Alonistioti (NKUA), Sokratis Barmpounakis (NKUA), Lina Magoula (NKUA), Nikolaos Koursioumpas (NKUA), Panagiotis Kontopoulos (NKUA), Oleksii Kolodiazhnyi (MIRANTIS), Konstantin Tomakh (MIRANTIS), Denys Kucherenko (MIRANTIS), Daniel Corujo (ITAV), Diogo Gomes (ITAV), Vitor Cunha (ITAV), João Fonseca (ITAV), Claudo Casetti (POLITO), Carla Chiasserini (POLITO), Corrado Puligheddu (POLITO), Carlos Parada (ALB), Pedro Bermúdez (TELCA), Pablo Murillo (TELCA), Aitor Zabala (TELCA), Josep Mangues (CTTC), Ricardo Martínez (CTTC), Engin Zeydan (CTTC), Jordi Baranda (CTTC), Paolo Dini (CTTC), Manuel Requena (CTTC), Chrysa Papagianni (NBL), Danny De Vleeschauwer (NBL), Teresa Pepe (TEI), Paola Iovanna (TEI), Giada Landi (NXW), Juan Brenes (NXW), Foresi Bruno (TI), Andrea Sgambelluri (SSSA/CNIT), Molka Gharbaoui (SSSA/CNIT), Koteswararao Kondepu (SSSA/CNIT), Barbara Martini (SSSA/CNIT), Luca Valcarenghi (SSSA/CNIT), Jose Ordonez-Lucena (TID), Diego R. López (TID), Sonia Fernández (TID), Luis M. Contreras (TID), Carlos J. Bernardos (UC3M), Kiril Antevski (UC3M), Carlos Guimarães (UC3M)
Target dissemination level	PU
Status of the document	Final
Version	1.0

Production properties

Reviewers	Aitor Zabala (TELCA), Carlos Parada (ALB), Diego San Cristobal (ERC), Carlos Guimarães (UC3M), Carlos J. Bernardos (UC3M)

Disclaimer

This document has been produced in the context of the 5Growth Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement Nº H2020-856709.

Contents

List of Figures

List of Tables

List of Acronyms

5Gr-RL – 5Growth Resource Layer

5Gr-SO – 5Growth Service Orchestrator

5Gr-VS – 5Growth Vertical Slicer

5GT-MTP – 5G-TRANSFORMER Mobile and computing Transport Platform

5GT-SO – 5G-TRANSFORMER Service Orchestration

5GT-VS – 5G-TRANSFORMER Vertical Slicer

AD – Anomaly Detection

AI – Artificial Intelligence

BSS – Business Support System

CMD – Cyber Mimic Defense

CP – Control Plane

CSMF – Communication Service Management Function

CSP – Communications Service Provider

CU – Cloud/Central Unit

DB – Database

DHR – Dynamic Heterogeneous Redundancy

DLT – Distributed Leger Technology

DSL – Domain-Specific Language

DSS – Decision Support System

DU – Distributed Unit

E2E – End-to-End

EBI – EastBound Interface

EC – Edge Computing

EE – Execution Entity

eMBB – Enhanced Mobile Broadband

FG – Forwarding Graph

gNB – next Generation Node B

GUI – Graphical User Interface

IPO – Input Process - Output model

KPI – Key Performance Indicator

LC – Lifecycle

LCM – LC Management

LSTM – Long term Short Term Memory

MEC – Multi-access Edge Computing

ML – Machine Learning

mMTC – Massive Machine Type Communications

MILP – Mixed Integer Linear Program

MP – Monitoring Platform

MQ – Message Queue

MTD – Moving Target Defense

NBI – NorthBound Interface

NFV – Network Function Virtualization

NFVI – NFV Infrastructure

NFV-NS – NFV Network Service

NFVO – NFV Orchestrator

NS – Network Slice

NSD – Network Service Descriptor

NSI – Network Slice Instance

NSMF – Network Slice Management Function

NSO – Network Service Orchestrator

NSSMF – Network Slice Subnet Management Function

NST – Network Slice Template

OSS – Operations Support System

PA – Placement Algorithm

PoP – Point of Presence

QoS – Quality of Service

RAN – Radio Access Network

RIC – Radio Intelligent Controller

RO – Resource Orchestrator

(R)RU – (Remote) Radio Unit

RT – Real Time

SBI – SouthBound Interface

SDN – Software Defined Networking

SFC – Service Function Chaining

SLA – Service Level Agreement

SLPOC – Single Logical Point of Contact

SO – Service Orchestrator

TSP – 5G-TRANSFORMER Service Provider

TSDB – Time Series Database

UC – Use Case

UE –User Equipment

UP – User Plane

URLLC – Ultra Reliable Low Latency Communications

VA – Vertical Application

VIM – Virtual Infrastructure Manager

VM – Virtual Machine

VNF – Virtual Network Function

VNFD – VNF Descriptor

VNFM – VNF Manager

VoMS – Vertical-oriented Monitoring System

vRAN – Virtual Radio Access Network

VSB – Vertical Service Blueprint

VSD – Vertical Service Descriptor

VSI – Vertical Slice Instance

WBI – Westbound Interface

WAN – Wide Area Network

WIM – WAN Infrastructure Manager

ZDM – Zero-Defect Manufacturing

ZSM – Zero-touch Service Management

Executive Summary and Key Contributions

The main goal of 5Growth is to validate the ability of core 5G technologies to accommodate multiple advanced vertical services over common infrastructure and, importantly, to extend baseline 5G platforms and architecture proposals with innovations that fill the gaps towards meeting the requirements of the vertical use cases described in D1.1 [5].

This document summarizes the gap analysis over state-of-the-art 5G platforms, with particular focus on 5G-TRANSFORMER, and a list of concrete innovations that are being (will be) designed and implemented during the scope of 5Growth project, which will contribute to meet the stringent requirements of 5Growth vertical use cases. Specifically, the innovations selected for 5Growth are:

- **Architectural innovations:**
  - Enhanced support of verticals
    - **I1**: Radio Access Network (RAN) segments in network slices
    - **I2**: Vertical-service monitoring extensions
  - **I3**: Monitoring orchestration
  - Control and Management innovations
    - **I4**: Control-loops stability
    - **I5**: AI/ML support
  - End-to-End orchestration innovations
    - **I6**: Federation and inter-domain
    - **I7**: Next-generation Radio Access Networks
- **Algorithmic innovations**
  - **I8**: Smarter orchestration and resource control
  - **I9**: Anomaly detection
  - **I10**: Forecasting and inference
- **Framework innovations**
  - **I11**: Security and auditability
  - **I12**: 5Growth Continuous Integration/Continuous Delivery (CI/CD)

These innovations, driven by the needs of the use cases defined in WP1, are designed and will be built on top of the 5Growth reference platform, 5G-TRANSFORMER, according to a work plan organized in different tasks and appropriately scheduled during the time scope of the project. In this way, 5Growth WP2 not only does provide novel features that will help the project to perform novel trials in WP3, but it also boosts the cost-efficiency of the platform, its security and lies the foundations for vertical applications not even considered in 5Growth to deliver a future-proof platform, e.g., novel AI/ML services, next-generation virtual RANs, etc.

Accordingly, this document introduces a release plan of the platform, and reports the design status of each of the innovations which will be wrapped up around an integrated and fully-functional platform at each release. Each release will follow a process of unit tests (UTs) and a set of integration tests (ITs). Namely, 5Growth plans out two software releases:

1. Release 1 (May, 2020), including a first release of innovations I1, I2, I4 and I8;
2. Release 2 (May, 2021), with the final implementation of all innovations.

1. Introduction

WP2´s main objective is to design and build a 5G End-to-End Service Platform that is able to cope with the demanding requirements of vertical industries, particularly in the area of service automation, monitoring and interfaces with vertical customers. To this aim, 5Growth builds on top of 5G-TRANSFORMER platform, extending and enhancing its functional blocks (vertical slicer, service orchestration and resource layer) with focus on the functional and service requirements of the use cases derived in WP1, which will be demonstrated in vertical pilots within WP3.

WP2´s Task 2.1 (T2.1) is first charged with finding and selecting the set of functional and technological innovations (over 5Growth baseline platform, 5G-TRANSFORMER [1]) all across the platform (vertical slicer, service orchestrator, in addition to the resource layer). In this way, T2.1 coordinates and integrates all this research and development work to ultimately deliver a fully-functional service platform ready to be integrated into WP3´s pilots. During the first six months of the project, T2.1 has revisited the baseline platform and executed a process of exploration and selection of such innovations. **The outcome of such process of revising the 5Growth baseline platform and shedding light to the functional gaps required to meet the project objectives is presented in Section 2.**

The remaining tasks in WP2 (T2.2, T2.3 and T2.4) have contributed with ideas and gap analyses across the three main layers of the 5Growth stack (respectively, vertical slicer, service orchestrator and resource layer). As a result, the consortium as a whole has made a selection of 12 innovations that include integral improvements over the baseline platform (security, continuous integration-continuous development), architectural upgrades (monitoring platform, AI/ML platform, radio access network technologies, enhanced vertical support) and algorithmic innovations (forecasting/inference, anomaly detection, and smarter and more dynamic service orchestration and resource control). **The functional description of such key 5Growth innovations is introduced in Section 3.**

In coordination between T2.1 and the remaining tasks, the project has set up a coherent time plan for the design, testing and development of each one of the selected innovations, assigning higher priority to those most needed for integration within WP3. Specifically, 5Growth's software stack will be released in two epochs: Release 1 in month 12, and Release 2 in month 24. Some initial work has been already carried out in some of these innovations (see annex, Section 7), in addition to a general overview of the architectural vision for the rest. **The release plan and the architectural work done so far on each of the 12 innovations is shown in Section 4.**

Finally, in order to guarantee coordinated and synchronized work across all tasks, which should finally be integrated into a single platform, this document presents a glossary in an Annex (Section 8). This glossary presents concepts and nomenclature, mostly inherited by our baseline platform 5G-TRANSFORMER [1], that will be used within this document and others to come, including technical specification and software releases.

2. Baseline Platform

The 5Growth project is set out to validate core 5G technologies, by performing field trials with vertical industries, spanning from Industry 4.0 to Transportation and Energy with diverse application-specific requirements and KPIs. Building upon the vision of 5G-PPP phase 1 and phase 2 projects to support several vertical industries using a platform that supports slicing of the 5G telco infrastructure, 5Growth attempts to bridge the gap with the evolving technological innovations. The goal is to enable the uniform and automated deployment and operation of customized slices for 5Growth verticals based on the requirements from the involved stakeholders. Towards that end, 5G-TRANSFORMER architecture [1] is used as the starting point to build the 5Growth platform.[1]

In the following sections we provide an overview of the 5Growth architecture highlighting the targeted innovations, analyzing the functionality of the corresponding building blocks of the baseline 5G-TRANSFORMER architecture and identifying their functional/technological gaps towards the WP goal: the 5Growth platform.

2.1. High-level 5Growth Architecture

5Growth is set out to enhance the 5G-TRANSFORMER platform along the following dimensions: usability, flexibility, automation, performance and security, through the set of innovations proposed in Section 3. The 5G-TRANSFORMER architecture [1] is decomposed into three building blocks; the Vertical Slicer (5GT-VS), supporting the creation and management of slices for verticals; the Service Orchestrator (5GT-SO), for end-to-end service orchestration and federation of resources and services from multiple domains, and the Mobile Transport and computing Platform (5GT-MTP), acting as the underlying fronthaul and backhaul transport network infrastructure. Details on these functional components can be found in Section 2.2.

The 5Growth baseline platform rebrands the three architectural building blocks of the 5G-TRANSFORMER architecture, as depicted in Figure 1, to reflect the innovations introduced by the project, namely:

1. The 5Growth Vertical Slicer (5Gr-VS), which is inherited from 5G-TRANSFORMER Vertical Slicer (5GT-VS), acts as one-stop shop entry point for verticals to request a custom network slice. In particular the northbound interface of the 5GT-VS exposed to the vertical OSS/BSS and the functionality of the 5GT-VS will be expanded to support: (i) 5Growth innovations that will enable customized per-slice capabilities for monitoring, security and performance assurance, powered by machine-leaning and analytics; (ii) provide verticals with more control over their slices (with emphasis on the Radio Access Network (RAN)), and (iii) additional support towards the multi-domain at the service level to request services offered by different Communication Service Providers (CSPs).

[1] https://github.com/5g-transformer

2. The 5Growth Service Orchestrator (5Gr-SO), which is inherited from 5G-TRANSFORMER Service Orchestrator (5GT-SO), provides both network service and resource orchestration capabilities in order to instantiate network slices within and across multiple domains. In particular, the 5GT-SO layer will be enhanced by novel frameworks, algorithms and architectural approaches pertaining to monitoring, security and the smart orchestration, lifecycle management and control of (federated and dynamic) slices, optimizing RAN, Transport, Core and cloud/edge computing (EC) resources.
3. The 5Growth Resource Layer (5Gr-RL), which is inherited from 5G-TRANSFORMER Mobile Transport and computing Platform (5GT-MTP), hosts all the compute, storage and networking physical and virtual resources where network slices and end-to-end services are executed. The resource layer is responsible for managing the infrastructure at the vertical sites and the required transport resources to interconnect them. 5GT-MTP will be augmented to support (re-programmable) mechanisms and algorithms to support and improve security, closed-loop management and control of individual resources in the RAN, EC, transport or core domains.

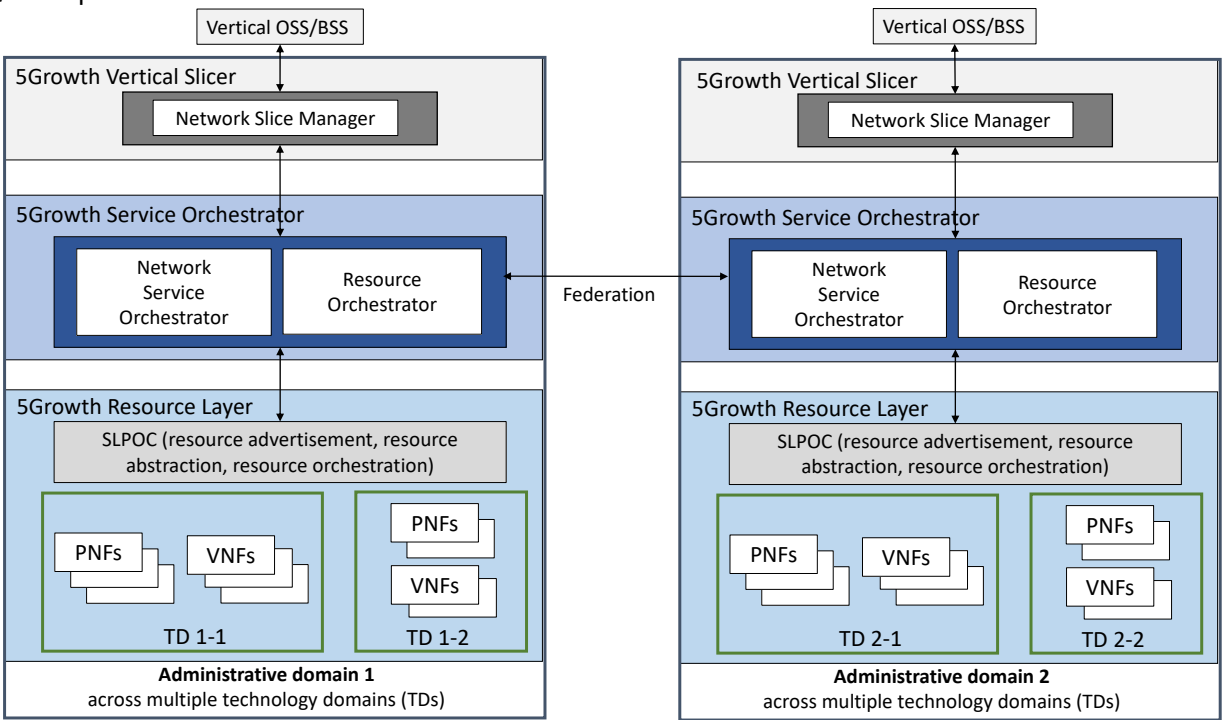

FIGURE 1: 5GROWTH ARCHITECTURE

Apart from the enhancements per each architectural building block, 5Growth will investigate and advance cross-layer interactions and dependencies that will eventually lead to more stable, performant and secure/resilient slices and thus end-to-end services for the verticals.

2.2. Baseline Architecture

We next detail the architectural design of the individual components 5Growth is based upon, namely, 5G-TRANSFORMER´s Vertical Slicer (5GT-VS), Service Orchestrator (5GT-SO) and Mobile Transport and computing Platform (5GT-MTP), and as well as the interfaces among them.

2.2.1. Vertical Slicer

The 5G-TRANSFORMER Vertical Slicer (5GT-VS) [3] operates at the northbound of the 5G-TRANSFORMER architecture and handles the requests for vertical services, issued by the verticals that constitute the consumers of the 5G-TRANSFORMER services. The Vertical Slicer manages internally the mapping and translation between the requested vertical services (Vertical Service Instances - VSI) and a number of network slices (Network Slice Instances – NSI), which are created on demand, by provisioning the associated NFV network services (NFV-NS), mediated through the 5G-TRANSFORMER Service Orchestrator (see next section for details).

At its northbound, the Vertical Slicer provides a common entry point for all the verticals to request the provisioning and management of vertical services through a simplified and vertical-oriented interface. In particular, the vertical can request a vertical service selecting initially a "template", to be used as basis for the service definition, from the catalogue of *Vertical Service Blueprints (VSB)* offered by the 5G-TRANSFORMER system and then completing their specification providing a number of service-oriented parameters that customize the desired service instance. This approach allows the verticals to focus only on the requirements, the high-level components and the logic of their service applications, without the need to specify how they must be deployed and interconnected in terms of resources and mobile connectivity. The final specification of the vertical service, provided by the vertical, is formally expressed through a *Vertical Service Descriptor (VSD)*, which is composed by the VSB filled with the user-defined parameters. The VSD comprises the input for the Vertical Slicer procedures related to the management of vertical services.

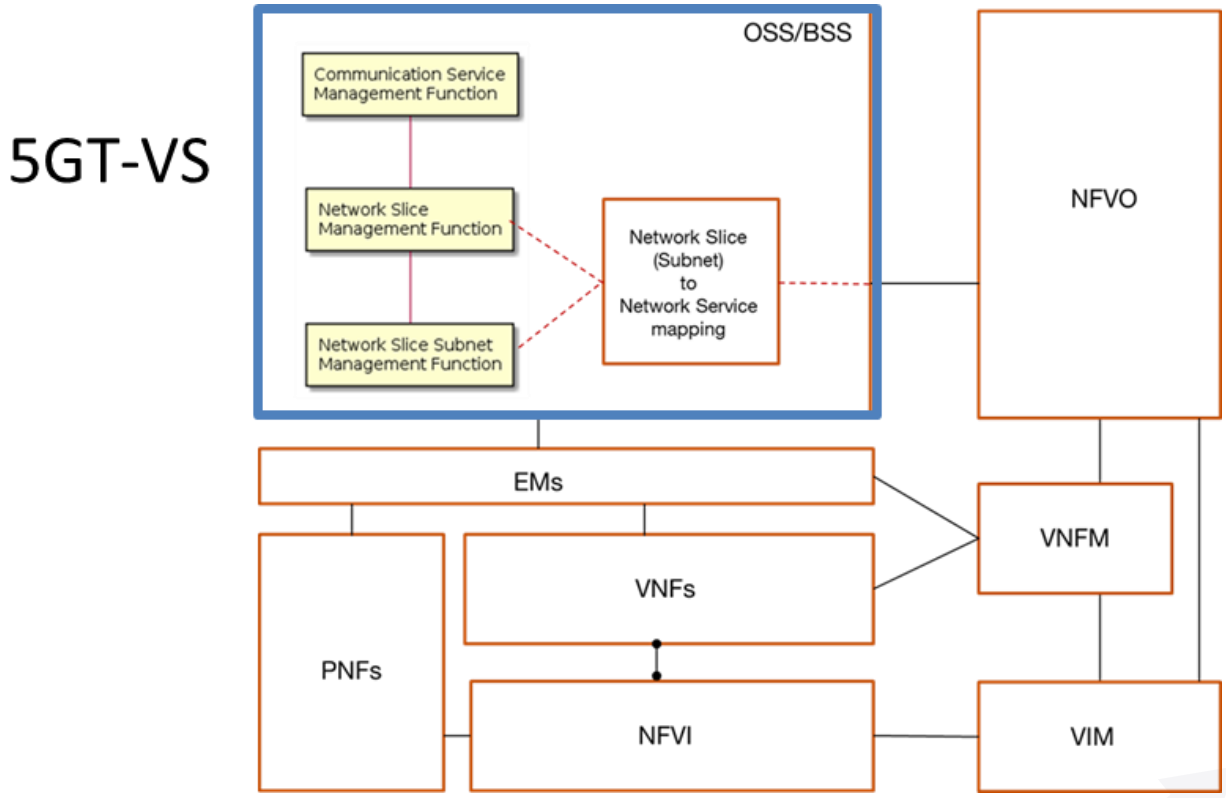

FIGURE 2: MAPPING BETWEEN 5GT-VS AND 3GPP NETWORK SLICING ARCHITECTURE

The 5GT-VS implements different functionalities identified by the network slicing architecture defined in [1], as shown in Figure 2. In particular, the Vertical Service Management Function of the Vertical Slicer (similar to the Communication Service Management Function (CSMF) defined by 3GPP) is responsible to identify the kind of network slice(s) required to provision the requested vertical service, together with their dimension. The VSD/NSD translation procedure is driven by translation rules, specified by the system administrator as part of the VSB design. These rules allow to map the service-oriented parameters specified by the verticals into the infrastructure parameters that define the dimension of the network slice (e.g. size of the VNFs, capacity of the virtual links, etc.). Moreover, the CSMF functionality is also responsible for arbitrating among multiple, concurrent services and for taking decisions about their final mapping into one or more network slices. In particular, this implies a decision about whether new network slices must be created to host the service or whether existing ones can be used and, if needed, about how they must be modified to accommodate the additional traffic and processing load. These decisions, defined as arbitration procedures, follow only business- and service-based criteria, like the vertical's SLAs, the service requirements in terms of isolation and components' sharing, or the service priority. In fact, the Vertical Slicer does not take into account any consideration related to the actual availability of network or computing/storage resources at the infrastructure level, since the resource management and logic is delegated to the lower layers of the 5G TRANSFORMER architecture.

The Vertical Slicer also implements the functionalities to actually manage the lifecycle of the network slices and their network slice subnets, in case of composed network slices. These functionalities can be mapped into the Network Slice Management Function (NSMF) and Network Slice Subnet Management Function (NSSMF) defined by 3GPP, and includes the procedures to instantiate new network slice (or subnet) instances, modify or terminate them, according to the directives received by the vertical service management logic. In this sense, the Vertical Slicer is able to handle different kinds of relationships between vertical services and network slices. In the simplest scenario, a vertical service is mapped into a single network slice that is entirely dedicated to that service and guarantees the highest level of isolation. However, in order to improve the efficiency of the service arbitration, a network slice can also be shared by multiple vertical services. Moreover, exploiting the concept of service decomposition and network slice subnets, a vertical service may be mapped into multiple network slices, where each of them provides a specific functionality (e.g. the connectivity to mobile equipment, a specific network functionality, or a service-oriented application functionality). This approach allows to share more atomic service components among different services and, consequently, share single network slice subnets among multiple "end-to-end" network slices, thus enabling the dimensioning optimization for specific portions of the slice.

Functional architecture of 5G-TRANSFORMER Vertical Slicer

The functional architecture of the Vertical Slicer is shown in Figure 3. At the northbound, the Vertical Slicer offers a Graphical User Interface (GUI) and a REST API for the verticals to request operations related to vertical services, as well as a management interface for the configuration of tenants, SLAs, VSB catalogue, etc. The southbound interface, based on REST APIs, enables the

interaction with the Service Orchestrator for the request and lifecycle management of NFV-Network Services (NFV-NS), as well as the on-boarding of the associated NSDs. In fact, in 5G-TRANSFORMER, the instantiation and management of network slices is implemented through the orchestration of a number of associated NFV-NS that provide the actual implementation of the slices at the resource level. The NFV-NS are requested dynamically to the underlying Service Orchestrator using a REST API which has been modelled following the messages and information models specified in ETSI GS NFV-IFA 013 [32] for the interaction with an NFV Orchestrator (NFVO) and in ETSI GS NFV-IFA 014 [33] for the modelling of NFV Network Service Descriptors (NSDs).

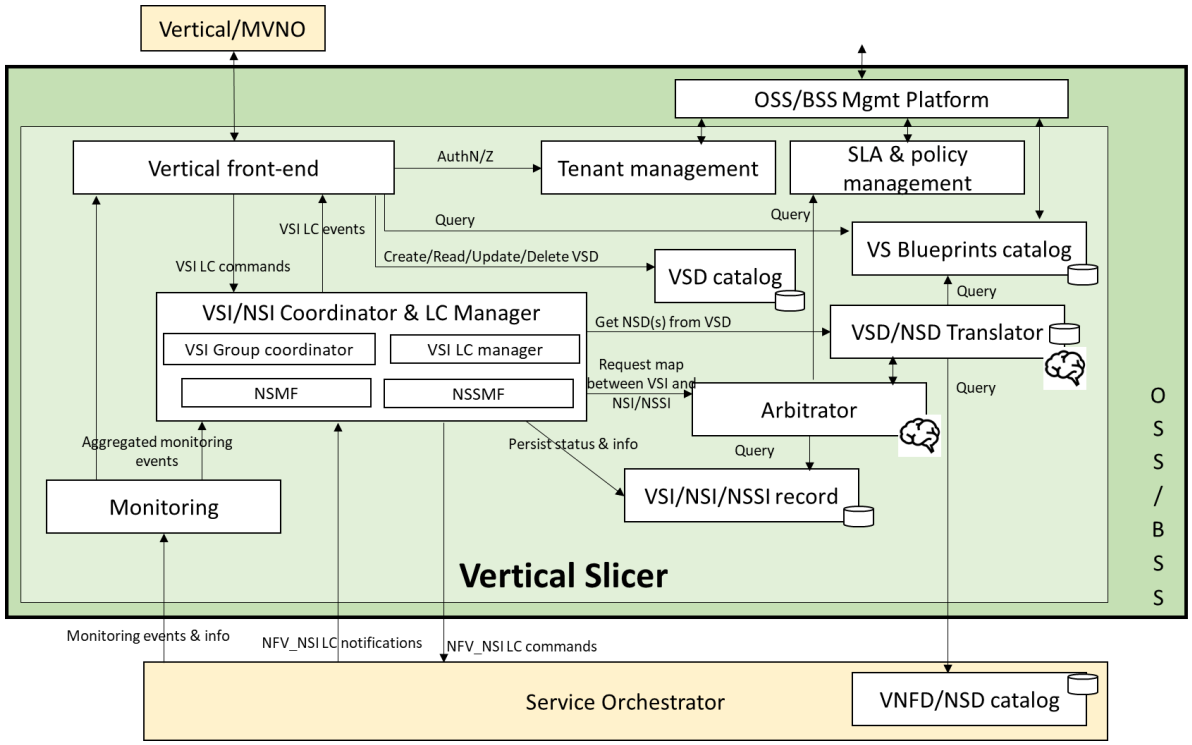

FIGURE 3: 5G TRANSFORMER VERTICAL SLICER ARCHITECTURE

The management and configuration of the Vertical Slicer system allows to configure tenants, their SLAs and the associated policies. Moreover, VS blueprint can be on-boarded in the internal catalogue. These functionalities are implemented through the "Tenant management", the "SLA & policy management" and "VS Blueprint catalog" modules and exposed to the system administrator through management REST APIs.

The Vertical Front-end of the Vertical Slicer provides the entry point for the entire system and receives requests from the verticals about the management of vertical services. The Vertical Front-end exposes the catalogue of VSBs, the APIs to customize them with open properties and to generate the corresponding VSDs and the APIs to request new vertical service instances based on such VSDs.

The lifecycle (LC) of Vertical Service Instances (VSIs) and the corresponding Network Slice Instances (NSIs) are managed through the "VSI/NSI Coordinator & LC Manager", which constitutes the engine of the Vertical Slicer and includes different logical modules dedicated to the lifecycle management of the different entities, namely, vertical services, aggregation of correlated vertical services, network slices and network slice subnets. In particular, the Vertical Slicer engine, through

the cooperation of these single modules, manages the association between VSIs and NSIs, regulates the sharing of network slices among different vertical services and the decomposition of network slices into network slice subnet instances[2]. Each module itself is specialized to handle the finite state machines of the related entity, i.e. for VSIs and NSIs/NSSIs lifecycle, coordinating operational commands, notifications and events associated to them. The network slice (subnet) management is actuated issuing requests for the instantiation, scaling and termination of the corresponding NFV-NSs, interacting with the 5GT-SO. The status of all the elements managed by the Vertical Slicer is persisted in VSI/NSI/NSSI records.

The Vertical Slicer algorithms and decision logic are implemented in the "VSD/NSD Translator" and in the Arbitrator. The VSD/NSD Translator is responsible for the mapping between vertical services and network slices, selecting the descriptors of the associated NFV network services and their required deployment size, in terms of deployment flavour and instantiation level. This approach allows to identify, e.g., the number of VNF instances, their amount of resources or the bandwidth of the virtual links interconnecting them, which characterize a network slice that is able to guarantee the service requirements specified in the VSD. The "Arbitrator", on the other hand, is the decision entity responsible to handle contention among concurrent services, based on their priority, sharing requirements and resource budget. The Arbitrator takes decisions about the number of new network slices to be instantiated, the existing ones to be re-used and how they must be modified or scaled, providing the directives towards the Vertical Slicer NSMF and NSSMF.

2.2.2. Service Orchestrator

The 5G-TRANSFORMER Service Orchestrator (5GT-SO) [4] is in charge of end-to-end (E2E) orchestration of NFV-Network Services (NFV-NS) and management of their lifecycles (including on-boarding, instantiation, update, scale, query, termination, etc.) across single or multiple administrative domains by interacting with the local 5GT-MTP and/or with other 5GT-SOs of neighbour administrative domains, respectively. This includes all tasks related with coordinating and offering to the vertical an integrated view of services and resources from the local as well as multiple administrative domains. Orchestration entails managing end-to-end services and resources, and maps them to the different administrative domains of local and/or other domains, based on service requirements and availability of the services/resources offered by each of the administrative domains.

5GT-SO receives the service requirements from 5GT-VS via its northbound interface in the shape of a Network Service Descriptors (NSD). A NSD describes a Network Service (i.e., NFV-NS) that 5GT-SO should provide (either on its own or by leveraging neighbouring 5Gr-SOs) and it is expressed in terms of chaining of VNF components (i.e., constituent VNFs) described by VNF Descriptors (VNFD). A VNFD describes a VNF in terms of its deployment and operational behaviour as well as specifies the resource and the connectivity (i.e., virtual links) requirements.

[2] A Network Slice Subnet Instance (NSSI) includes a subset of the components (Network Functions or other NSSIs) that constitute a Network Slice Instance. For a formal definition of the NSSI the reader can refer to [1].

The 5GT-SO processes the NSD in order to decide the optimum placement of the VNFs/Vertical Applications (VAs) and assign virtual networking, computing and storage resources in the local 5GT-MTP domain and/or across one or multiple other administrative domains (in case of federation) for meeting the service requirements specified in the NSD. To this purpose, the 5GT-SO embeds a series of modules, most notably, the Network Service Orchestrator (NFV-NSO) and the Resource Orchestrator (NFV-RO) with functionalities equivalent to those of the ETSI NFV Orchestrator [6]. The NFV-NSO has the responsibility of coordinating the deployment of NFV-NSs along with their lifecycle management. The NFV-RO is in charge of deciding function placement and orchestrating virtual resources under the control of the underlying 5GT-MTP(s). Additionally, the 5GT-SO provides multi-domain network service orchestration through service and/or resource federation. In this direction, the 5GT-SO interacts with 5GT-SOs of other administrative domains through its eastbound-westbound interface (federation) on the end-to-end deployment of network services and placing them in most suitable execution environment based on the offered services and/or available resources in each administrative domain.

Service orchestration involves the management and instantiation of VNFs and/or VAs (Vertical Applications) at local, edge and cloud NFVIs. The problem of mapping VNFs to (virtual) computing entities (nodes, NFVI-PoPs) and the mapping of virtual links between VNFs into (virtual) paths, depending on the granularity of abstraction offered by the 5GT-MTP, can be tackled by different optimization strategies, e.g., heuristics or mixed-integer linear programming. Moreover, automated network service management and self-configuration algorithms (e.g., NFV-NS auto-scaling) are also required to adapt service deployments to network changes and/or the infrastructure resource utilization (e.g., virtual machine CPU load) in order to prevent service degradations or SLA violations due to concurrent usage of resources. To this purpose, the collection and aggregation of monitoring data is foreseen to trigger self-adaption actions.

Summarizing, the 5GT-SO key functionalities are the following:

- Decide the optimal service (de)composition for the whole NFV-NS based on service availability as well as the resource capabilities exposed by local 5GT-MTP and by other administrative domains;
- Perform the lifecycle management of the whole NFV-NS as well as of each VNF composing the NFV-NS, including service catalogue management;
- Decide the optimal placement of VNFs/VAs[3] along with the optimal deployment of virtual links connecting VNFs through mapping operations, thereby enabling the execution of the NFV-NS or a portion of NFV-NS on the local 5GT-MTP;
- Request the needed network services to federated 5GT-SOs to address the execution of portions of the NFV-NS in other administrative domains;

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

- Perform monitoring tasks and SLA management functions to enable the triggering of self-adaptation actions (e.g., healing and scaling operations) thereby preventing service performance degradations or SLA violations.

Functional architecture of 5G TRANSFORMER Service Orchestrator

Figure 4 presents the functional architecture of 5GT-SO building blocks and their interactions designed to achieve the essential 5GT-SO operation described before. The described 5GT-SO architecture follows ETSI NFV guidelines [7] and is in line with orchestration system designs developed in related EU (European Union) projects (i.e., 5GEx [8] and 5G-Crosshaul [9]).

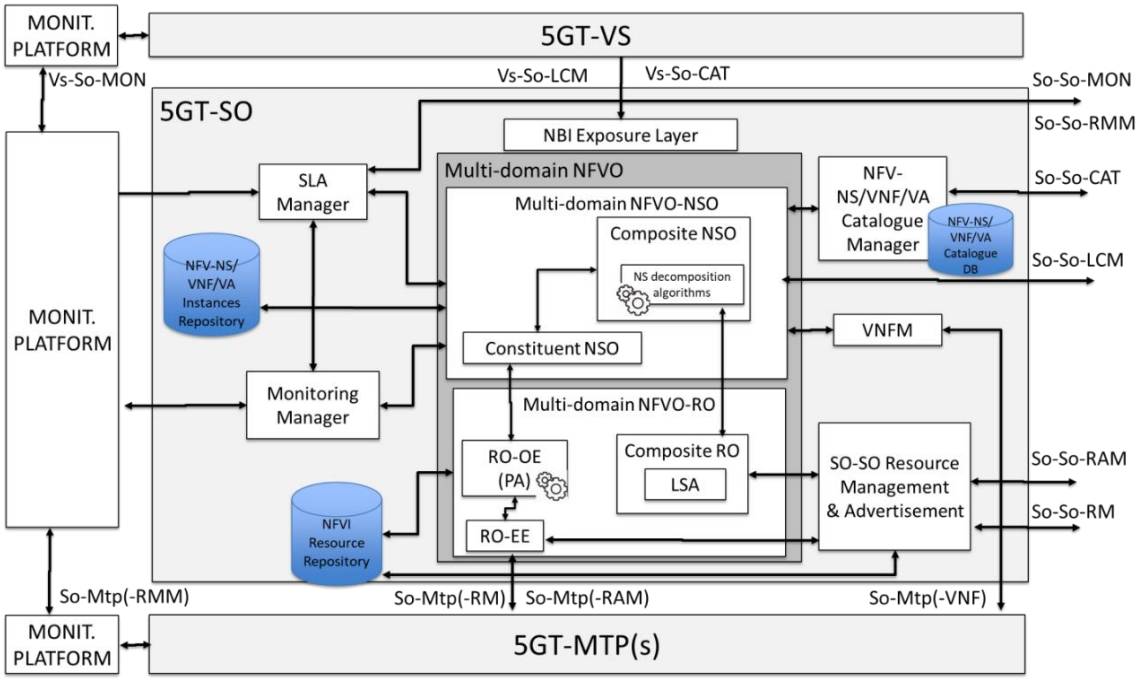

FIGURE 4: 5GT-SO FUNCTIONAL ARCHITECTURE

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 instantiation, service modification (e.g., scaling), and service termination.
- **NFV-NS/VNF/VA Catalogue DB/Manager:** Catalogue DB is the repository of all usable NFV Network Service (NFV-NS), Virtual Network Function (VNF) and Vertical Application (VA) descriptors that can be accessed through the Catalogue Manager. Such descriptors are used by the 5GT-SO in the process of NFV-NS/VNF/VA instantiation and its lifecycle management to obtain relevant information, e.g., deployment flavours. The Catalogue Manager also contains the aggregated information on the available NFV-NSs/VNFs/VAs offered to the external/federated domains for federation. The aggregated information is stored by the Composite Network Service Orchestration (NSO).
- **Multi-Domain NFV Orchestrator (NFVO):** NFVO has the responsibility of orchestrating virtual resources across multiple domains, fulfilling Resource Orchestration (NFVO-RO)

functions, as well as of coordinating the deployment of NFV-NSs and their lifecycle management, i.e., Network Service Orchestration (NFVO-NSO) functions. More specifically:

- **Multi-Domain NFVO-NSO** coordinates all the NFV-NS deployment operations as well as formal checks of service requests based on attributes retrieved from NSDs and VNFDs. In particular, the Composite NSO, using the NS decomposition algorithms (which are designed to transform/decompose a single NFV-NS into several components or several nested NFV-NSs) decomposes the NSDs into several nested NFV-NSs or does not decompose the NFV-NS (keeping it as a single compact NFV-NS). Note that this is the case when the received NSD is not explicitly a composite NSD. In case of explicit composite NSD, the decomposition is executed by default as it is configured in the NSD. Then the Composite NSO decides where to deploy each nested NFV-NS either locally or in a federated domain, i.e., whether using a local 5GT-MTP or leveraging neighbour 5GT-SOs. Accordingly, the Composite NSO requests (i) the Constituent NSO and then the local NFVO-RO to deploy a local nested NFV-NS into its local administrative domain; and/or (ii) the federated NFVO-NSO to deploy a nested NFV-NS into other federated administrative domains. In the latter case, the Composite Resource Orchestration (Composite RO) within the Multi-Domain NFVO-RO, which includes a LSA (Link Selection Algorithm/Entity), is in charge of selecting and triggering the creation of the links/paths between the deployed nested NFV-NSs either locally or across the federated 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 network services according to their associated VNF forwarding graphs.
- **Multi-Domain NFVO-RO** maps the nested NFV-NS into a set of virtual resources through the RO Orchestration Engine (**RO-OE**) by deciding the placement of each VNF in the virtual infrastructure, based on specified computing, storage and networking (e.g., bandwidth) requirements. The decision is made by the RO-OE leveraging Placement Algorithms (PAs) [4][46][48] based on heuristics that optimize the placement of VNFs/VAs pursuing objectives like minimizing latency and overall cost. On the other hand, it is based on the available virtual resources that are exposed by the 5GT-MTP via the So-Mtp Southbound Interface (SBI) or from other domains via the So-So/East-Westbound Interface (EBI/WBI) and stored into the NFVI Resource Repository. In the latter case, sharing abstract views is needed to build up a comprehensive view of resources available from different domains, which is carried out by the SO-SO Resource Management & Advertisement element. Then, the RO Execution Entity (**RO-EE**) takes care of resource provisioning by coordinating correlated actions to execute/forward the allocation/release requests to either local 5GT-MTP or to the 5GT-SO NFVO-RO of other domains, in the latter case leveraging the SO-SO Resource Advertisement & Management building block.

- **VNF Manager (VNFM):** the VNFM is in charge of the lifecycle management of the VNFs deployed by the 5GT-SO. Its functionality is in accordance with the generic VNFM function

defined by ETSI NFV, integrated in MANO platforms (e.g., Cloudify, OSM). 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 Management & Advertisement:** in resource federation, this block is in charge of exchanging abstract resource views (e.g., abstract topologies, computing and storage capabilities) with other domains consolidating inputs and storing federated resources into the NFVI Resource Repository. Either static or dynamic approach can be used to this purpose, e.g., static views with topological information only or dynamic views with also operational information (like current virtual resource load) that is updated periodically. This block is also in charge of issuing virtual resource allocation/release requests commands to/from federated domains that are triggered by the RO-EE within a resource federation process. For service federation, it simply acts as a proxy for the inter-connection of Composite RO, to exchange resource-related information for inter-domain inter-nested path setup (i.e., set-up of paths between two nested NFV-NSs that are deployed in different domains).
- **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 Management & Advertisement block in case of abstract resource views received from other 5GT-SOs/domains through the So-So/East-Westbound Interface (EBI/WBI).
- **NS/VNF/VA Instance Repository**: This repository stores the instances of VNFs, NFV-NSs, and VAs that have previously been instantiated.
- **SO Monitoring Manager**: This block is the one responsible for translating the high-level, service-centric monitoring requirements of instantiated NFV-NSs into low-level, resource-centric jobs to be activated within the Monitoring Platform[4] to retrieve the necessary information from the virtual infrastructure elements (e.g., "CPU usage of caches" requires the "average CPU usage" of every cache VM). After activating the required jobs, this block is also in charge of configuring and managing the monitoring jobs lifecycle by interacting with the Monitoring Platform. It makes the details of these jobs available to the SLA Manager, enabling the configuration of threshold values for notifications. Finally, it is also notified whenever a service is scaled or terminated in order to update or remove the monitoring jobs related to the service.
- **SLA Manager:** This block assures that the agreed SLAs between the 5GT-VS and the 5GT-SO on the network service deployed are continuously satisfied through on-line SLA verification. To this purpose, this block leverages information about trends of relevant monitoring data that may indicate potential SLA breaches or anomalous behaviours that may lead to system under-performance. For this reason, the SLA Manager interacts with the Monitoring Platform following a subscription-notification paradigm, in order to promptly react to any alert associated to the target monitoring data. In particular, the SLA Manager

[4] The 5G-TRANSFORMER Monitoring Platform will be introduced in Section 3.2.2.

subscribes with the Monitoring Platform (Config Manager), registering monitoring parameter queries (which may include arbitrarily complex expressions on many different time series) and setting threshold values for notifications, such that whenever the given threshold is passed, a notification is sent to the SLA Manager, which will then trigger the needed reactions at the 5GT-SO (e.g., scaling).

2.2.3. Resource Layer

The 5GT-MTP [2] is responsible for the orchestration of resources and the instantiation of VNFs over the infrastructure under its control, as well as for managing the underlying physical mobile transport network, computing and storage infrastructure. It manages all the infrastructures as a Single Logic Point of Contact (SLPOC), providing a common abstraction view of the managed resources to the Service Orchestrator via the IFA005 interface [25].

The SLPOC is connected to different plugins: the transport and radio Wide Infrastructure Manager (WIM) plugins, the Virtual Infrastructure Manager (VIM) plugin, and an ETSI-compliant Multi-Access Edge Computing (MEC) plugin. These plugins expose different resources view to the SLPOC via the IFA005 interface.

Functional architecture of 5G TRANSFORMER MTP

The functional architecture of the 5GT-MTP architecture is reported in Figure 5.

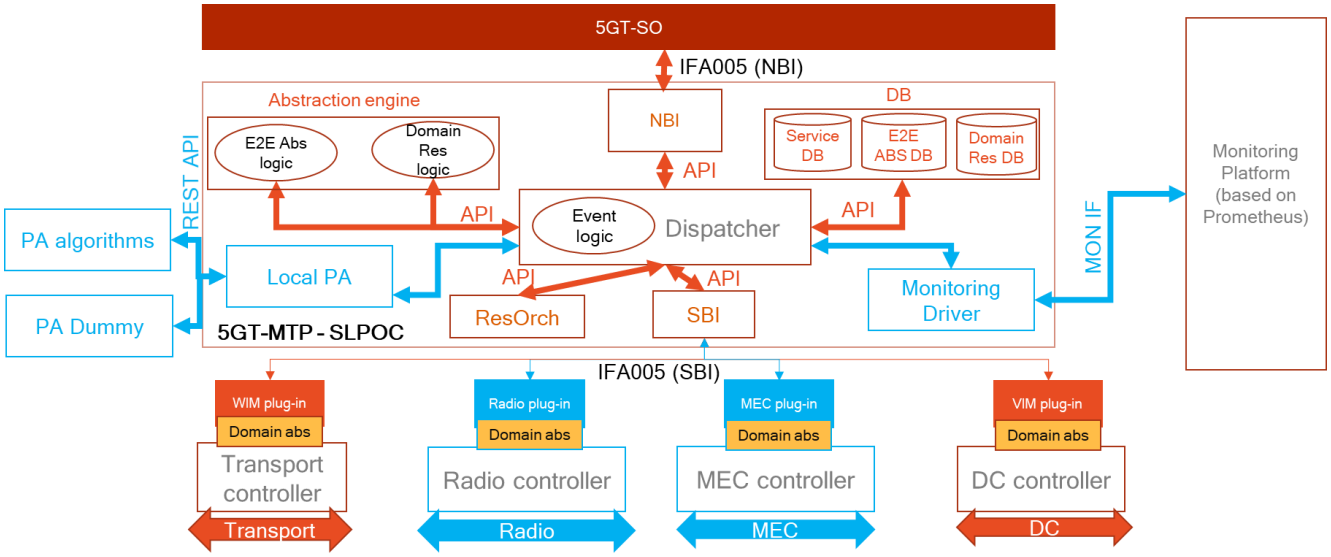

FIGURE 5: 5G-TRANSFORMER MTP ARCHITECTURE

The MTP consists in the following components:

- ***Abstraction Engine***: this module implements all the algorithms and procedures related to the abstraction. The scope of the module is to provide an abstracted view of the available resources (e.g., transport (WIM), data-centers (VIM), RAN and Multi-access Edge Computing (MEC)) to the Service Orchestrator. In particular, the "Domain Resource Logic" performs the aggregation of the domain resources, while the "E2E Abstraction Logic" performs the computation of the abstracted view in terms of service parameters

(e.g., bandwidth, latency) that can be guaranteed with the available resources. In the service allocation process, such modules are also responsible to select the optimal domain resources to allocate for the required service.

- ***Database (DB)***: it is a common module that stores all information and details regarding the domain resource including RAN and MEC resources, the abstraction view and the allocated services. It is a front-end that connects to an external SQL server and handles all the queries (i.e., to insert entries, update data and retrieve parameters) towards the database to store and retrieve abstraction, aggregation and service information.
- ***Dispatcher***: it is an Event bus that manages the inter-process communication between all the 5GT-MTP components (i.e., NBI, SBI, DB, RO, Abstraction Engine, monitoring driver and Local Placement Algorithm) using a publish-subscribe pattern. It allows all MTP modules to post events that are handled by specific handlers.
- ***Resource Orchestrator (ResOrch)***: it orchestrates the 5GT-MTP operations between the VIM/Radio and Transport WIM/MEC domains.
- ***Local PA:*** it is a module that provides the selection and optimization of resources thanks to dedicated and specialized resource Placement Algorithm (PA). It handles the communication with an external PA module by means of a new REST API. The aim of local PA module is two-fold. First, the local PA module contacts an external PA module in order to get decision about the specific VIM where to put a VNF in a multi-VIM domain. Second, the local PA module may contact an external PA module to compute the interNfviPop connectivity for a logical link between a pair of NFVI-PoP GWs with specific network constraints.
- ***Monitoring Driver:*** it manages the communication with the monitoring platform of the 5G-TRANSFORMER architecture (based on Prometheus[5]). It uses the monitoring interface exposed by the Monitoring Platform to register itself to the platform, create performance monitoring jobs and get alert notifications. Moreover, it handles notifications about the status of infrastructure resources (i.e., compute, storage, networking, radio, and MEC resources) by posting specific events to the dispatcher.
- ***North-Bound Interface (NBI)***: it handles the IFA005 [25] communication with 5GT-SO.
- ***South-Bound Interface (SBI)***: it handles the IFA005 [25] communication with the plugins.

2.2.4. Interfaces

This section explains the reference points and interfaces that 5G-TRANSFORMER defined as part of its architecture [1], which are the basis for those to be defined in the 5Growth architecture.

[5] https://prometheus.io/

2.2.4.1. Vertical – Vertical Slicer

The 5GT-VS provides the interface of the 5G-TRANSFORMER system to the customers as well as to the 5G-TRANSFORMER service provider (TSP) to manage the service offerings (Figure 6). Therefore, the northbound interface (NBI) of the 5GT-VS is as well the NBI of the 5G-TRANSFORMER system. Two reference points are defined at the northbound of the 5GT-VS (see [3]):

- **Ve-Vs**, between a vertical and the 5GT-VS. This reference point provides the mechanisms to allow the vertical to retrieve vertical service blueprints (VSBs), to manage Vertical Service Descriptors (VSD), to request operational actions on vertical service instances (VSI), like instantiation, termination, modification, and to monitor performance and failures of instantiated vertical services.
- **Mgt-Vs**, between the TSP's OSS/BSS Management Platform and the 5GT-VS. This reference point provides primitives to manage tenants, SLAs and VSBs. It is used mainly for management and administrative issues and it is handled internally within the 5G-TRANSFORMER service provider.

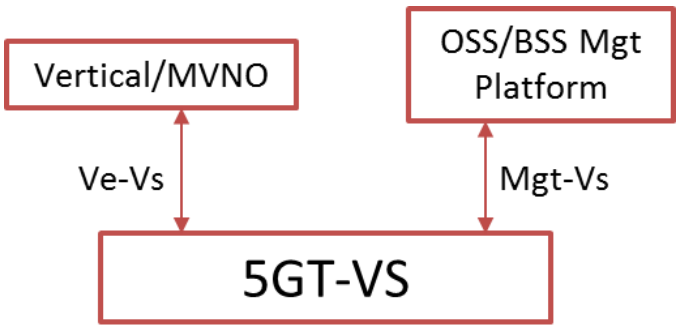

FIGURE 6: REFERENCE POINTS ON THE NORTHBOUND OF THE 5GT-VS

The 5GT-VS NBI implementing both reference points is a REST API based on HTTP/JSON messages, where 5GT-VS acts as REST server and verticals and the OSS/BSS Management Platform act as REST clients. For a full specification of the primitives and their abstract messages, please refer to documentation in 5G-TRANSFORMER D3.3 [3]. Protocol messages encoding can be found in the 5G-TRANSFORMER public Github[6].

The 5GT-VS NBI specified in this section focuses on the provisioning of Vertical Services. For Network Slice as a Service (NSaaS) it is considered network slices of 5G access and core networks. Management of these slices can be done by the vertical or an MVNO through a dedicated management service access point. For NFVIaaS, 5G core networks were considered, where specific functionality such as subscriber management is provided by the MVNO. These NSaaS and NFVIaaS can be described by vertical service blueprints and descriptors and handled within the 5GT-VS as described below.

The Ve-Vs reference point implements the following operations:

[6] https://github.com/5g-transformer

- Query VSBs.
- Create, query, update, and delete VSDs.
- Instantiate, query, terminate and modify Vertical Service Instances (VSI).
- Trigger notifications about vertical service lifecycle events.
- Query monitoring parameters for VSIs.
- Subscriptions/notifications about vertical service monitoring parameters.
- Trigger notifications about vertical service failures.

As an example, we provide the definition of the Query VS blueprints operation. The full list of operations of this reference point are described in D3.3 [3]. The Query VS blueprints operation allows a vertical to retrieve one or more VSBs from the 5GT-VS catalogue. The blueprints are then used by the vertical to create the VSDs for the vertical services to be instantiated.

The Query VS blueprints messages are specified in Table 1.

TABLE 1: QUERY VS BLUEPRINTS MESSAGES

Message	Direction	Description	Parameters
Query VS blueprint request	Vertical → 5GT-VS	Request to retrieve one or more VSBs matching the given filter.	• Filter (e.g. VSB ID, …) Vertical ID.
Query VS blueprint response	5GT-VS → Vertical	Response including the details of the requested VSBs.	• List<VSB>

The Mgt-Vs reference point between the OSS/BSS Management Platform (Mgt in the following) and the 5GT-VS implements the following operations:

- Create, query and delete tenants.
- Create, query, modify and delete SLAs.
- Create, query and delete VSBs.

Beyond these operations, the OSS/BSS Management Platform has also access in read mode to the information related to all the entities managed by the 5GT-VS, i.e., VSDs, VSIs, NSIs and NSSIs, which can be retrieved from the related catalogues and records.

2.2.4.2. Vertical Slicer – Service Orchestrator

The 5GT-SO NBI refers to the interface between the 5GT-VS and the 5GT-SO, based on the ETSI NFV IFA 013 interface (reference point *Os-Ma-nfvo* between the OSS/BSS and the NFVO in the NFV MANO architecture) [32]. This interface is used for:

- Network service lifecycle management and forwarding of information related to the status of the NFV network services;
- Management of NFV NS Descriptors, VNF packages and PNF Descriptors (PNFDs);
- Monitoring of Network Service Instances (NFV-NSI); and
- Policy management.

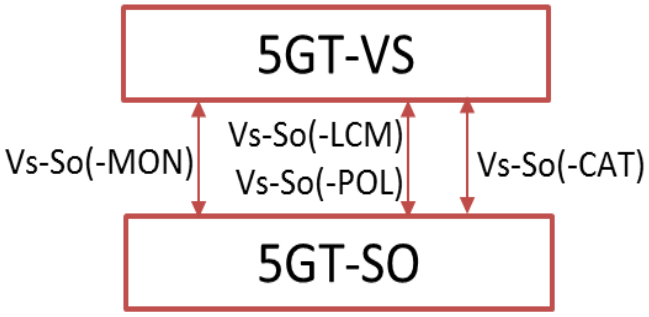

FIGURE 7: REFERENCE POINTS BETWEEN 5GT-VS AND 5GT-SO

The interactions between the 5GT-VS and 5GT-SO comprise the four reference points shown in Figure 7 and described below:

- **Vs-So (LCM – LifeCycle Management):** used to manage the lifecycle of NFV-NSs. It offers primitives to instantiate, terminate, query, and reconfigure NFV network service instances or receive notifications about their lifecycle.
- **Vs-So (MON – MONitoring):** used to monitor NFV-NS and VNF instances through queries or subscriptions/notifications about performance metrics. Additionally, this interface provides APIs for fault management.
- **Vs-So (CAT – CATalogue):** used to manage of NFV NSD, PNFDs, VNF and MEC Application package descriptors, namely on-boarding, removal, updates and queries.
- **Vs-So (POL – POLicy):** used to manage policies. It offers primitives to transfer, delete, activate, deactivate, associate, disassociate policies and receive notifications about policy conflicts.

The mapping between the four Vs-So reference points and the specific interfaces defined in the ETSI NFV IFA 013 is reported in deliverable D3.3 [3] together with the required extensions in terms of descriptors' information model (e.g. in support of MEC applications integrated in NFV-NSs) and interfaces' information elements (e.g. in support of geographical or latency constraints specification).

2.2.4.3. Service Orchestrator – Service Orchestrator

The 5GT-SO provides the interface of the 5G-TRANSFORMER system to another external 5G-TRANSFORMER system. Therefore, the eastbound/westbound interface (EBI/WBI) of the 5GT-SO is as well the EBI/WBI of the 5G-TRANSFORMER system. Six reference points are defined at the EBI/WBI of the 5GT-SO (see Figure 8):

- **So-So** (**L**ife **C**ycle **M**anagement), between consumer 5GT-SO NFVO-NSO and provider 5GT-SO NFVO-NSO. This reference point provides the mechanisms to instantiate, terminate, query or re-configure nested NFV-NS or receive notifications for federated nested NFV-NS.
- **So-So** (**MON**itoring)**:** between consumer 5GT-SO NFVO-NSO and provider 5GT-SO NFVO-NSO. This reference point provides monitoring of nested NFV-NS through

queries or subscription/notification of performance metrics, VNF indicators and NFV-NS failures.

- **So-So (Cat**alogue**):** between consumer 5GT-SO NFVO-NSO and provider 5GT-SO NFVO-NSO. This reference point provides primitives to subscribe/notify for Catalogue changes, queries of descriptors (NSDs and AppDs) and packages (VNF and MEC Application Packages).
- **So-So (R**esource **M**anagement**):** between consumer 5GT-SO NFVO-RO and provider 5GT-SO NFVO-RO. This reference point provides operations for configuration of resources, configuration of network paths for connectivity among VNFs/VMs. These operations mainly depend on the level of abstraction applied to the actual resources.
- **So-So (R**esource **M**onitoring **M**anagement**):** between consumer 5GT-SO NFVO-RO and provider 5GT-SO NFVO-RO. This reference point provides monitoring of different resources, computing power, network bandwidth, latency, storage capacity, VMs, or MEC hosts, provided by the peering administrative domain. The detail level depends on the agreed abstraction level.
- **So-So (R**esource **A**dvertising **M**anagement**):** between consumer SO-SO Resource Advertisement and provider SO-SO Resource Advertisement. This reference point provides the mechanisms for advertising available resource abstractions to/from other 5GT-SOs. Periodic or event-triggered updates for near real-time availability of resources are considered.

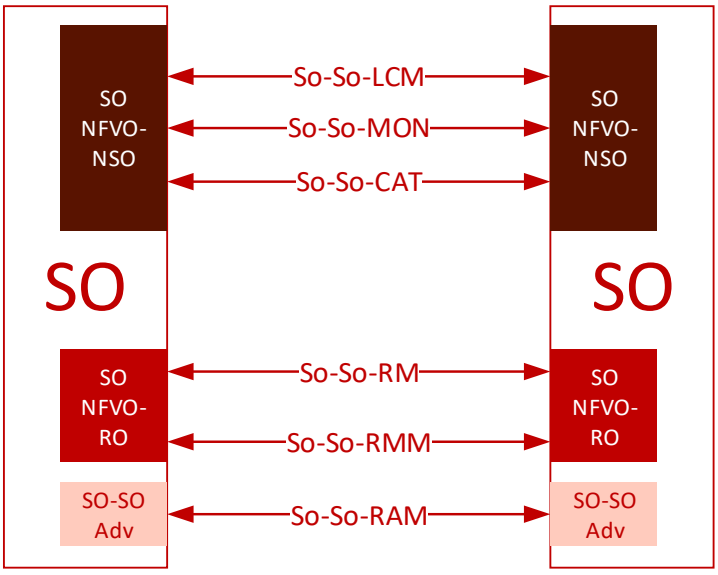

FIGURE 8: REFERENCE POINTS FOR 5GT-SO EBI/WBI (I.E.,SO-SO INTERFACE)

The 5GT-SO EBI/WBI implementing all reference points is a REST API based on HTTP/JSON messages, where the provider 5GT-SO acts as REST server and the consumer 5GT-SO acts as REST client. This section focuses only on the main functionalities supported at each reference points, while the full specification of the primitives and their abstract messages is documented in D4.3 [4]. The 5GT-SO EBI/WBI specified in this section focuses on the provisioning of the Network Service as

a Service (NFV-NSaaS) and the NFVI as a Service (NFVIaaS) cases. The reference points on the EBI/WBI are based on operations defined in the ETSI NFV IFA documents.

The **So-So-LCM** reference point implements the following operations specified in IFA 013 [32]:

- Instantiate, query, terminate, modify nested NFV-NSs.
- Subscribe/Notify about nested NFV-NSs lifecycle change notification.

The **So-So-MON** reference point implements the following operations specified in IFA 013 [32]:

- Subscribe/notify, query about nested NFV-NSs performance information.
- Create, delete, query nested NFV-NSs threshold operation.
- Subscribe/notify about nested NFV-NSs fault alarms.

The **So-So-CAT** reference point implements the following operations specified in IFA 013 [32]:

- Query, update, subscribe/notify changes on NSDs.
- Query, update, subscribe/notify changes of MEC Application Package

The **So-So-RM** reference point implements the following operations specified in IFA 005/006/008 [25][26][28]:

- Add, query, update, delete software image operations.
- Allocate, query, update, terminate, operate on compute/network/storage resources. Due to abstraction of the resources, operations and information are limited to the abstracted level.
- Create, instantiate, scale, scale to level, change, terminate, delete, query, operate, modify, get operation status of VNFs. Due to abstraction of the resources, operations and information are limited to the abstracted level.

The **So-So-RMM** reference point implements the following operations specified in IFA 005/006/008 [25][26][28]:

- Query, subscribe/notify, create/delete/query threshold operation for performance information on compute/network/storage resources. Due to abstraction of resources, operations and information are limited to the abstracted level.
- Subscribe/notify for lifecycle changes of VNFs.
- Subscribe/notify, query, create/delete/query threshold performance operation of VNFs/VMs.
- Subscribe/notify fault alarms of VNFs/VMs.

The **So-So-RAM** reference point implements the following operations specified in IFA 005/006 [25][26]:

- Query quota information for compute/network/storage resources. Due to abstraction of resources, information is limited to the abstract level.
- Subscribe/notify for change of compute/network/storage resources. Due to abstraction of resources, information is limited to the abstract level.

2.2.4.4. Service Orchestrator – Resource Layer

The So-Mtp Interface (5GT-SO southbound interface (SBI) as seen by the 5GT-SO and 5GT-MTP northbound interface (NBI) as seen by the 5GT-MTP) addresses the interworking between the 5GT-SO and the 5GT-MTP building blocks of the 5G-TRANSFORMER architecture. A single 5GT-SO interacts with a single 5GT-MTP via the defined So-Mtp interface. In a nutshell, such an interface allows eventually handling the configuration and programmability of a number of resource domains (including virtualized resources for compute, storage and networking) being coordinated by the underlying 5GT-MTP. Besides managing the utilization (i.e., de/allocation) of the virtualized resources, the So-Mtp interface provides the required functionalities for deploying (updating and terminating) demanded VNFs by a given NFV-NS.

As depicted in Figure 9, the So-Mtp reference point includes multiple interfaces providing different functions. In this regard, the So-Mtp reference point has been designed to effectively support within a single and common API multiple operations and functionalities derived from the required interworking and communication between 5GT-SO elements and the 5GT-MTP entity.

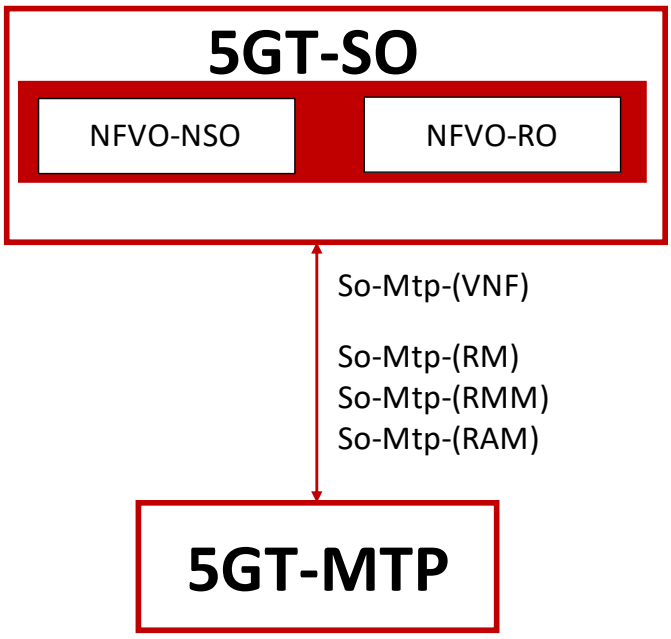

FIGURE 9: REFERENCE POINTS FOR 5GT-SO SBI (I.E., SO-MTP INTERFACE)

Formerly, the design of the So-Mtp reference point was conceived to be based on (or leverage at most) existing set of standard documents being produced within the ETSI NFV framework, namely ETSI GS NFV-IFA005 [6], ETSI GS NFV-IFA006 [26] and ETSI GS NFV-IFA008 [28]. However, it was realized that specific functions and requirements exclusively in the scope of handling virtualized network resources between NFVI-PoPs (i.e., within the WAN interconnecting remote NFVI-PoPs) were not fully supported by these standard documents. In this context, some efforts were devoted to better cope with the specificities to manage network resources in the WAN by means of a WIM controller. Documents such as ETSI GR NFV-IFA022 [34] provided some ideas in this respect. Regardless of the considered implementation, the following high-level operations and functions

have been defined and deployed to be supported by the So-Mtp interface in terms of a set of messages and workflows:

- Providing abstracted information (e.g., capacities, availability, connectivity, etc.) of the virtualized resources managed by each 5GT-MTP. This entails both internal NFVI-PoP resources (e.g., aggregated amount of available CPU or memory, etc.) as well as inter-NFVI-PoP network resources. For the latter, networking links interconnecting a given pair of NFVI-PoPs are referred to as logical links. For each derived logical link handled in the 5GT-SO and exposed by the 5GT-MTP specific attributes such as the available bandwidth and total delay are provided. Notice that this information is necessary in the Placement Algorithm execution at the 5GT-SO. Carried out through the SO-Mtp(-RAM) reference point.
- Managing (i.e., instantiation, allocation, scaling up/down and release) of the virtualized resources required to support both 5GT-SO operations. For instance, this entails allocating network resources in the logical links connecting a pair of NFVI-PoPs, or allocating compute resources within a NFVI-PoP that will eventually host a specific VNF. Carried out through the SO-Mtp(-RM) reference point.
- Supporting the lifecycle management (i.e., creation, configuration, modification and termination) of the VNFs to be placed in selected NFVI-PoPs. Carried out through the SO-Mtp(-VNF) reference point.
- Supporting fault management and performance monitoring jobs that involves the resources and infrastructures allocated in VIM, WIM, Radio domains for a service. Carried out through the SO-Mtp(-RMM) reference point.

According to the previous operations to be covered by the So-Mtp interface, these are divided into the following subset of interfaces: **So-Mtp(-RAM)** provides the Resource Advertisement Management functions; **So-Mtp(-RM)** encompasses the Resource Management operations over the virtualized resources; **So-Mtp(-RMM)** provides the Resource Management and Monitoring operations; **So-Mtp(-VNF)** takes over the general VNF lifecycle management (e.g., scaling up/down a particular VNF instance, fixing VNF malfunctions, etc.). For more details, please, refer to D2.3 [2].

In light of the above defined pool of functionalities and operations to be supported over the So-Mtp interface, its implementation (i.e., API) was designed to leverage on:

1. Selected messages compliant with the current standard ETSI IFA 005 specification. Such messages support the operations for intra-NFVI-PoP virtualized resource management (e.g., allocating / terminating virtualized compute resources, instantiating VNFs, etc.).
2. The use of a proprietary 5GT-designed set of messages for handling the interactions related to inter-NFVI-PoP operations that are better tailored to the intricacies and features of the 5GT solution (e.g., the management and abstraction used by the 5GT-SO in terms of logical links between a pair of NFVI-PoPs' gateways). It is worth stating that the design of those inter-NFVI-PoP-oriented messages was done inspired and reusing (as much as possible) some information elements actually supported in the ETSI IFA 005.

2.3. Gap Analysis

5G-TRANSFORMER is indeed a suitable baseline platform for 5Growth. However, both integral and particular enhancements and upgrades may be designed and implemented to contribute to meeting the demanding requirements of the use cases defined in D1.1 [5].

Specifically, 5Growth consortium has identified a set of features that are not supported by our baseline platform and are therefore object of research and focus during the scope of 5Growth project, namely:

- Enhanced Vertical Service (VS) slice sharing: Re-use/combine already running nested-NSs to build different types of E2E NFV-NS, while improving the criteria driving sharing decisions according to cross-layer predictions (e.g., exploiting AI/ML algorithms);
- VS arbitration at runtime: Dynamically modify the size and the composition of network slices according to arbitration decisions triggered by changes in already running services, e.g., due to scaling actions actuated at the SO level;
- VS layer federation and multi-domain: VS layer federation procedures, i.e., at a higher level than service or resource federation supported by 5G-TRANSFORMER;
- VS dynamic service composition: Build from scratch a new (composite) NFV-NS based on an on-boarded VSB. Currently, 5GT-VS needs that the NSD is on-boarded offline;
- Service Orchestration automatic network service management: Zero-touch management of live NFV-NSs (e.g., SO service/VNF migration, federation, scaling up/out/down/in, etc.);
- SO self-adaptation actions: Automated adaptation to dynamic changes on the underlying infrastructure resources;
- SO dynamic monitoring orchestration: Dynamically deploy monitoring probes and configure them (e.g., set range of thresholds);
- SO geolocation dependent federation: Allow choosing the federation provider domain with specific geo-footprint requirements;
- Resource Layer PNF integration: Seamless integration of PNF resources as part of the underlying infrastructure;
- RL geo-specific resources: Allow the instantiation of NFV-NSs over computing/networking resources placed on a specific geo-location area;
- Radio Access Network (RAN) support: Seamless integration and mapping of RAN resources and services;
- Integral security: Secure communication between modules across all platform layers;
- Continuous development and integration: Simplify accessibility and usability of the platform for verticals, developers and operators.

To summarize such analysis, Table 2 maps the abovementioned features, which are currently not supported by the baseline platform, and which 5Growth use cases (defined in D1.1 [5]) may benefit from them.

TABLE 2: MAPPING BETWEEN FEATURES NOT CURRENTLY SUPPORTED BY OUR BASELINE PLATFORM AND 5GROWTH USE CASE REQUIREMENTS

Features (gaps)	Pilot 1		Pilot 2			Pilot 3		Pilot 4	
	UC1	UC2	UC1	UC2	UC3	UC1	UC2	UC1	UC2
Enhanced VS network slice sharing			X	X	X			X	X
VS arbitration at runtime			X	X	X				
VS layer federation		X				X	X	X	
VS dynamic service composition			X			X	X	X	
SO automatic network service management	X							X	X
SO self-adaptation actions	X	X	X	X	X	X	X	X	X
SO dynamic monitoring orchestration	X					X			X
SO geo-location dependent federation	X		X	X	X		X	X	X
RL PNF integration	X				X	X	X	X	X
RL geo-specific resources			X		X		X	X	X
RAN support	X	X	X	X	X	X	X		X
Integral security	X	X	X	X	X	X	X	X	X
Continuous development and integration	X	X	X	X	X		X	X	X

3. 5Growth Innovations

This section summarizes the set of functional and technology innovations selected for design and/or implementation within the 5Growth platform. We first present an overview of the selected innovations in Section 3.1, clustered into three categories: architecture, algorithms and framework innovations; and then we present a more detailed functional and technical description in Sections 3.2 (architecture innovations), 3.3 (algorithm innovations) and 3.4 (framework innovations).

3.1. Selection of innovations

The analysis performed in Section 2 has fostered an intense phase of exploration for innovations to be designed and implemented during the scope of 5Growth project; the goal being to enhance and upgrade the capabilities of our baseline platform such that WP2 delivers a platform that contributes consistently to satisfy the demanding requirements of the project use cases.

Overall, 12 innovations, summarized in Table 3, have been selected. For better organization of the work and presentation of the innovations, they have been grouped into three main innovation clusters: Architecture, Algorithms and Framework. These 12 innovations have been carefully selected to fill the gaps in 5Growth´s baseline platform towards the fulfilment of the requirements introduced in Section 2.3.

TABLE 3: SELECTION OF 5GROWTH INNOVATIONS. SUMMARY.

Cluster	Category	Innovation	Brief Description
Architecture	Verticals Support	I1: Support of Radio Access in network slices	Modelling of the RAN requirements in network slice information models, based on latest 3GPP specs. Vertical Slicer logic (e.g. at the arbitration level) to handle the RAN segment of network slices. Additional southbound interfaces to request the creation or usage of network slices in the RAN segment.
		I2: Vertical-oriented Monitoring System	Separating monitoring from the infrastructure and verticals level. Use Mirantis StackLight LMA for infrastructure level monitoring. Vertical level monitoring uses the Monitoring Platform, developed in 5G-TRANSFORMER. Both monitoring subsystems require extending workloads monitoring capabilities, user metrics and parameters addition, monitoring user services according to SLAs, extending trigger rules and improving the monitoring system with AI features.
	Monitoring Orchestration	I3: Monitoring Orchestration	The 5Growth platform will deal with heterogeneous services/slices that need to be monitored in order to satisfy certain requirements, e.g. performance requirements, SLAs, etc. Due to the heterogeneous nature of the services/slices supported by the 5Growth platform, it will not be possible to monitor all of them in the same manner. Therefore, each service/slice will need different monitoring functions depending on what is to be monitored in each case. This innovation will implement a 5Gr-SO software module that manages the monitoring functions (e.g. probes, etc.) needed to monitor different services/slices according to the requirements gathered by the 5Gr-VS, SLAs, etc.
	Control and Management	I4: Control-loops stability	5Growth platform aims to provide closed-loop automation and SLA control for vertical services lifecycle management throughout the system across different layers (VS-SO-RL). This closed loop includes the process of collecting monitoring data from the services and networks, performing real-time data analytics for identifying events and alarms, so as to take proper orchestration decisions for optimization and re-configuration of the system, such as auto-scaling, self-healing and fault-tolerance, anomaly detection and automated troubleshooting, automated authentication and traffic management. Such closed-loop is to provide a continuously and automated SLA management lifecycle. It is envisioned that this closed-loop not only takes place inside each of the layers (5Gr-VS, 5Gr-SO, 5Gr-RL) to carry out their internal automated control of service and/or resources, but also across the layers which interact with each other according to predefined SLAs, policies/rules, mapping models, etc. It is essential to design the 5Grwoth architecture along with novel mechanisms and models to ensure a stable and closed loop through the system.
		I5: AI/ML Support	This innovation point will look at the opportunities to apply AI and, more specifically, ML techniques using monitoring, telemetry and analytics in the 5Gr-VS and 5Gr-SO to support resource allocation, resource control, etc. It will focus on the identifications of training sets of data for the application of ML (and even Deep Learning) techniques to monitor slice deployment and SLA compliance and to support 5Gr-VS operations. It will also concern itself with determining where in the architecture these solutions are best deployed.
	End-to-End Orchestration	I6: Federation and inter-domain	Through federation, 5Growth service providers will extend their offerings by aggregating the service catalogue and resources from other providers. This could be done at the service level (service federation) or at the resource level (resource federation). The diverse nature of services and technologies naturally leads to multi-/inter-domain scenarios in which each domain has its own orchestration deployment that must be coordinated with that of other domains towards an end-to-end service offering. In this context, novel concepts will be explored, such as Distributed Leger Technology (DLT)-based orchestration architectures, the application of intent declarations, or extending coverage for continued network service provisioning to verticals. Furthermore, federation will exploit (and possibly shape) basic functionality of the 5Growth architecture, such as algorithms for smart service orchestration (I8), abstraction and access control mechanisms, request and access to monitoring and measurement data (I2, I3), which in turn, is closely linked with auditability (I11).
		I7: Next-Generation Radio Access Network	Radio Access Network orchestration capabilities, including RAN functions, radio and computing resources, and UE profiling. The design may be inspired by O-RAN reference architecture, which is based on well-defined, standardized interfaces to enable an open, interoperable supply chain ecosystem. In this way, this innovation is the foundation for building virtualized RANs, exploiting open hardware and AI-powered radio control and orchestration mechanisms (e.g. UE profiling –based traffic steering, etc.).

Algorithms	Smart Orchestration and Control	I8: Smart Orchestration and Resource Control	5Growth aims at fully automated network slice lifecycle management. Towards that direction innovations are needed that enable ultra-fast and cost-efficient slice setup and support dynamic slices given their respective service level agreements. Therefore, innovative service orchestration, arbitration and adaptive resource allocation approaches powered by AI/ML (capitalizing on I5) need to be introduced (leveraging 5Growth monitoring capabilities/innovations), within and across domains, that go beyond the current static, use-case-specific, simplistic existing solutions, which do not address the dynamic problem in an end-to-end fashion. Due to the highly dynamic nature of the problem, stability related issues will be investigated in innovation I4. Furthermore, 5Growth will introduce dynamic resource control algorithms/techniques for performance assurance, enhancing the 5Gr-RL.
	Anomaly Detection	I9: Anomaly Detection	5Growth platform, due to heterogeneity of the supported services, requires an AI module in order to monitor deployed slices and better detect unforeseen anomalies between the services of the different slices. The current innovation will exploit 5Growth AI/ML platform module and the Monitoring Platform of 5Gr-RL/5Gr-SO in order to analyze the network, computing and storage resource utilization, slice-specific KPIs, RAN measurements, data traffic patterns, mobility patterns (if available) so as to identify anomalies, their root causes and predict future anomalies in order to enable fast recovery. Potentially, the aforementioned module will create new types of events/alerts towards 5Gr-VS.
	Forecasting and Inference	I10: Forecasting and Inference	Forecasting based on data analytics might allow to lower the number of required resources and to minimize KPIs such as energy consumption, as well as enable pre-emptive measures to be taken. However, efficient and effective algorithms shall be devised that provide plausible results by the required deadlines. This innovation will develop algorithms for demand prediction to be applied at different level of the 5Growth architecture, such vertical slicer, service, and resource layer. They will be utilised by arbitration, service orchestration and resource orchestration. The algorithm will be based on time series analysis, artificial intelligence, and machine learning.
Framework	Security and Auditability	I11: Security and Auditability	Enable dynamic defense methodologies to protect infrastructure, management, and other network functions against different types of attacks. Mission-critical assets which require strong protection against powerful adversaries (such as state-sponsored actors), and obey the input-output model, shall be protected from unknown vulnerabilities using a Dynamic Heterogeneous Redundancy (DHR) architecture in a CMD (Cyber Mimic Defense)-like approach. Others shall benefit from the exploitation of the exploration space (the basis of Moving Target Defense (MTD)) to curb the asymmetric relationship between the attacker-defender, introducing uncertainty during the information gathering phases, and disrupting the subsequent critical phases required for a successful attack. The 5Growth platform (VS-SO-RL) will deal with multiple request/response exchange between verticals and the platform that need to be guaranteed by integrating non-repudiation mechanisms. Therefore, both actors can demonstrate that a certain request/response has effectively been generated by the other one. To do this, both will save their evidences of both request and response on a private trusted store. The integration of these non-repudiation mechanisms can be applied to support the verification of orchestration SLAs, and extended to incorporate monitoring actions and even measurement themselves.
	5Growth CI/CD and containerization	I12: 5Growth CI/CD and containerization	Services containerization, Kubernetes underlay for infrastructure and automated CI/CD allows to automate whole verticals lifecycle and ship the entire platform in an automated way

The abovementioned set of innovations are specifically selected to fill the gaps identified in Section 2.3. Specifically, Table 4 depicts the mapping between each selected innovation and features which contribute to supporting 5Growth use cases but are not available in our baseline platform (gaps).

TABLE 4: MAPPING BETWEEN PLATFORM REQUIREMENTS (GAPS) AND SELECTED INNOVATIONS

Feature (gap)	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
Enhanced VS network slice sharing				X	X							
VS arbitration at runtime		X	X	X	X			X				
VS layer federation						X						
VS dynamic service composition				X		X		X				
SO automatic network service management		X	X	X	X			X	X	X		X
SO self-adaptation actions		X		X	X	X		X	X	X		
SO dynamic monitoring orchestration		X	X	X					X			
SO geo-location dependent federation				X		X						
RL PNF integration	X						X					
RL geo-specific resources	X			X	X			X				
RAN support	X						X		X			
Integral security											X	
Continuous development and integration												X

3.2. Architecture Innovations

3.2.1. Innovation 1 (I1): Support of Verticals. RAN segments in network slices

3.2.1.1. Functional and technical description

A typical end-to-end *Network Slice* spans from the Radio Access Network (RAN) to the core segment, as shown in Figure 10. On the core side, it is usually composed by a number of *Network Services* that define the "network functions" of the slice and how they are interconnected. Such network functions can be implemented through *Virtual Network Functions (VNFs)* or *Physical Network Functions (PNFs)* that are combined and interact among each other through a number of *Virtual Links (VLs)* to comprise Network Services, optionally structured in a hierarchy of *nested* and *composite* Network Services. The lifecycle of these Network Services is typically managed through Network Function Virtualization (NFV) Management and Orchestration (MANO) platforms and they

can be provisioned, configured, scaled, operated and terminated on-demand as part of the lifecycle management of the associated *Network Slice Instances (NSI)* and *Network Slice Subnet Instances (NSSI)*.

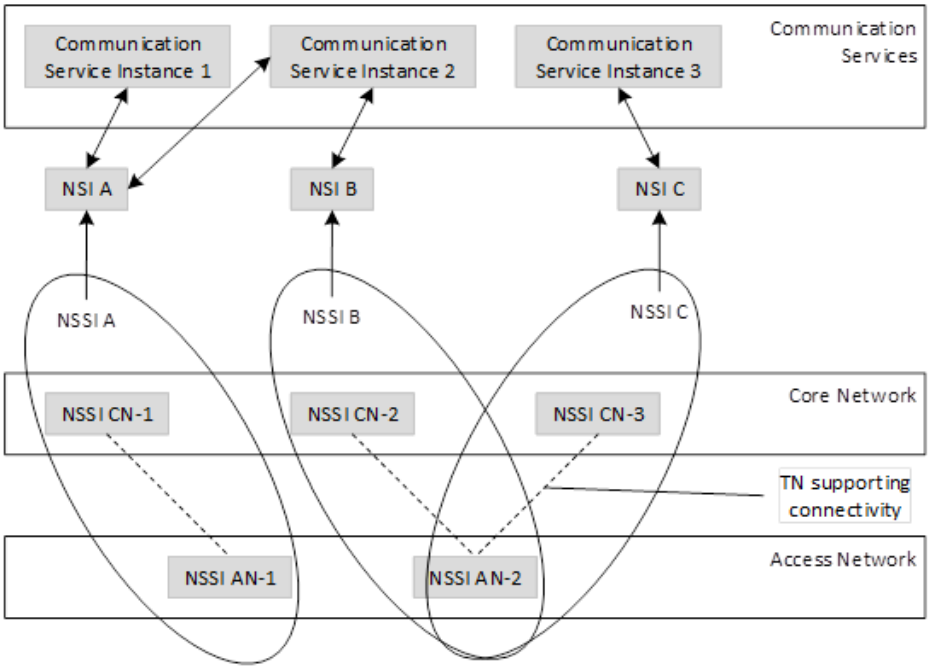

FIGURE 10: COMMUNICATION SERVICES PROVIDED BY NETWORK SLICE INSTANCES ENCOMPASSING CORE AND ACCESS NETWORK [13]

Beyond the Network Services composing the "core" segment, an end-to-end network slice includes also an "access" segment and it is characterized by the specific profile of the mobile traffic that will be served by the slice. This profile has an impact on the configuration of the RAN resources associated to the network slice, in order to guarantee the expected capabilities and performance of the slice itself. The support of vertical services built over end-to-end network slices modifies the traditional segregation between RAN and core elements. Actually, in most cases, elements from the core (i.e. from the 5G-Core network functions) are moved towards the access, including cases with core elements dedicated to the vertical and located in its premises. Hence the "border" connection points included in the abstraction view of the services, which should overlap the demarcation points of the vertical services, will typically embed the access part (i.e., the RAN) and some elements of the core. For the sake of simplicity, we name that as "RAN abstraction" but, case by case, it could include one or more element of the core network.

This enhanced modelling of the vertical services allows building slices that are customized to meet the requirements of different categories of traffic, in particular *enhanced Mobile BroadBand (eMBB)*, *Ultra-Reliable Low-Latency Communication (URLLC)* and massive *Machine Type Communication (mMTC)* traffic; each of them characterized by a different set of specific parameters.

As reported in [13], network slice requirements may include parameters like area traffic capacity, coverage area, end-to-end latency, overall user density, service availability, service reliability and User Equipment (UE) speed, which have a direct impact on the network slice capabilities offered at the RAN segment. Therefore, the network slice provisioning procedures need to involve also the

reservation and configuration of appropriate resources on the RAN segment, in the target geographical locations. Such configuration is typically handled through dedicated "RAN controllers", which are in charge of managing the RAN resources interacting with the mobile infrastructure equipment (e.g. the Remote Radio Heads (RRHs)), often using proprietary interfaces.

The support of vertical services with specific RAN characteristics in the management of the network slices, requires some enhancements in the architecture developed in the 5G-TRANSFORMER project, which constitutes the basis for the 5Growth platform, as follows:

1. **Extension of the information models in support of RAN modelling**. The information model of the *Vertical Service Blueprint (VSB)* must be extended to describe (i) the desired service type (i.e. eMBB, URLLC, MTC) and, based on this, (ii) the high-level target category of the service to identify a default range for the parameters[7] to be guaranteed by the network slice. The *Vertical Service Descriptor (VSD)* should be also extended to allow the Vertical to further refine these parameters and define tighter characteristics for the expected mobile traffic, if needed. Moreover, the *Network Slice Template (NST)* adopted in the Vertical Slicer (5Gr-VS) to describe a Network Slice will be also extended to map these parameters related to the RAN domain. A possible information model that can be adopted for the NST is represented in Figure 11, following the 3GPP specification TS 28.541 [15]. Finally, depending on the 5Growth platform architecture design and the procedures adopted to enforce the RAN configuration, additional information models used at the lower layers of the architecture may also need to be extended. For example, if the RAN configuration will be coordinated through the Service Orchestrator (5Gr-SO) as part of the wider procedures for the lifecycle management of the NFV Network Services associated to the Network Slice Instances, then the information model for the *Network Service Descriptors* (NSD) will need to be properly extended to capture the RAN requirements.

[7] For example, the service categories and the related target parameters specified in table 7.1-1 and 7.2.2-1 of [16] could be adopted for eMBB and URLLC service types respectively.

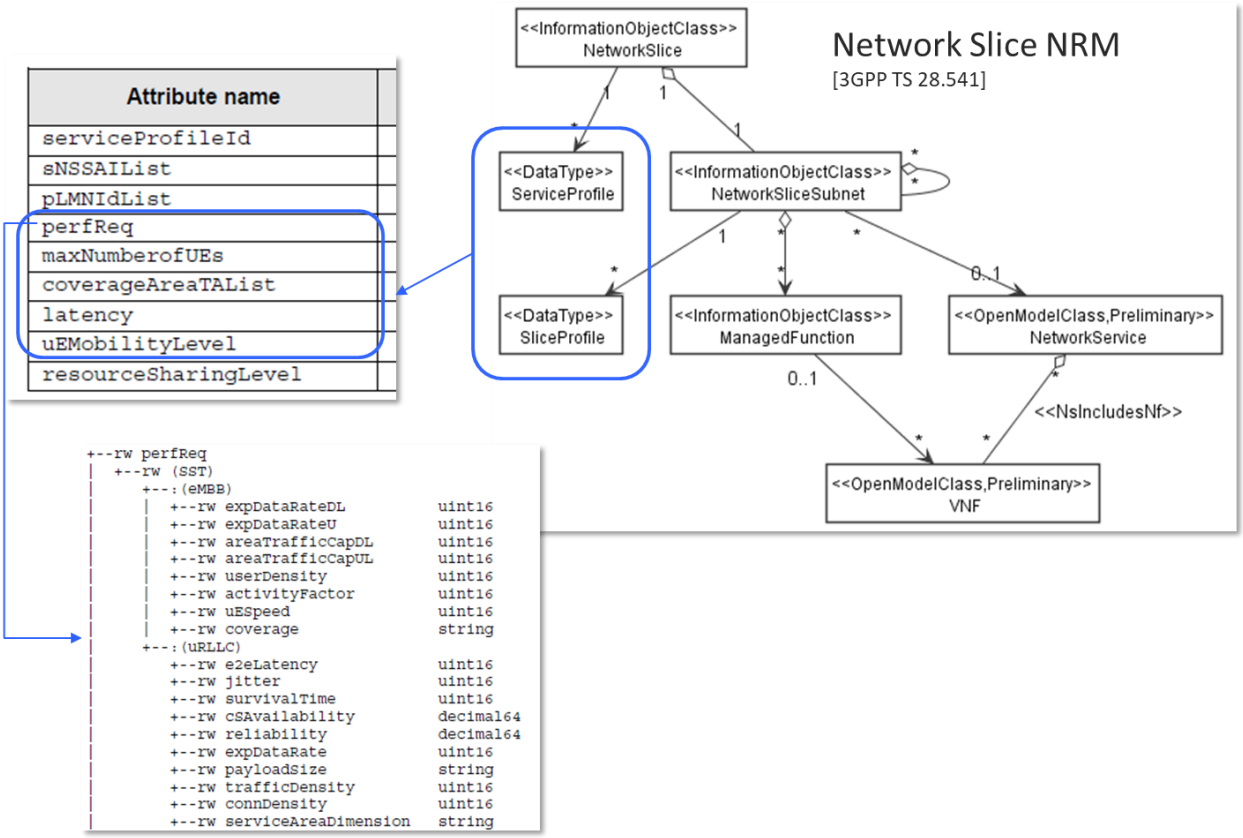

FIGURE 11: INFORMATION MODEL FOR E2E NETWORK SLICES

2. **Extension of the architecture**, in terms of interfaces and internal procedures of its functional elements to support the reservation or configuration of the RAN resources to meet the requirements of the RAN characteristics of the network slices. This aspect is further discussed in section 4.2.1.1.
3. **Definition of different levels of RAN abstraction exposed from the 5Growth Resource Layer (5Gr-RL) towards the upper layers of the 5Growth architecture,** for example, in terms of mapping between geographical area where a service is requested and coverage areas where the mobile network is available. Moreover, the abstraction view of the 5G network infrastructure will include the corresponding transport resources that can be shared among different services in each area. Depending on the abstraction level, it may also expose additional details about RAN capabilities or radio resources.

3.2.1.2. 5Growth use case support

The management of the RAN segment as part of the procedures for the provisioning of network slices can contribute to support all the 5Growth use cases where the vertical services have explicit requirements on the profiles of their mobile traffic flows. This is applicable to all 5Growth use cases so, in principle, this feature can be supported for all the different pilots. However, since the capability to configure differentiated behaviours at the radio channels depends specifically on the hardware of the 5G infrastructure installed in the target facilities, the various pilots may implement their own version of radio control and management. This has impact mostly on the implementation of the radio and transport controllers at the 5Gr-RL component, which will provide hardware-

specific interfaces towards the infrastructure equipment and, accordingly, may implement different strategies for the abstraction and configuration of the radio segment. At the upper layers, i.e. from the 5Gr-RL-RO up to 5Gr-SO and 5Gr-VS the same architectural extensions are valid for all 5Growth use cases.

3.2.2. Innovation 2 (I2): Support of Verticals. Vertical-service monitoring

3.2.2.1. Functional and technical description

Vertical-oriented Monitoring System (VoMS) provides a mechanism for collecting, storing and processing information, based on metrics, received from the deployed services. Further usage of the information may expect either analytics or enabling features such as scaling, self-healing and triggering other tasks. In addition, such monitoring metrics may be consumed by data processing modules, e.g., AI/ML platform or forecasting module, which can build models around this data.

VoMS will be a part of a Monitoring Platform (MP) which has been already implemented in the 5G-TRANSFORMER project (5GT-MP). Indeed, improvements over 5GT-MP will enable a set of innovations devoted to enhancing reliability, vertical control-loops stability, and analytical features, such as forecasting, inference and anomaly detection.

5GT-MP focused on metrics collection. In the scope of the 5Growth project, it will be extended by adding functionality for more advanced and agile metric collection, basing on dynamic probes configuration, alerting, allowing to configure triggers by verticals at any time without any downtime, triggering configuration, by utilizing messaging interaction capabilities between monitoring platform components and logs aggregation and processing for vertical services.

5Growth VoMS will lay the ground for advanced control of 5Growth platform verticals based on artificial intelligence technologies. The Monitoring System should include an API interface supporting metrics, thresholds and actions configuration in case a measure goes out of bounds. In addition, VoMs shall utilize Messaging Queues (MQ) for messages exchange between the monitoring platform core services and vertical-side monitoring agents. Messages will allow to configure agents and receive feedback data such as triggering signals. Messaging interaction between the monitoring management and monitoring agents also allows to enable dynamic probes configuration and make changes of the vertical monitoring on demand.

Finally, a very important aspect of the system architecture should be the extensibility and the ability to enable a modular design. Along with extensibility, integrability is also important and for this, an API should provide a set of functions for other systems referred for pulling data for processing and analysis – forecasting, as an example.

3.2.2.2. 5Growth use case support

VoMS can contribute to support all use cases defined in WP1. It gives the possibility to implement important features such as self-healing and auto-scaling in other innovations such as I4 (Control-loops stability, Section 3.2.4), I5 (AI/ML support, Section 3.2.5), and all algorithmic innovations

(forecasting/inference, service and resource control and anomaly detection, Sections 3.3.1-3.3.3). VoMS allows customers to integrate its software more easily with the 5Growth ecosystem and receive monitoring updates in a timely manner to fine tune the behaviour of their applications.

3.2.3. Innovation 3 (I3): Monitoring orchestration

3.2.3.1. Functional and technical description

The 5Growth platform will deal with heterogeneous services/slices that need to be monitored in order to satisfy certain requirements, e.g. performance, SLAs, etc. Due to the heterogeneous nature of the services/slices supported by the 5Growth platform, it will not be possible to monitor all of them in the same manner. Therefore, each service/slice will need different monitoring functions depending on the entities monitored in each case. This innovation will implement a 5Gr-SO software module that manages the monitoring functions (e.g. probes, etc.) required to monitor different services/slices according to the requirements gathered by the 5Gr-VS, SLAs, etc, included into the blueprints/descriptors of the services, or using descriptors of monitoring functions that have been added previously as templates.

The orchestrator will enable a dynamic deployment of monitoring probes by allowing the orchestrator to auto-discover relevant probes, leveraging in the analysis and post-processing of monitoring probes metadata gathered from the infrastructure.

The available probes will carry some information about what they are able to provide. The orchestrator will process this information and it will be able to deploy the probes looking at the requested services by the vertical. This will allow to have a dynamic deployment of probes instead of having a static definition.

The Monitoring Orchestrator can activate the monitoring for all the Network Services requested to the 5Gr-SO, according to the monitoring parameters defined in their NSD. Then, different entities (e.g. the 5Gr-SO or the 5Gr-VS) may subscribe to receive a specific subset of these parameters or notifications based on threshold mechanisms.

The Monitoring Orchestrator also controls the lifecycle of the monitor functions instances (e.g. MQs, Time Series DB, visualization services, etc.) which compose the Monitoring Platform.

Further technical details depend on the work on other innovations, such as I2 - Monitoring platform, I4 - Control-loop stability, etc., which are ahead in the time plan.

3.2.3.2. 5Growth use case support

The orchestration of monitoring functions innovation will provide a significant number of enhancements to 5Growth use cases, such as:

- Precise measurements of KPIs, locating monitoring functions optimally (using less resources to extract metrics, concentrate resources where needed). Currently, the placement of probes and monitoring functions is static, there exists a necessity to dynamically manage function placement.

- Automatic deployment, Zero-Touch orchestration, etc. Improvement of VNF and application placement in order to assist SLA assurance mechanisms depending on the monitoring results.
- Dedicated monitoring support for heterogeneous 5G use cases under a common monitoring platform.
- Enhancement of the abstraction of monitoring mechanisms for verticals, since verticals can leverage this innovation to request monitoring information at a higher level (e.g. monitor certain slices or services) and delegate to the monitoring orchestration platform the responsibility to decide what specific resources to monitor and how to aggregate the monitoring information, improving data visualization and easing the monitoring tasks for verticals.
- Dynamic monitoring probe deployment, leveraging from probe auto-discovery, reducing static configuration and definition of monitoring probes.

3.2.4. Innovation 4 (I4): Control and Management. Control-loops stability

3.2.4.1. Functional and technical description

This innovation aims to provide closed-loop automation and SLA control for vertical services lifecycle management throughout the 5Growth system. This closed loop, as shown in Figure 12, includes the process of following steps:
Step (1) collecting monitored data from services and networks (refer to I2, Section 3.2.2);
Step (2) performing real-time data analytics for identifying events and alarms and providing recommended actions with the help of AI/ML (refer to I5, Section 3.2.5);
Step (3) make proper decisions for optimization and re-configuration of the service and/or the system (refer to I8, Section 3.3.1), such as auto-scaling, self-healing, automated traffic management, which may include the following:

- Reconfigure the deployed slice by reconfiguring the slice data-plane functions (queues or routing paths) or managing the data plane traffic per slice/service.
- Redeploy/move functions or applications to another location.
- Add new functions (i.e. VNFs) to the service.
- Redeploy/migrate network services in/to another location.
- Scale network services leveraging SLA inferred scaling policies.
- Federate or delegate network service deployments to another administrative domain.
- Request resources from another administrative domain through Resource Federation.
- Arbitrate among services of the same vertical customer according to service SLAs, priorities, and business needs, at the vertical slicer or at the service orchestrator, or notify verticals that there is no possibility to ensure and enforce the SLA, and manual intervention is required.

Once the actions in all step (1) - (3) in Figure 12 are performed, the system will continue with step (1). Such a closed-loop will provide a continuously and automated SLA management lifecycle throughout the system and throughout the whole life-cycle of the services.

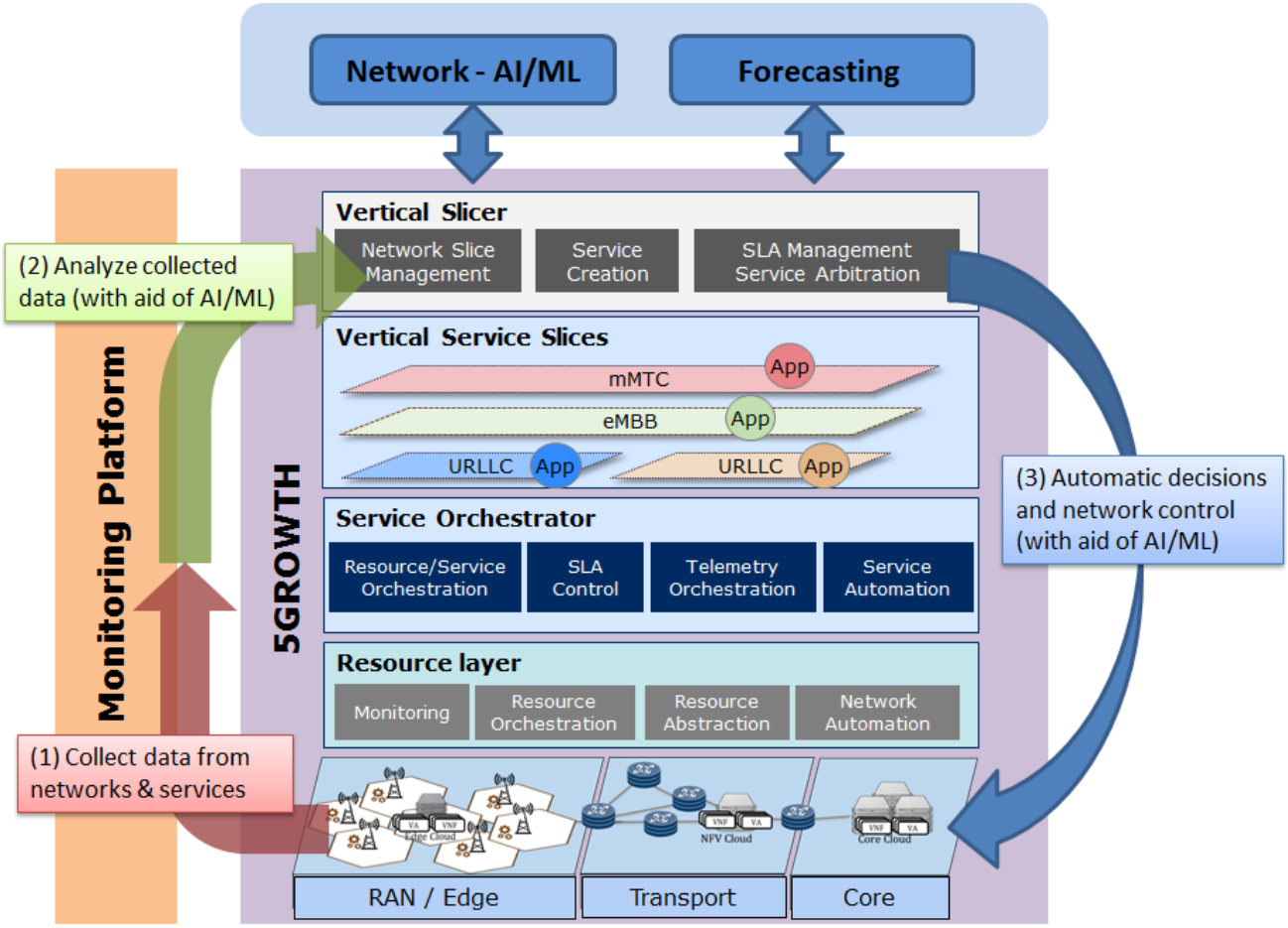

FIGURE 12: CLOSED-LOOP AUTOMATION AND SLA CONTROL FOR SERVICES LIFECYCLE MANAGEMENT

3.2.4.2. 5Growth use case support

This innovation can contribute to supporting all use cases so as to manage their service LCM and their service SLAs effectively.

3.2.5. Innovation 5 (I5): Control and Management. AI/ML Support

3.2.5.1. Functional and technical description

In order for verticals applications running within slices of the 5Growth platform, required to operate as much as possible within the constraints of SLAs, network slices must not only be correctly dimensioned according to the resources required, but should be able to rely on an orchestration and arbitration service that can respond effectively to unexpected events that may occur. The variety of applications that can be hosted in the platform makes it convenient to design AI-based algorithms that adapt to the dynamic behaviour of each slice and that can provide useful indications in real time to the decision-making logic of the platform. The data supplied to/by 5Gr-VS, 5Gr-SO and 5Gr-RL, both directly and through the elaboration of additional logic blocks, which is currently developed by other Innovations, will provide better context awareness to support their operations, ultimately allowing them to make smarter choices.

The first objective of this innovation is the identification of a suitable 5Growth platform architecture that allows the introduction of a module capable of hosting AI/ML algorithms. The dependence on pre-existing elements of the platform architecture, such as monitoring and orchestration, constitutes the main functional requirement: the innovation involves the identification of the most convenient location for the module within the architecture and an investigation into the possible need for the introduction of new supporting functional blocks or interfaces, if the existing elements are not deemed adequate.

Starting from the data made available by the telemetry and monitoring platform, this innovation focuses on identifying proper training and testing datasets, in collaboration with the verticals. The objective would then be the determination, at the architectural level, of the appropriate approach to support machine learning techniques, such as deep learning, with which, by monitoring the vertical slice deployment, it is possible to support the operations of the decision-making logic.

The following innovations, which will provide AI/ML algorithms, will be supported by the developed module:

- I2: Vertical-oriented Monitoring System
- I4: Control-loop Stability
- I8: Smart Orchestration and Resource Control
- I9: Anomaly Detection
- I10: Forecasting and Inference

Moreover, if needed by the innovations mentioned above, I5 will exploit functionalities of I3 (Section 3.2.3): Orchestration of monitoring functions, such as dynamic instantiation of probes, to support the execution of specific AI/ML algorithms that can take advantage of improved monitoring.

The architecture offered by this innovation will constitute the fundamental building block for all the supported innovations, which will be provided with a common set of features and interfaces suited for the needs of all of them.

The AI/ML algorithms supported by this innovation can continuously adapt to individual instances of slices to detect patterns of events, which can then be associated with future situations, trends or problems. If these are appropriately reported by the algorithms developed by the supported innovations, it would be possible to support features such as anomalies detection (I9, Section 3.3.2), resource consumption optimizations (I8, Section 3.2.5), etc. Therefore, they can proactively support the platform elements in the allocation and control of computing and network resources (I4, Section 3.2.4). In this way, it is possible to further guarantee the compliance with the SLA requirements imposed by the verticals.

Before deploying the vertical applications within slices, the behaviour of the AI/ML algorithms developed in other innovations (and with a direct effect on deployed slices) should be assessed to ensure that expectations are met. Therefore, the innovation will integrate mechanisms that allow to train and/or validate 5Growth AI/ML algorithms before deploying a vertical slice. Therefore, different means will be explored, including the modelling of a set of scenarios that generate an amount of data that will be used for training or validating the algorithms.

3.2.5.2. 5Growth use case support

A dynamic execution environment can expose applications to unexpected difficulties caused by unforeseen situation at runtime. Smart AI-based algorithms can effectively minimize these risks compared to more conventional algorithms and therefore many innovations in 5Growth involve their use. What is currently lacking in the platform, and we propose to provide, is a unified modular architecture that homogenizes and centralizes the utilization of AI algorithms with a common set of interfaces and functionalities that will be defined with the contribution of supported innovators. In this way, compared to the case in which each innovator implements its own solution for deployment of AI applications inside the 5Growth platform, this innovation will simplify the development of algorithms, speed up AI model training and ultimately help to provide a better service to verticals. This innovation aims to indirectly support all use cases since each of them will take advantage of one or more supported AI-powered innovations.

3.2.6. Innovation 6 (I6): End-to-End orchestration. Federation and Inter-domain

3.2.6.1. Functional and technical description

Through federation and inter-domain support, 5Growth service providers will extend their offering by aggregating the service catalogue and resources from other peering providers. This could be done at the service level (service federation or hierarchical multi-domain service support) or at the resource level (resource federation). Concerning service federation and multi-domain support, there are different types of services that can be handled, namely communication services, network slices, and NFV network services. The first two are handled between 5Growth vertical slicers (the building block embedding the CSMF and NSMF functionalities defined by 3GPP, among others) and the latter is handled between peering service orchestrators. As for resource federation, it is handled at the service layer (between peering service orchestrators).

The diverse nature of services and technologies naturally leads to multi-/inter-domain scenarios, where each domain has its own orchestration deployment that must be coordinated with that of other domains towards an end-to-end service offering. In this context, novel concepts will be explored, such as Distributed Leger Technology (DLT)-based orchestration architectures or the application of intent declarations.

Furthermore, federation will exploit (and possibly shape) basic functionality of the 5Growth architecture, such as algorithms for smart service orchestration (I8, Section 3.3.1), abstraction and access control mechanisms, request and access to monitoring and measurement data (I2, I3, Sections 3.2.2 and 3.2.3), which in turn, is closely linked with auditability (I11, Section 3.4.1).

Notice that the term federation is used when referring to non-hierarchical interactions, i.e., between two entities offering the same functionality in both providers. Furthermore, federation implies multi-domain, but multi-domain does not imply federation (e.g., hierarchical relationship).

3.2.6.2. 5Growth use case support

Innovalia

This innovation supports the first use case of the INNOVALIA pilot: *Connected Worker Remote Operation of Quality Equipment*. In this scenario, a teleworker located at INNOVALIA headquarters is able to operate remotely and in real-time a measuring machine located at INNOVALIA's customer premises. More specifically, one of the functional requirements of the use case, gathered in D1.1 [5], states that its radius of action comprises distances from Bilbao (where INNOVALIA headquarters is located) to Europe (where the different remote factories can be located). However, the network operator responsible for the public interconnection between INNOVALIA headquarters and its customer premises may not be present at all possible locations within Europe. In this case, federation is required to guarantee the full operation of the service across different administrative domains. Particularly, the network operator with presence in Bilbao managing the network and computing infrastructure for INNOVALIA will act as a consumer domain, leveraging multi-domain federation to deploy the requested service using the infrastructure of the network operator (provider domain) located closer to the remote factory, in order to increase the radius of operation while guaranteeing performance SLAs.

Aveiro

In the future, operators will provide a large variety of services based on NFV artefacts, creating heterogeneous layers to offer tailored services to verticals. In this context, operators will increasingly rely on other operators to extend their portfolio (e.g., lack of coverage of a certain area, or even secondary connectivity for reliability purposes). New business models will emerge, bringing new players to the arena (e.g., service providers). This not only allows service provisioning to verticals to be done relying on the composition of network pieces from multiple operators, but actually empowers verticals to take upon themselves to realize such multi-domain composition, based on open APIs made available by the operators. In such a complex ecosystem, flexibility is of vital importance to enable the interaction with each other at different layers of the service "stack".

In the Aveiro site, 5Growth will explore emerging business models, in particular, the one where a new player – the service provider – creates services by composing pieces coming from different network operators. In a scenario like this, service providers (operators as well as verticals may also play that role) need to interact with operators at different layers, depending on the specific implementations they have. For example, in the 5Growth context, the integration with 5G-VINNI and 5G-EVE project need to follow different models. While for 5G-VINNI the interaction is only possible at the NSMF level, for 5G-EVE the integration can only take place at CSMF level. These examples should not be seen as exceptional, but rather a sample of the heterogeneity and variety of interactions that will be available.

3.2.7. Innovation 7 (I7): End-to-End orchestration. Next Generation RAN

3.2.7.1. Functional and technical description

This innovation will allow 5Growth to support next generation Radio Access Networks (RANs) such as virtual RANs or vRANs. This includes supporting (v)RAN orchestration capabilities, e.g., RAN functions and radio and computing resource control.

In 5Growth we will first focus on O-RAN's[8] reference architecture for this purpose (see Section 3.2.7 for more details). The O-RAN Alliance was founded by operators to define requirements and build a supply chain ecosystem with the following two principles:

- Openness. The specification of open interfaces is essential to build agile and cost-effective next-generation RANs, such that smaller vendors and operators can quickly introduce new services. Moreover, open interfaces enable multi-vendor deployments, fostering competition.
- Intelligence. The advent of 5G and beyond result in highly complex networks, with high densification and richer and more demanding applications. To tame such complexity, networks must be self-driving, leveraging new learning-based technologies to automate operational network functions and reduce operational costs.

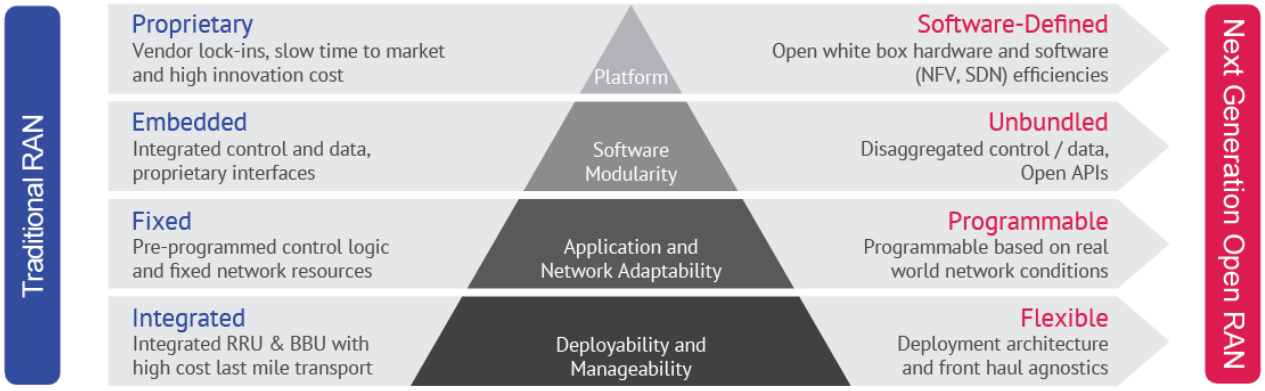

FIGURE 13: TRADITIONAL RAN VS. NEXT GENERATION OPEN RAN[9]

As illustrated in Figure 13, we will focus on the following dimensions:

- Software defined, AI/ML enabled RAN Intelligent Orchestration and Control. Next generation RANs will adopt the SDN concept by decoupling control and user planes. This extends the control plane/user plane (CP/UP) split of the Centralized/Cloud Unit (CU), being developed within 3GPP through the E1 interface, and further enhances the traditional Radio Resource Management (RRM) functions with embedded intelligence. O-RAN proposes a hierarchical orchestration and control system, with 5Growth's Service Orchestration

[8] https://www.o-ran.org/
[9] From Radisys

integrating a non-RT (Real Time) Intelligent Controller (RIC), which interacts with a Near-RT RIC via an A1 interface, the ultimate responsible to control the CU's protocol stack via an E2 interface. This is described in more detail in Section 4.2.1.1. CU CP/UP decoupling will bring substantial benefits, namely, enabling easy scaling and cost-effective solutions for the UP, and advanced control functionality. Within this dimension also techniques for guaranteeing slice performance isolation in SDN based transport networks will be proposed. Network slice isolation management represents an important aspect of Zero-touch network and Service Management (ZSM) [43], especially when sufficient level of independency between the network slice instances with tolerable interference in respect of the requirements of the slice customers shall be guaranteed. This contribution would like to guarantee slices carrying different traffic in the RAN (e.g., different types of fronthaul functional splits or different application traffic) with the requested performance isolation. The considered performance parameters will be, for example, minimum capacity and latency. An example of exploitable techniques is the utilization of meters in SDN based networks.

- Virtualized RAN (vRAN). RAN cloudification is well-recognized as key technology for next generation RANs. Operators are delivering NFVI/VIM requirements to enhance virtualization platforms in support of various RAN splits. Whenever possible, 5Growth will integrate orchestration and resource control features required for vRANs.
- Open Interfaces. Motivated by the work of O-RAN's reference architecture, 5Growth will support a set of key interfaces between multiple decoupled RAN components. These include enhanced 3GPP interfaces (F1, W1, E1, X2, Xn) for true multi-vendor interoperability; and will study new ones proposed by O-RAN (or other initiatives) to support novel RAN features such as open fronthaul interfaces between Distributed Unit (DU) and Remote Radio Unit (RRU), an E2 interface and an A1 interface between the Service Orchestrator containing the non-RT RIC functionality and the near-RT RIC functionality deployed closer to the CUs.

3.2.7.1. 5Growth use case support

O-RAN architecture allows great flexibility from the service deployment perspective. The open interfaces defined in this initiative permit coexistence of different components and so it enables easy integration of architectural components even deployed at premises from different actors.

The fact of interoperability, per se, enable a more competitive supplier ecosystem and a more cost-effective RAN deployment. This is especially interesting when considering the vertical use cases of 5Growth, requiring in some of them an extremely high availability. Thus, the more cost-efficient the solution, the more economic to make it robust.

It will be possible for vertical customers, in some cases, to assume the deployment of RAN capabilities in their own premises for providing additional coverage or capacity to the network operator completing the connectivity towards the external networks. These scenarios of interconnection and usage of resources of multiple providers require from coordinated orchestration capabilities to ensure smooth end-to-end service delivery. In that direction, O-RAN is also defining APIs and interfaces.

The 5Growth use cases can benefit from this open approach at RAN level in several ways. One can be the aforementioned interconnection of resources from the Vertical and the Service Provider to provision the service in a multi-domain fashion. To this respect, the neutral host approach currently proposed in O-RAN's "The Open Fronthaul Interfaces WorkGroup (WG4)" could permit the exploration of new business scenarios with different Service Providers sharing RU deployments.

Alternatively, O-RAN could be leveraged to complement existing and conventional deployments on the Service Provider side by augmenting footprint through virtualized RAN deployments for specific service scenarios as demanded by Verticals, while retaining the ownership by the Service Provider administrative domain.

From a general perspective, all the use cases in 5Growth can be supported by the O-RAN approach, independently if the RAN architectural components pertain to the Network Operator, or if a multi-domain scenario where e.g. RU is owned by the Vertical and the rest is owned by the Operator applies. Some scenarios of robustness and reliability are intended to be analysed to determine a proper deployment schema for future Vertical services.

3.3. Algorithm Innovations

3.3.1. Innovation 8 (I8): Smart orchestration and resource control

3.3.1.1. Functional and technical description

5Growth aims at fully automated network slice lifecycle management and SLA enforcement. In addition to the architectural innovations to achieve so (see I4, Section 3.2.4), novel framework, algorithms and techniques are needed. Introduced innovations will enable ultra-fast and optimized deployment, management and control of elastic network slices (and their resources) that support dynamic traffic / workload demands and their respective SLAs. 5Growth therefore sets the following targets.

Target 1: The enhanced software stack will include innovative SLA-aware service orchestration/arbitration, adaptive resource allocation and scaling approaches, leveraging 5Growth monitoring capabilities and data analytics, within and across (administrative and technology) domains. Such approaches need to go beyond the current static, use-case-specific, simplistic existing solutions, which do not address the dynamic end-to-end problem. The problem will be addressed across all three layers of the 5Growth architecture; from service arbitration at the 5Gr-VS to service orchestration at the 5Gr-SO and actual allocation of resources at the 5Gr-RL. Solutions powered by AI/ML will be investigated, given (i) the uncertainty in terms of workload/traffic, slice instances, etc. of the environment and (ii) the required agility considering the evolving or different operational objectives.

Target 2: 5Growth will incorporate into the baseline platform, dynamic resource control algorithms/techniques for performance management, enhancing the 5Gr-RL. In particular, 5Growth will introduce (re)programmable SLA-aware traffic management algorithms at the data plane (such as scheduling, queue management schemes etc.) focusing on potential bottlenecks (e.g., the RAN), investigating performance assurance per slice via closed loop interactions at the resource layer.

To this aim, 5Growth will introduce the following innovations per layer of the architecture:

- 5Growth Vertical Slicer (5Gr-VS):
  - Devise new algorithms for service arbitration in the vertical slicer using optimization and machine learning.
- 5Growth Service Orchestrator (5Gr-SO):
  - Devise new algorithms for service and resource orchestration within and across different technology and administrative domains.
- 5Growth Resource Layer (5Gr-RL):
  - Devise dynamic resource allocation and re-optimization algorithms within heterogeneous data plane infrastructures. This entails: i) satisfying the SLA's requirements of the network services (related to a targeted network slice) in terms of end-to-end latency, minimum bandwidth, etc.; ii) provide required resource adaptation among the different technological layers (i.e., wireless, packet and optical switching) fostering grooming and tunnelling policies and strategies (i.e., statistical multiplexing); iii) targeting decisions to enhance the optimal resource utilization (e.g., bandwidth, transceivers, optical spectrum, etc.).
  - Support data plane customization and performance isolation per slice in order to meet network level KPIs and SLAs, via real time control and programmable traffic management.
  - Enable programmable data plane arbitration (e.g., via scheduling) across and within network slices.

3.3.1.2. 5Growth use case support

This innovation can contribute to supporting all use cases in WP1 (see D1.1 [5]) and pilots to perform service and resource orchestration according to the use case needs.

3.3.2. Innovation 9 (I9): Anomaly detection algorithms

3.3.2.1. Functional and technical description

5Growth platform, due to the heterogeneity of the services that it will support, requires an AI-based module in order to help administrators and slice tenants to detect and diagnose anomalies among the services of the different slices deployed on the same infrastructure. The current innovation proposes to build a flexible AI module, which will rely on 5Growth Monitoring Platform for receiving monitored network context information; the proposed AI framework will be capable of analyzing aggregated and fine-grained data such as resource-level data (e.g., computing and storage utilization, RAN measurements etc.), flow-level data, service KPIs and infrastructure KPIs as well as traffic patterns and mobility patterns (if available), in order to identify network anomalies

and their root cause. In order for the anomaly detection module to attempt performing a solid prediction, it will analyze historical data so as to a) potentially identify new types of anomalies, and extend the already defined set of anomalies, b) enable faster recovery in the future for similar types of previously-monitored anomalies. In the context of the work that will take place during the innovation's design and implementation, it will also be investigated whether the aforementioned framework will extend the existing events/alerts towards the 5GT-VS.

The initial high-level description of anomaly detection workflow is illustrated in Figure 14:

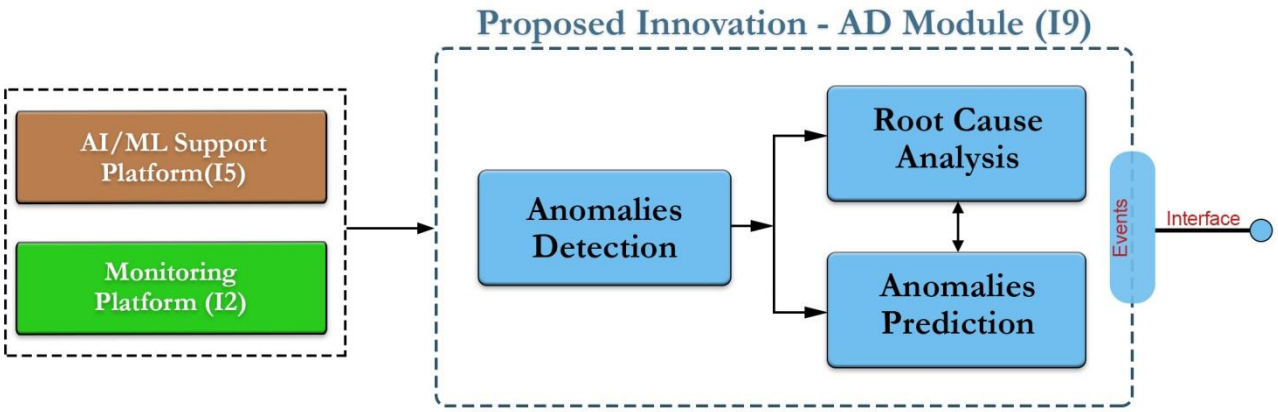

FIGURE 14: INITIAL HIGH LEVEL DESCRIPTION OF ANOMALY DETECTION (AD) MODULE

The Anomaly Detection (AD) module will provide a significant number of features to 5Growth. The main ones are presented below.

- Development of algorithms (enhance 5Gr-SO), which are able to detect near-real time resource utilization events and anomalies (e.g., high CPU consumption, high memory usage etc.), service, network (RAN, transport/backhaul and core), UE/session as well as cloud related anomalies which will potentially lead to a resource shortage or system malfunction. In order to achieve that, processing multiple entry data from hybrid sources is essential while keeping balance between computing power (and as a result system's latency) and data volume.
- Development of algorithms, which are able to perform root cause analysis. It is of major importance being able to identify the root cause of the problem in order to prevent future occurrences.
- Development of algorithms that will combine the detected events/anomalies along with the root cause analysis results, in order to perform prediction of already observed types of anomalies.
- Interaction of Anomaly Detection module with the AI/ML Support (I5, Section 3.2.5) in order for the I5 to determine the optimal placement of this innovation in the 5Growth architecture and identify proper datasets, in collaboration with the verticals.
- Make use in I2 (Section 3.2.2) concerning Vertical-oriented Monitoring System. More specifically, make use of the extended Monitoring Platform that take place in the Service Orchestrator in order to collect the required data for the analysis.

3.3.2.2. 5Growth use case support

The anomaly detection module will be of significant importance in the 5Growth platform due to the fact that it can support many different use cases. It is essential to define the potential anomalies, which can take place in the UCs, in collaboration with the pilot partners so as to better decide were the AD module fits best. Nevertheless, this innovation will be exploited by all use cases that require different types of service, network, UE/session as well as cloud related anomalies to be timely detected, either in reactive or a proactive type of operation and adapt their operation accordingly.

Additionally, this Innovation will provide a significant number of benefits in the 5Growth platform:

- The AD Module will enhance the 5Gr-SO in order to detect anomalies which could result in resource shortage and system malfunction.
- The Prediction sub-module will be able to detect already observed anomalies so as to proactively produce events which will be exploited by the 5Gr-SO to prevent system failures.
- The AD Module will create additional and more complex type of events in order to be exploited by the final consumers.

3.3.3. Innovation 10 (I10): Forecasting and inference

3.3.3.1. Functional and technical description

Due to the strict requirements (e.g., availability or minimum guaranteed bandwidth) of the vertical use cases to be run over the 5Growth platform, predicting the occurrence of a given event (e.g., when a node reaches full capacity) or forecasting how demand will evolve over time (e.g., forecast an increase of the number of users accessing the service) become key capabilities that need to be supported by 5Growth to be able to pre-emptively adjust the offered services. For example, auto-scaling, self-healing and self-reconfiguration mechanisms can be triggered to mitigate the effects that such events could have in the offered services and, in the worst-case scenario, to avoid their downtime. Predictive analytics is also one of the building blocks of the ETSI Zero-touch network and Service Management (ZSM) [43]. There, it is reported that prediction/forecasting can be used in the following aspects "(…) prediction of resource usage to avoid congestion; prediction of service KPIs to optimize service; prediction of service health state to ensure reliability; etc."

This is the goal of this innovation, which is proposed to deal with forecasting and inference. Forecasting based on data analytics may allow to lower the number of resources required and to satisfy KPIs such as energy constraints, as well as enable pre-emptive measures. However, efficient and effective algorithms shall be devised that provide plausible results by the required deadlines. This innovation will develop algorithms for demand and event prediction to be applied at different level of the 5Growth architecture, such vertical slicer, service, and resource layer. They will be utilized by arbitration, service orchestration and resource orchestration. The algorithm will be based on time series analysis, artificial intelligence, machine learning, and statistical inference.

3.3.3.2. 5Growth use case support

Because the innovations developed within I10 could be utilized by many layers and functional elements of the 5Growth platform there is no specific use case to be supported but this innovation could be beneficial to all the considered use cases. For example, forecasting and inference based on the monitored requests for a specific service might allow to proactively allocate resources (i.e., VMs, containers, connections) to support the users while preserving specific SLA. Additionally, forecasting and inference can be utilized at the resource layer in an agnostic way to better allocate VMs to physical resources based on historic data of VM requests.

3.4. Framework Innovations

3.4.1. Innovation 11 (I11): Security and auditability

3.4.1.1. Functional and technical description

By using the 5Growth platforms deployed at the industry vertical premises, these verticals will be able to carry out testing and KPI validation activities for use case trialing. To support the execution of these experimentation-related activities, the deployed 5Growth platforms will need to go through multiple request-response exchanges, within the vertical environment and beyond. Depending on the considered use cases, these activities will be circumscribed to the non-public network (NPN) used by the vertical or require the interaction with public networks (in the 5Growth project context, ICT-17 facilities such as 5G-VINNI and 5G-EVE). In this regard, both internal (within a given platform deployment) and external (between 5Growth platforms and the corresponding ICT-17 management systems) control message exchanges must be supported. Therefore, the 5Growth platform shall provide means to allow secure interactions among multiple actors from (potentially) different administrative domains.

This innovation focuses on making the 5Growth platform a verifiable and trustable stack, able to support the exchange of request-response messages involving multiple actors for vertical-oriented experimentation activities. To this end, non-repudiation mechanisms complemented with advanced security methodologies need to be defined and integrated into the platform.

The non-repudiation principle means that each pair of actors taking part in any interaction (typically, a requestor and a responder) can demonstrate that a certain request or response message has been effectively generated by the other one, relating them to previous relevant messages, and associating them with a consistent temporal line. To achieve this, both actors shall save their (equivalent) evidence of the exchanges on a private trusted store, as shown in Figure 15. These evidences will provide means for external verifiability of all messages exchanges, allowing system auditability, and further applications enabled by it. The integration of non-repudiation mechanisms will allow the 5Growth platform to keep audit trails for traceability purpose, by storing a trusted record of all the request-response interactions exchanged on the platform. These mechanisms shall be compatible with the protocol(s) selected for these exchanges.

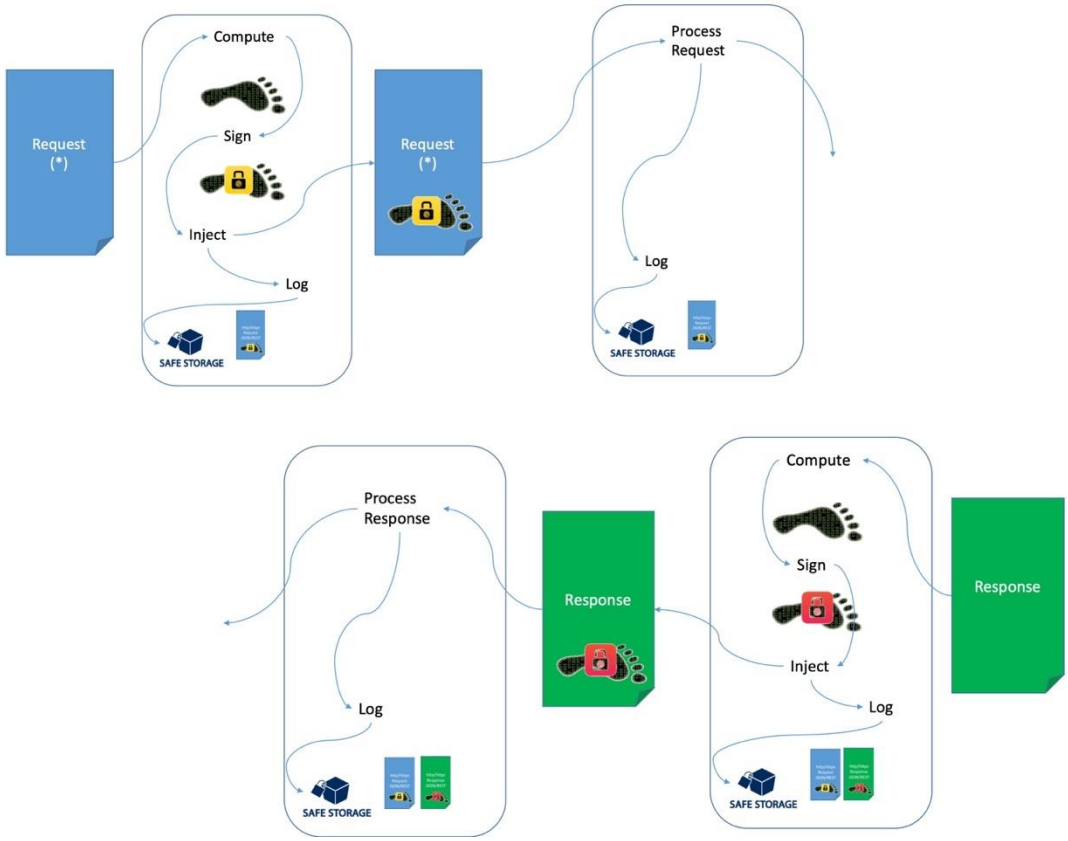

FIGURE 15: NON-REPUDIATION MECHANISM: REQUEST AND RESPONSE

The operations described above as sources for the audit trails and non-repudiation typically happen on isolated, private *management networks* (virtual or physical). Acquiring illegitimate access to one of these private networks broadens the scope of compromise (or disruption) over the system to more than just forging interactions/tampering with legitimate exchanges. Attackers can now interact with the lower levels of the service stack, bypassing the application logic, and subverting entire functions. For instance, exploiting vulnerabilities in the operating system that runs the service, or the webserver that powers the service, the attacker can subvert the service controls enforced at the application level. This kind of access to a management network can threaten all parameters of the CIA/IAC triad (Integrity, Availability, and Confidentiality), should an exploitable vulnerability exist in the services of that network (e.g., the 5Growth platform).

The sophistication of the attack depends on the type of attacker, its motivations, and its objectives. Powerful attackers may have the resources to discover and exploit zero-day vulnerabilities, which are hard to defend with traditional methods. These are the type of attackers that conduct high-valued industrial espionage or state-sponsored hacking. Less powerful attackers, such as cybervandals, are likely to use newly disclosed vulnerabilities or vulnerabilities that should already be known to the defender. Knowing about the existence of these vulnerabilities does not necessarily imply they are properly resolved in the platform, or that their detection is easy before the successful exploitation happens (with potential consequences over the system).

According to what is mentioned above, it is desirable to have an additional layer of protection on top of the traditional defense systems (following a defense-in-depth philosophy). A critical fact to

have in mind is that performing a remote attack against a target always carries some degree of uncertainty over its outcome. This uncertainty is a characteristic of the exploitation process, which is sensitive to small variabilities in the target systems, additional unforeseen protections that may be in place, or even plain luck, much like in any penetration testing endeavor. However, the longer the attacker maintains access to that network, the more information it can gather to improve the chances of being successful (e.g., tailoring the exploit to the specific versions and configurations of the targeted systems). This advantage is often referred to as the asymmetric relationship between the attacker and the defender because the attacker typically has time on its side.

A defense mechanism such as Moving Target Defense (MTD) will curb the above-referred advantage by imposing a time limit upon the validity of the gathered information, after which becomes unactionable. Therefore, exploiting the private service(s) should become as hard as the first instants in which the attacker got a foothold to that network. In particular, when applying Moving Target Network Defense (MTND), this means that:

- Mutating end-point information such as IP addresses or layer-4 ports will make it harder for an attacker to achieve a lasting connection to/from the vulnerable service. Even if the attacker knows what is the service to exploit (and how), as long as the attacker is not a legitimate user of the service (e.g., with lower privileges) and the secret key to the movement remains unknown, the attacker will first have to figure out a way to counter the movement to send the exploit's payload.
- Not knowing the secret to the movement will aid other security systems, such as an Intrusion Detection System (IDS), to detect if an attacker is in the network (i.e., missing the right service port will be an evident anomalous access pattern).
- Resolved Path mutation can help counter Denial of Service (DoS) attacks, or alleviate the effects of traffic patterning conducted by a line eavesdropper.
- Fingerprint mutation can hinder the attacker's information gathering process, feeding wrong (but believable) information about the service, which would otherwise be critical to the construction of a successful exploit.
- MTD may also be a honeynet enabling tool, which is instrumental in understanding how attackers are attacking our networks/services.

Figure 16 depicts how the MTD mechanism would work, showing a mutator function that is performed within the endpoints. The mutator can either be directly integrated with the application or be a proxy-alike program that intercepts the requests of an existing application. If a shared secret exists between the parties, the mutator can use this shared secret to derive short-lived session keys that will control the mutation. Those who hold the secret will be able to revert the mutation in a deterministic fashion, while those who do not will observe an unpredictable behavior.

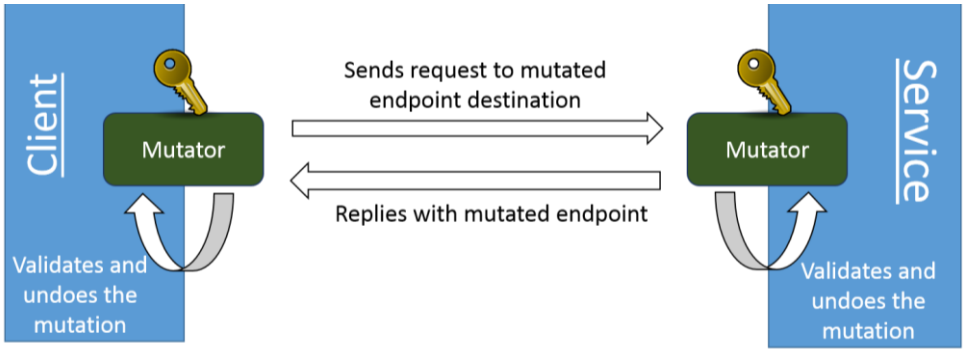

FIGURE 16: MTD ENDPOINT BASED MUTATION MECHANISM

Should the threat model allow for, the mutator function can be offloaded to a component outside of the service being protected (as shown in Figure 17). This strategy is useful to eliminate the modification of service functions, such as when protecting functions that are controlled by another party. It is also useful to offload the resource usage of the mutation function. This approach requires traffic steering capabilities, which can be done either by Service Function Chaining mechanisms or in a plain SDN controller application.

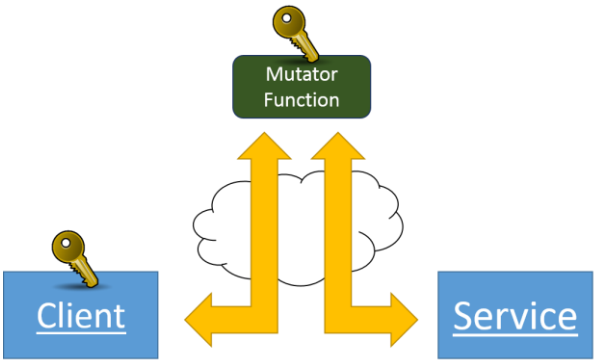

FIGURE 17: MTD OFFLOADED MUTATION MECHANISM

MTD is a versatile defense mechanism that can be used to protect many types of services, without needing prior knowledge of the inner-workings of the service itself. However, the MTD mechanisms are not meant to assure the correctness of the function when on an adversarial setting; it will just curb the asymmetric relationship.

If such a stringent requirement must be fulfilled, as is the case in mission-critical functions that impact public utilities (energy grids) or public safety (transportation signals), the correctness of the functionality in an adversarial setting will have to be assured by another type of approach. In this context, functionality refers not just to the Network Function, but also the 5Growth platform services and underlying providers/resources which are entrusted with rules enforcement. Cyber Mimic Defense (CMD) is a suitable approach for this type of scenario. However, unlike MTD, which is very versatile, the use of CMD requires multiple implementations of the same function by different parties, and the function being protected must obey the Input-Process-Output (IPO) model. That is, the output of the function must be calculated in a fully deterministic way when given the same input.

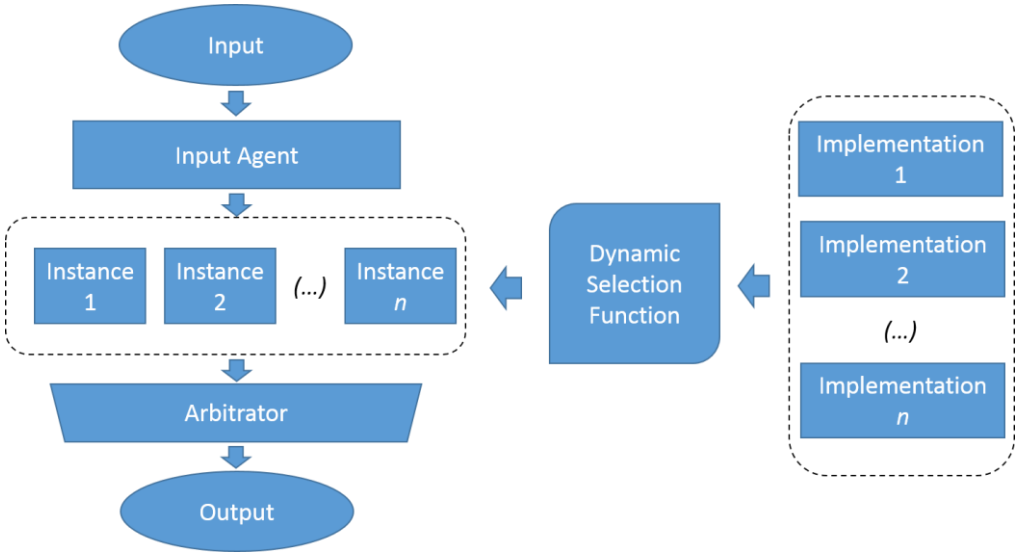

FIGURE 18: MISSION CRITICAL CORRECTNESS ASSURANCE MECHANISM (DHR FROM CMD)

Figure 18 depicts the CMD mechanism architecture, which is also known as the Dynamic Heterogeneous Redundancy (DHR) architecture. The use of this mechanism is tightly coupled with the deployment of the mission-critical function.

3.4.1.1. 5Growth use case support

The non-repudiation principle allows any requestor to prove that the alleged responder in fact sent a specific message, and vice-versa. This feature is needed for verifiable traceability and trustworthiness among different actors taking part in the service value chain, especially across different administrative domains. As a result, the integration of non-repudiation mechanisms is applicable to all 5Growth vertical pilots, and their respective use cases: UC1, UC2, UC3, and UC4.

Similarly, the MTD mechanisms are usable in all 5Growth vertical pilots and their respective use cases, as these mechanisms protect the internal communications between the components of the platform.

The CMD mechanisms target the transportation and energy vertical pilots, in particular, the UC1 of transportation (safety-critical communications), and the UC2 of the energy pilot (public-utility critical signalling).

3.4.2. Innovation 12 (I12): 5Growth CI/CD

3.4.2.1. Functional and technical description

5Growth CI/CD aims to 5Growth platform lifecycle management automation, especially such time and resource demanding stages as developing, testing, integration and deploying the platform. In order to achieve full platform automation and keep it as close as possible to modern Dev-Ops paradigms, like end to end automation, continuous integration and continuous deployment, infrastructure as code, containerization, orchestration and test automation.

CI/CD, Dockerizing 5Growth platform, alongside with Kubernetes orchestration brings flexibility, repeatability, reliability, convenience to the project and allows to build easily manageable platform.

Using open source, well known to community and production ready tools alongside with widespread automation allows to significantly reduce time-to-market and get joint software platform, that doesn't require new skills for supporting staff.

3.4.2.1. 5Growth use case support

5Growth CI/CD can contribute to supporting all use cases, and allow to reduce time of every operation, increase repeatability, reliability and allows to build easily manageable platform.

4. Extensions over 5Growth Baseline Architecture

The abovementioned set of selected innovations will be led by the different tasks T2.2, T2.3, and T2.4, according to a work plan that is ultimately coordinated by T2.1. Each task T2.2, T2.3 and T2.4 is responsible to duly perform unit tests (UTs) that validate the ability of each innovation to do as promised. All work is coordinated and integrated in T2.1, which is the task responsible to perform integration tests (ITs) to ensure a coherent and safe operation of the 5Growth platform as a whole. To organize the design and implementation operations, T2.1 has prepared a work plan for UTs and ITs which will result ultimately into two software releases: Release 1 in M12 and Release 2 in M24.

The remaining summarizes 5Growth release plan and introduces the design of each innovation.

4.1. Release plan

T2.1 has planned out design, implementation and test work to ensure smooth evolution of the 5Growth platform (some innovations have dependencies between them) and a balanced work load across tasks T2.2, T2.3, T2.4. In addition to considering those innovations that are prioritize for the work in WP3, such plan has resulted in a 2-release plan that is summarized in Table 5.

Accordingly, Release 1 (R1) will include a first version of I1, I2, I4 and I8, (design/implementation) work that has already been initiated in the first months of the project (e.g. preliminary results are shown in Section 4.2.2.1 for I8). The remaining innovations, in addition to extensions to I1, I2, I4 and I8, will appear in its final version in M24 in Release 2 (R2). Prior to each release, a set of unit tests (UTs) by each task T2.2, T2.3 and T2.4 and integration tests (ITs) will be performed in the scope of T2.1 to guarantee a consistent and bug-free platform.

TABLE 5: RELEASE PLAN

Cluster	Category	Innovation	Release 1 (M12)	Release 2 (M24)
Architecture	Vertical support	I1: RAN segment in network slices	X	X
		I2: Vertical-service monitoring	X	X
	Monitoring orchestration	I3: Monitoring orchestration		X
	Control and management	I4: Control-loops stability	X	X
		I5: AI/ML support		X
	E2E orchestration	I6: Federation and inter-domain		X
		I7: Next Generation RAN		X
Algorithms	Smart orchestration and resource control algorithms	I8: Smart orchestration and resource control algorithms	X	X
	Anomaly detection	I9: Anomaly detection		X
	Forecasting and inference	I10: Forecasting and inference		X
Framework	Security and auditability	I11: Security and auditability		X
	5Growth CI/CD and containerization	I12: 5Growth CI/CD and containerization	X	X

4.2. 5Growth Extensions

This section summarizes the design of the work carried out since the selection process and M6. Like before, we present such summary for each of the individual innovations selected for the project.

4.2.1. Architecture Innovations

4.2.1.1. Innovation 1 (I1): Support of Verticals. RAN segments in network slices

As discussed in section 3.2.1.1, the support of the RAN management in the end-to-end network slice provisioning requires to reserve and configure RAN and transport resources according to the slice requirements related to the expected mobile traffic. This configuration is handled through radio and transport controllers, which will be placed at the 5Gr-RL[10]. Such RAN controller system is in charge of RAN resource management and control. This entity will be an enhanced version of the RAN controller initially introduced in the 5G-TRANSFORMER MTP architecture (see section 2.2.3) and it will enable the configuration of the RAN according to the related parameters specified in the network slice model.

In particular, the 5Gr-RL will perform the following tasks with reference to the RAN management:

- Provides an abstract view of the infrastructure resource as combination of radio and transport in terms of service parameters.
- Receives requests from the 5Gr-SO on this abstract view and translate them in order to provide the correct configuration of resources (both transport and radio).

The 5Gr-RL is thus in charge of building and operating the abstract view of the 5G mobile infrastructure, which includes the radio resources and the corresponding transport network. This abstract view will need to provide the right level of information and configuration capabilities towards the upper functional building blocks (e.g. 5Gr-SO and/or 5Gr-VS), to enable the management of the RAN segment of network slices. Internally, the 5Gr-RL will translate the upcoming requests into commands towards the physical and virtual resources, using direct interfaces towards the related entities. A "Resource Orchestrator" within the 5Gr-RL (5Gr-RL-RO in Figure 19 and Figure 20) will combine the transport and radio resources, in order to expose to the upper layers an abstract view with service parameters that hide the specific and often technology-dependent details of the transport and radio domains. Moreover, this Resource Orchestrator will translate such abstract view towards the physical and virtual nodes on the basis of the received requests.

[10] It should be noted that advanced RAN management architectures for 5G NR (for example the ORAN architecture) may involve RAN controllers that include both "management/orchestration" and "resource control" functionalities. We assume that the 5Gr-RL is in charge of managing all these functionalities, providing a global abstraction of the radio segment towards the 5Gr-SO.

It should be noted that the actual capabilities of the radio controller, as well as its internal procedures, will be tightly associated to the RAN control features offered by the RAN equipment deployed in the physical 5G infrastructure. In other terms, the radio controller can actually create and configure dedicated network slices in the RAN domain following an on-demand approach only if this functionality is supported by the target equipment. Otherwise, it may apply procedures based on a different and vendor-specific logic. For example, a number of network slices compliant with a pre-defined set of service categories may be statically pre-configured in the RAN domain and selected on-demand based on the service profile declared in the request. The internal logic of vendor-specific RAN controller is considered out of scope for WP2, since entirely dependent on the particular infrastructure. However, some solutions based on standardized interfaces for next-generation RANs will be studied towards the end of the project as part of the innovation described in Section 3.2.7. For this reason, ad-hoc solutions will be implemented in the context of WP3/4 for the target pilots and the related RAN infrastructure. In the context of WP2, we only assume that the RAN controller will offer an external interface to enable the request of RAN resource based on a given service profile matching the parameters defined in the network slice information model.

Related to the transport, instead, the 5Gr-RL will directly trigger the corresponding control (e.g. SDN) and/or management procedures of the transport nodes, by translating the requests received by the upper layers on the abstract view built by its Resource Orchestrator.

At the upper layers of the 5Growth architecture, two different approaches can be adopted to support the RAN management at the network slice level, each of them with a different impact on the extensions required on the original components of the 5G-TRANSFORMER architecture. In both cases, the considerations about the RAN-related extensions of the information models at the VSB, VSD and NST level already analysed in section 3.2.1.1 are valid. This has an impact at the 5Gr-VS level, where all catalogues and North-Bound Interfaces (NBI) for the specification of VSBs, VSDs and NSTs will need to be updated accordingly, as well as the internal logic of the Vertical Service Management Function (VSMF) and the related Translator and Arbitrator functionalities. This is represented by the 5Gr-VS internal components in red depicted in Figure 19 and Figure 20.

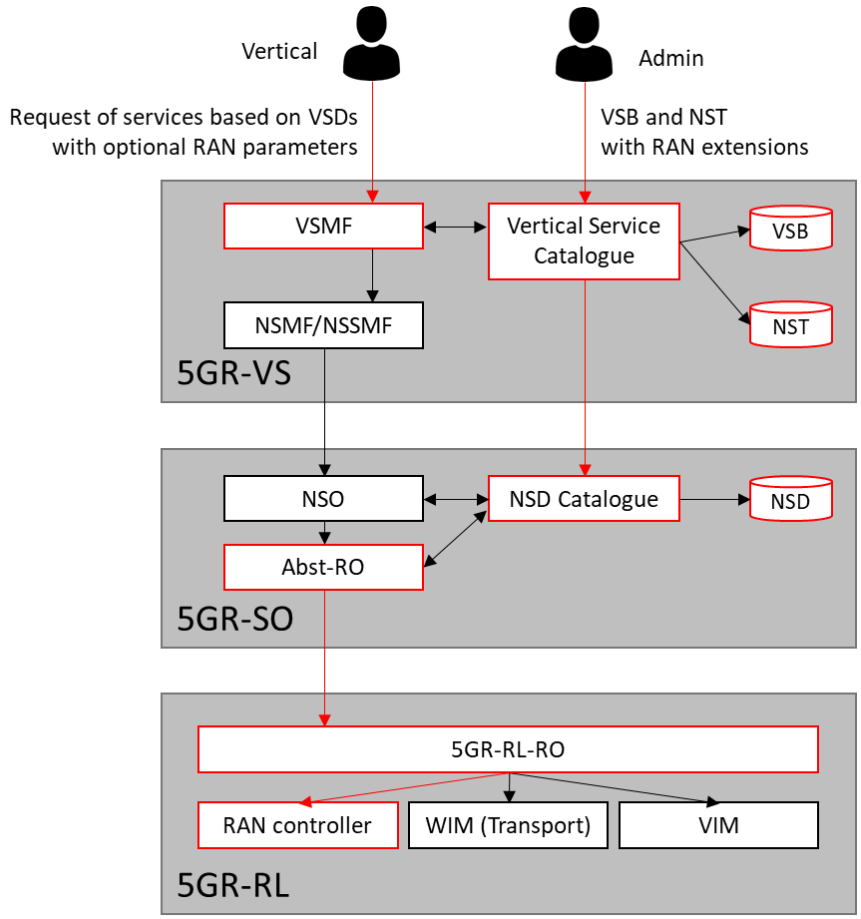

FIGURE 19: 5GROWTH ARCHITECTURE IN SUPPORT OF RAN MANAGEMENT (OPTION 1)

However, the actual request of RAN configurations can be either coordinated at the 5Gr-SO level, as represented in the first architectural option in Figure 19, or handled directly at the Network Slice (Subnet) Management Function (NSMF/NSSMF) in the 5Gr-VS, as represented in the second architectural option in Figure 20. The second option would require an extension of the NSMF/NSSMF components of the 5GT-VS to interact directly with the Resource Orchestrator at 5Gr-RL, without requiring any modification of the 5GT-SO. However, this approach would break one of the basic concepts of the 5Growth architecture, where the Vertical Slicer does not have any view or control on the underlying resources. The first option, more aligned with the overall principles of the global architecture, would require extensions at the 5Gr-SO level. In particular the NSDs and/or the Network Service Instance request should be extended with RAN-related parameters and the 5Gr-SO Resource Orchestrator (the Abst-RO in Figure 19) would need to interact with the 5Gr-RL-RO at the 5Gr-RL to request the RAN configuration accordingly.

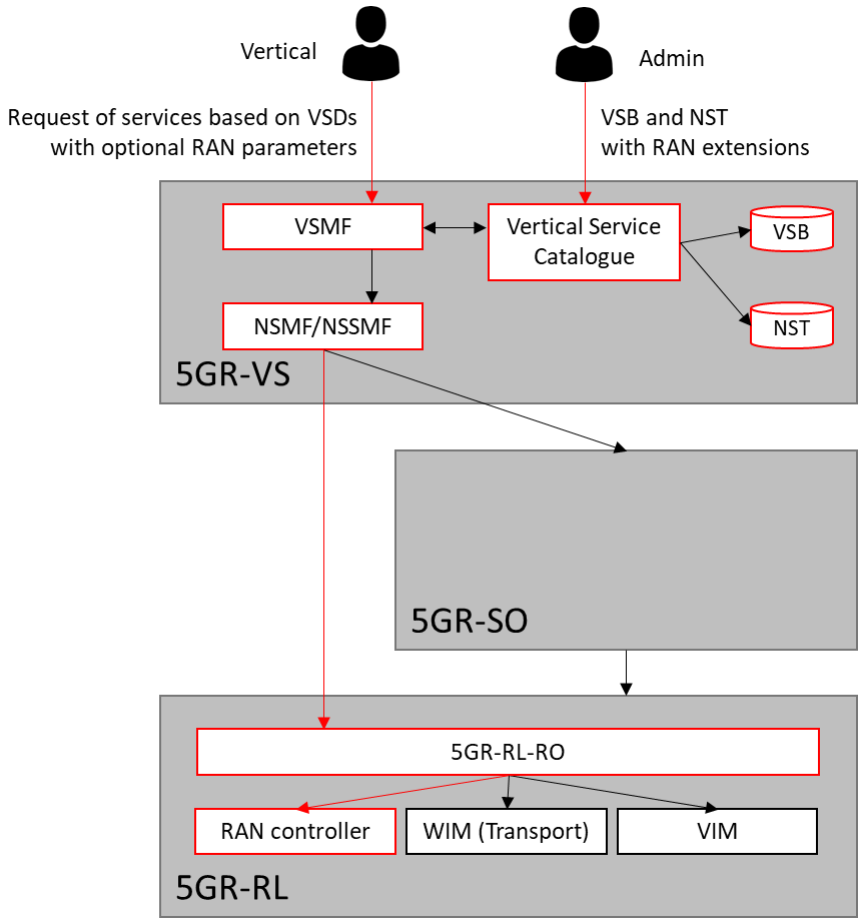

FIGURE 20: 5GROWTH ARCHITECTURE IN SUPPORT OF RAN MANAGEMENT (OPTION 2)

4.2.1.2. Innovation 2 (I2): Support of Verticals. Vertical-service monitoring

At the architectural ground, 5Growth Monitoring Platform (5GT-MP) will be based and evolve from the 5GT Monitoring Platform. For more clarity, the design of the 5GT-MP will be overviewed in high-level and innovations will be proposed as a patch to the existing architecture. More detailed description of the 5GT-MP architecture can be found in 5G-TRANSFORMER D4.3 [4].

The 5G-TRANSFORMER Monitoring Platform, as shown in Figure 21, has been implemented by integrating a Prometheus[11] backend with the 5GT-SO through the development of an adapter called Monitoring Configuration Manager (or Config Manager for short). This component acts as a unified entry point for all the configuration functionalities offered by the monitoring platform, simplifying and adapting the several APIs exposing monitoring options to the 5GT environment.

[11] https://prometheus.io

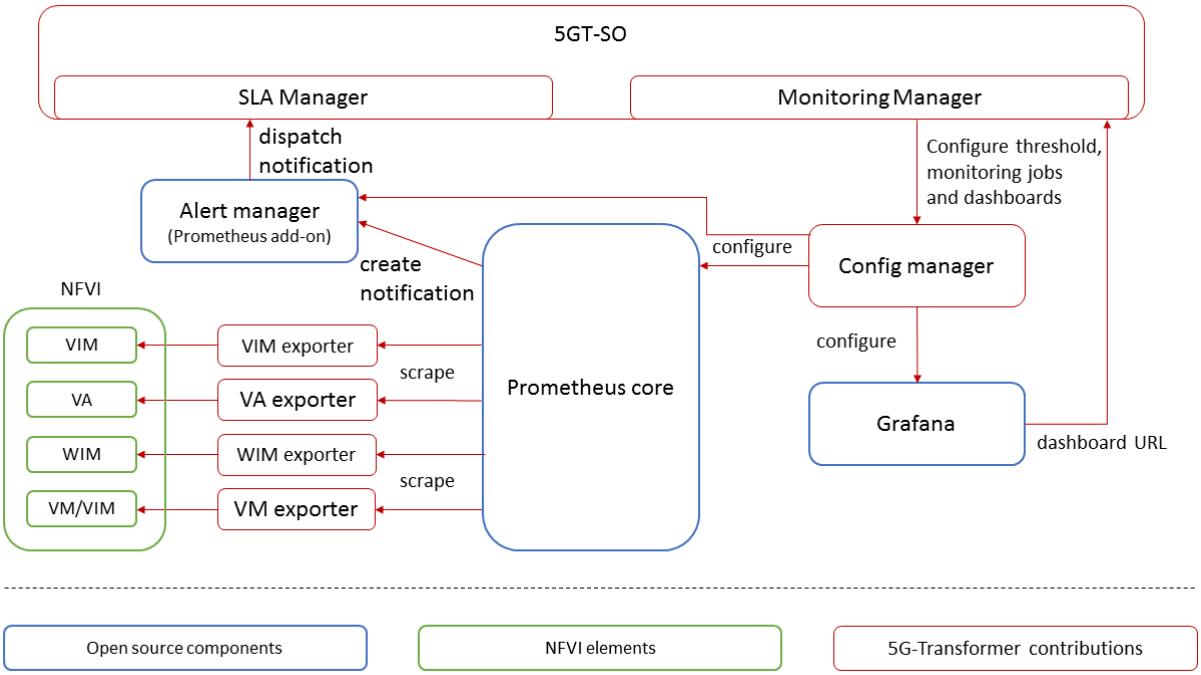

FIGURE 21: 5G-TRANSFORMER MONITORING PLATFORM ARCHITECTURE

In particular, with respect to the functional architecture of the monitoring framework reported in Figure 21, the Agents at the monitoring sources are implemented as Prometheus exporters; the Monitoring collector, the storage and the processing engine components are all implemented by the Prometheus core engine, while PromQL (Prometheus Query Language)[12] expressions take the role of the processing algorithm, aggregating and correlating elementary data as needed. The Notification dispatcher is implemented by the Prometheus Alert Manager add-on, while the Reporting component is implemented by a Grafana instance connected to the time series database (TSDB) in Prometheus core.

All the Prometheus-based monitoring platform is integrated with the 5GT-SO through the mediation of the Config Manager, a lightweight REST adapter exposing a single, simplified REST API for the configuration of the whole monitoring system (e.g. to create monitoring jobs for target monitoring parameters, or thresholds for monitoring alerts), acting as the "management" NBI of the Monitoring Platform. The Config Manager takes care of translating the platform-independent requests for monitoring-related management actions into requests directed to both Prometheus and Grafana, taking care of all the necessary configuration steps.

On the 5GT-SO side, the Config Manager interacts with the Monitoring Manager and the SLA Manager. The Monitoring Manager is the component in the 5GT-SO translating requests for high-level monitoring jobs referred to NFV service instances into low-level requests related to the monitoring of resource-level parameters, which are the data actually collected at the Monitoring Platform.

[12] https://prometheus.io/docs/prometheus/latest/querying/basics/

The SLA Manager deals instead with SLA assurance. This component is thus interested in monitoring parameters, like Key Performance Indicators (KPIs) or real-time measurements mirroring the load of the resources, that may indicate potential SLA breaches or anomalous behaviours that may lead to system under-performance. The SLA Manager interacts with the Monitoring Platform following a subscription-notification paradigm, in order to promptly react to any alert associated to the target monitoring data. In particular, the SLA Manager subscribes with the Config Manager, registering monitoring parameter queries (which may include arbitrarily complex expressions on many different time series) and threshold values for notifications. Such subscription is translated into the appropriate configuration of the Prometheus Alert Manager add-on, so that whenever the given threshold is passed, a notification is sent to the SLA Manager, which will then trigger the needed reactions at the 5GT-SO.

High-level 5Growth VoMP architecture is grounded on the existing 5GT Monitoring Platform and is shown in Figure 22.

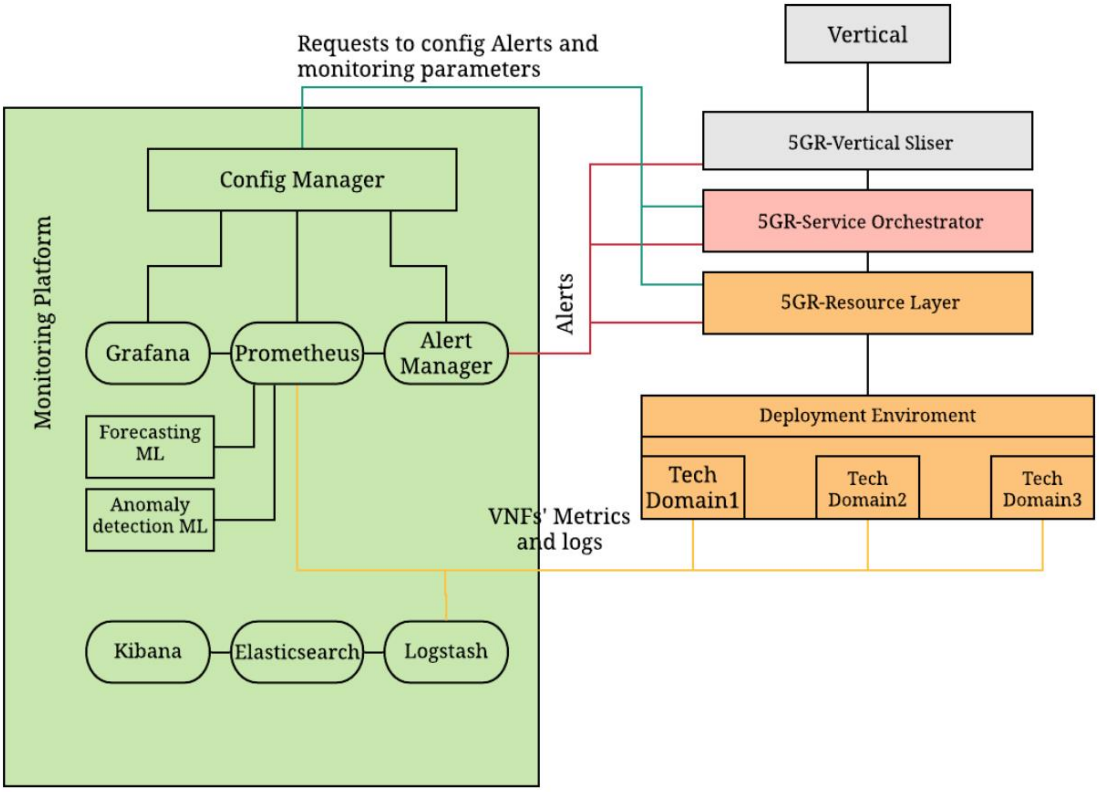

FIGURE 22: 5GROWTH ARCHITECTURE SCHEME FOR VERTICALS

As stated above, VoMS will use 5G-TRANSFORMER Monitoring Platform and its main building blocks will be: Prometheus, Alert Manager, Grafana and Config Manager.

Config Manager (CM) manages Prometheus, Alert Manager, and Grafana settings according to requests received from 5GR-Service Orchestrator (5GR-SO or SO) and 5GR-Resource Layer (5GR-RL or RL). Also, the CM handles the configuration of all MP components, monitored parameters and threshold values for notifications, so whenever the given threshold is overtaken, a notification is sent to the assigned 5Growth stack module. The configuration parameters are passed to CM

through NSD. The NSD contains the definition of parameters, their monitoring properties, thresholds and actions, activated by triggers.

The VoMS collects information from VNFs and allows data consumers (AI/ML platform, 5Gr-VS, 5Gr-SO, 5Gr-RL, etc.) to analyse system CPU consumption, memory, disk, and I/O utilization, number of sessions, etc.

The design of the logging subsystem is not defined yet. But may include components of the ELK stack[13]. For collecting logs, 5Growth may use Filebeats[14], Logstash[15], ElasticSearch[16] and Kibana[17].

Filebeat is "a lightweight shipper for forwarding and centralizing log data. Installed as an agent on your servers, Filebeat monitors the log files or locations that you specify, collects log events, and forwards them to either to Elasticsearch or Logstash for indexing."[18]

Logstash is the data collection pipeline tool. It collects data inputs and feeds them into the Elasticsearch. It aggregates all types of data from different sources and makes it available for further use.

Elasticsearch allows you to store, search and analyse big volumes of data. It is mostly used in these applications as the underlying engine for implementing search tasks functionality. It has been adopted in search engine platforms for modern web and mobile applications. Apart from a quick search, the tool also offers complex analytics and many advanced features.

Kibana is used for visualizing Elasticsearch documents and helps developers to have a quick insight into it. Kibana dashboard offers various interactive diagrams, geospatial data, and graphs to visualize complex quires. It can be used for search, view, and interact with data stored in Elasticsearch directories. Kibana helps you to perform advanced data analysis and visualize your data in a variety of tables, charts, and maps.

Despite 5G-TRANSFORMER´s powerful monitoring platform, we have identified a set of areas to improve by bring innovative features into its architecture. Such innovative monitoring schema is provided on Figure 23. The main innovation resides in the introduction of Message Queues (MQs) to 5G-TRANSFORMER´s Monitoring Platform. More specifically, different technology domains (e.g. RAN, edge, cloud, transport, etc.) are distinguished at the Resource Layer, each instance (VM or container) has an integrated agent that connects to the MQ. Potentially this schema may be extended with dedicated MQs for each domain (Tech Domain Y example on Figure 23) or even a complete Monitoring Platform (normally, different technology domains have different monitoring

[13] https://www.elastic.co/what-is/elk-stack

[14] https://www.elastic.co/products/beats/filebeat

[15] https://www.elastic.co/products/logstash

[16] https://www.elastic.co/products/elasticsearch

[17] https://www.elastic.co/products/kibana

[18] https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-overview.html

needs as shown for Tech Domain Z on the Figure 23). The monitoring information collected by agents from different technology domains is aggregated in a general MQ at the Service Orchestrator layer. Subsequently, the Alert Manager, the time series database and the analytics engine (which in 5GT are implemented using Prometheus) would obtain the information from the SO MQ, in order to further process it and generate the appropriate monitoring data.

The goal of using MQs is to have several entry points where monitoring data can be both pushed and pulled. External probes can inject information into the MQs (e.g. Prometheus probes) increasing the elasticity of the environment because different components can obtain information from the MQ without having to be limited to use specific tools.

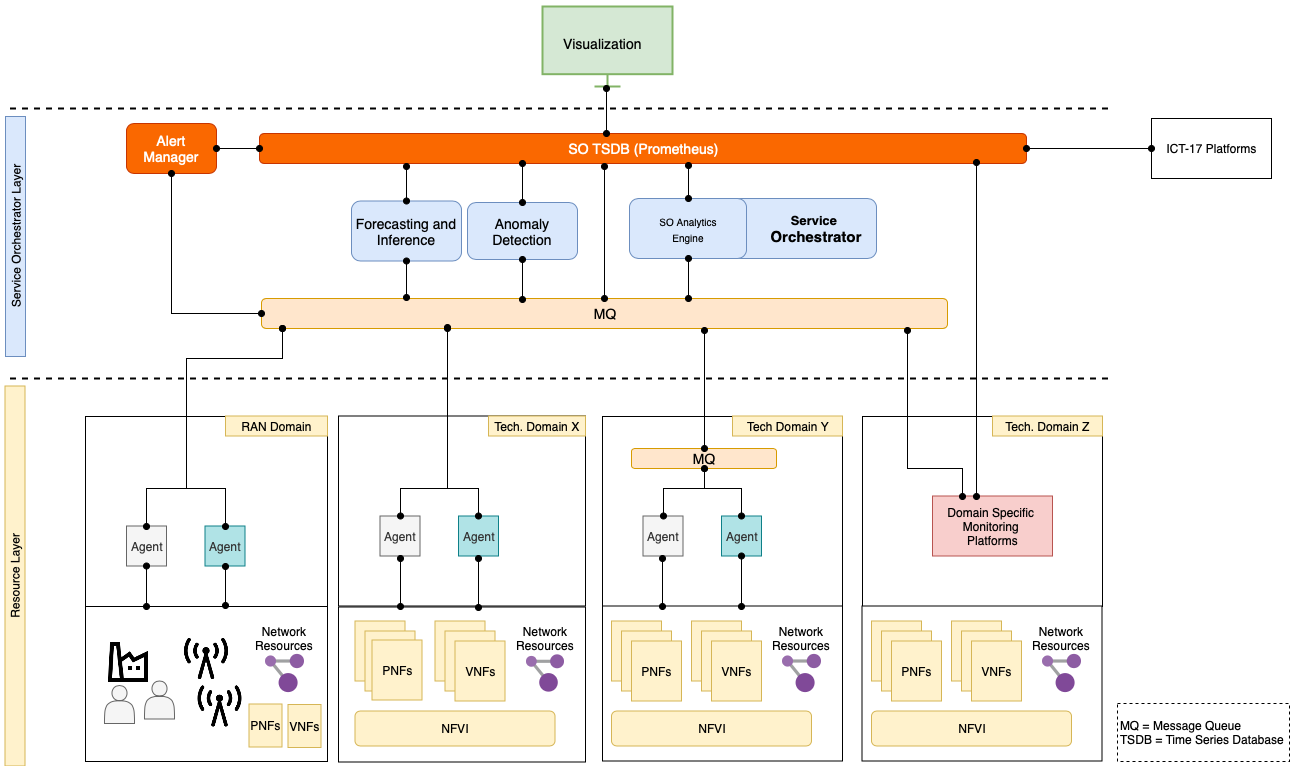

FIGURE 23: INNOVATIVE MONITORING SCHEME FOR 5GROWTH

Moreover, this solution is characterized by a high scalability since the MQ instances may be spawn individually for the domains, or the entire monitoring system instance may be dedicated to the particular Tech Domain. If needed, creating new MQs and add them to the stack is straightforward and does not increase the complexity of the architecture.

Approaches for monitoring VMs and containers are different. The flow of the monitoring data from the VM instances to the Prometheus is shown on Figure 24.

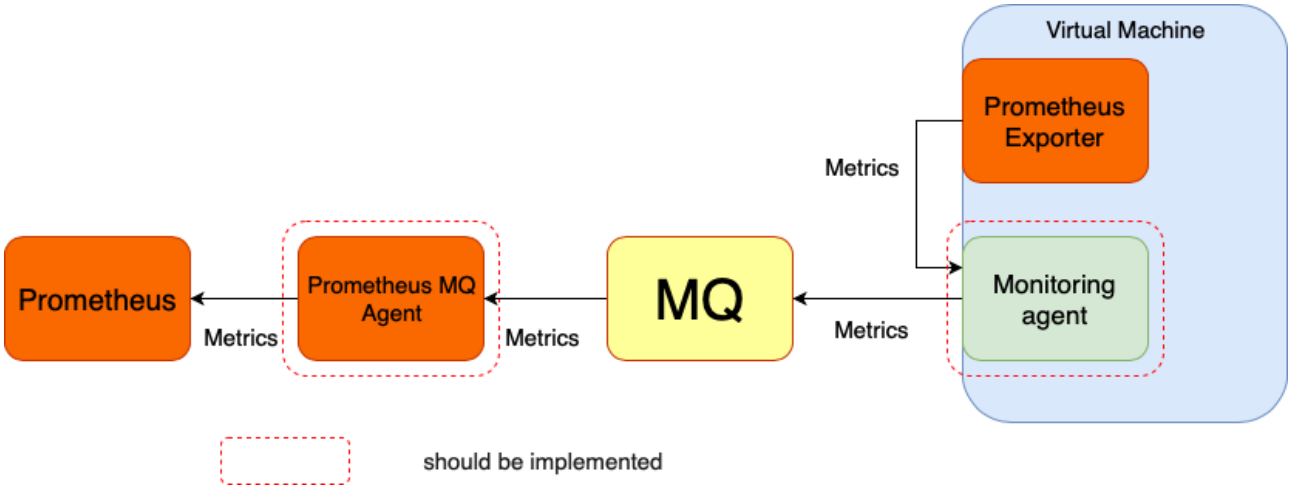

FIGURE 24: METRICS FLOW FROM VM TO TSDB

The VM case monitoring procedure is the following: Prometheus exporter and Monitoring agent are placed on the VM instance. Prometheus Exporter gathers metrics data and expose it through an HTTP service. A monitoring agent gets metrics from a Prometheus Exporter through HTTP and push them into the MQ. On the other side, MQ is listened by the Prometheus MQ agent and metrics get pulled out and placed into Prometheus TSDB once they appear in the queue.

The procedure for monitoring container instances will differ on the client side (at the right-hand side of the MQ in the figure). In the case of containers, the monitoring agent and Prometheus exporter should be placed outside of the container and could be implemented as a side car container, for example. The monitoring procedure for containers should be developed.

Finally, MQs can be used as an interface for the exchange of information between different technologies and components of the architecture. In this way, internal and external components (e.g, ICT-17 projects, federated domains, etc.) can read/publish information in a common manner, avoiding the definition, creation, and implementation of new APIs. As a result, it will be easier for dynamic probes to collect and publish metrics on demand, and during a particular period of time. Dynamic instantiation of the monitoring probes is described in Figure 25. The figure shows the VM case. The Config Manager configures Monitoring Agent through MQ. Once Monitoring Agent receives the management signal from the Config Manager it may setup and configure Prometheus Exporter and FileBeat. FileBeat as a service appears to implement logging feature. It gathers logs from the instance (VM in particular case) and pushes them to logStach through MQ.

The main storage component for storing monitoring data in the Monitoring Platform is Prometheus, which acts as TSDB. Prometheus server has the necessary functionality to collect metrics from domains and has the possibility to collect metrics from child Prometheus servers as well, which is important for implementing dedicated domain monitoring instances. In the case of federation, Prometheus can enable filtering to retrieve only needed information from federated domain Prometheus servers. For integration purposes, Prometheus has an API to provide information to other modules such as Anomaly Detection, Forecasting and Inference or Alert Manager. These analytical blocks shall receive data from Prometheus, process it and pass it to 5Gr-SO to decide what action should be executed.

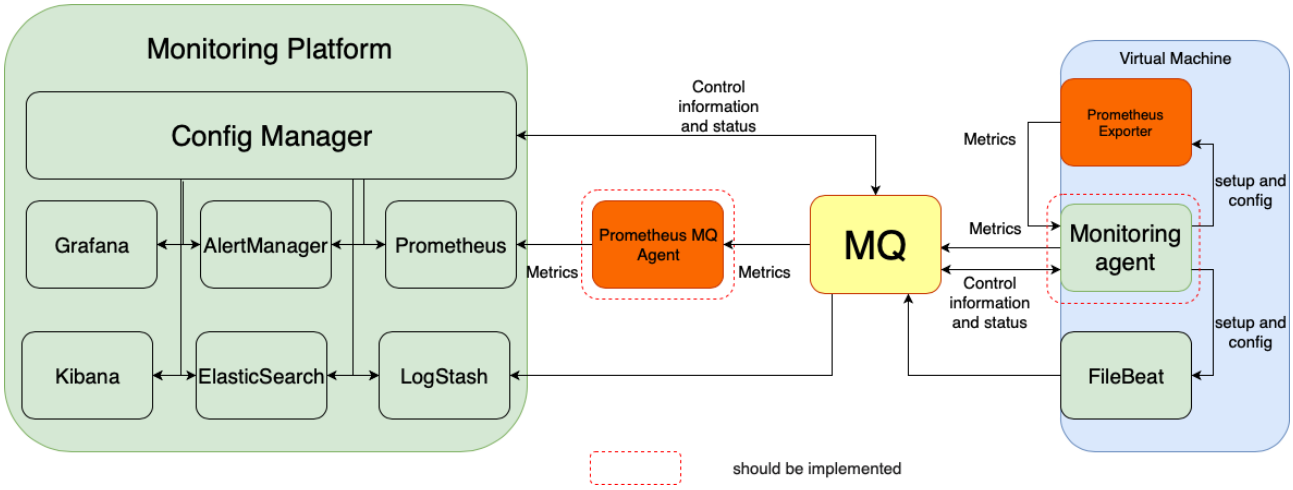

FIGURE 25: DYNAMIC INSTANTIATION OF THE MONITORING PROBES

4.2.1.3. Innovation 3 (I3): Monitoring orchestration

Figure 26 defines the first proposal of the architecture for this innovation. It illustrates the main components, the connections between them and where they can be further integrated into the 5Growth architecture. The architecture will be refined and more detailed in subsequent phases of the time plan for the design and implementation of the innovation.

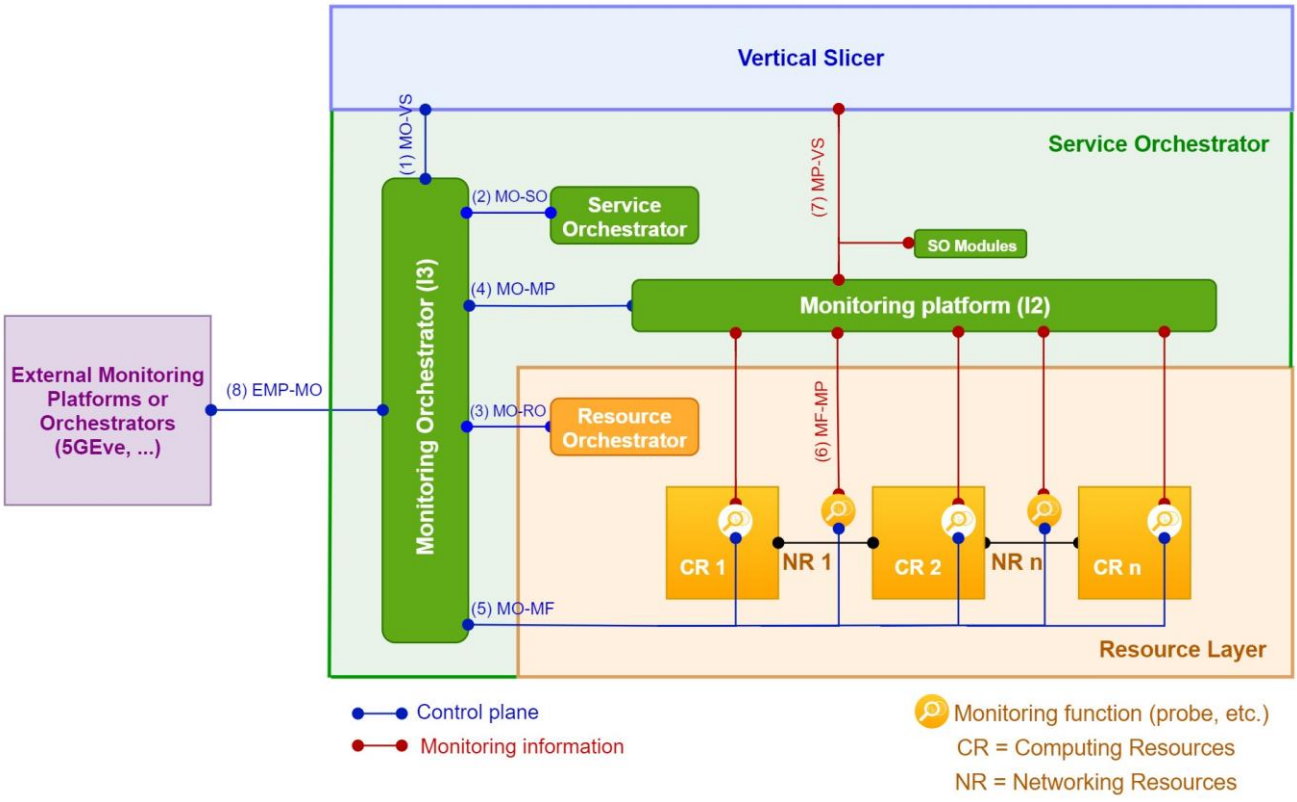

FIGURE 26: ORCHESTRATION OF MONITORING FUNCTIONS REFERENCE ARCHITECTURE

TABLE 6: ORCHESTRATION OF MONITORING FUNCTIONS ARCHITECTURE COMPONENTS

Component	Description
Monitoring Orchestrator (MO)	Receives monitoring requests from the 5Gr-VS and the 5Gr-SO, manages the Monitoring Platform lifecycle, auto-discovers monitoring probes and optimally places monitoring functions based on information about the 5Gr-RL, provided by the RO; and also, the information about the NSDs, provided by the SO.
Monitoring Platform (MP)	Manages and aggregates the Monitoring Data gathered from the Monitoring Functions for further processing and visualization. The MP is composed by some services, like MQ, TimeSeriesDB, visualization services, etc. The lifecycle of these services must be dynamically managed by de Monitoring Orchestrator.
Service Orchestrator (SO)	Provides the Monitoring Orchestrator (MO) with the location of the different Resource Orchestrators that oversee the resource domains (RAN, transport, etc.) where the monitoring functions will be placed. Furthermore, the 5Gr-SO provides information about the Network Services that will be monitored, in order to calculate the optimal placement of Monitoring Functions.
Resource Orchestrator (RO)	Manages the Resource Layer. Abstracts a global view of the resources to the Monitoring Orchestrator. Furthermore, the RO may also participate in the final placement of MFs when a more granular and detailed view of the physical resources is required for an optimal placement decision.
Monitoring Functions (MFs)	Used to gather monitoring data and metrics. (Probes, logs, etc.)
External Monitoring Platforms or Orchestrators (EMP)	ICT-17 monitoring and/or orchestration platforms (5GEVE, 5GVINNI, etc.) (Federation interface).

TABLE 7: ORCHESTRATION OF MONITORING FUNCTIONS ARCHITECTURE INTERFACES

Interface	Description
EMP-MO	Used by Monitoring Orchestrator to gather information from External Monitoring Platforms. This interface facilitates the integration with external platforms to support monitoring for multi-domain and federation use cases.
MO-VS	Connects the Monitoring Orchestrator and Vertical Slicer, allowing the 5Gr-VS and 5Gr-SO to dynamically request the Monitoring Orchestrator to monitor specific 5G services or slices.
MO-MP	Connects Monitoring Orchestrator to Monitoring Platform. The MP opens this interface for the Monitoring Orchestrator for instantiating/provisioning monitoring functions by the Monitoring Orchestrator.
MO-MF	Used by the Monitoring Orchestrator to manage the Monitoring Functions, Life Cycle Management and metadata aggregation.
MF-MP	Used by the Monitoring Platform to receive raw monitoring data from Monitoring Functions.
MP-VS	Expose the monitoring information and results to the Vertical Slicer.
MO-SO	Used by Monitoring Orchestrator to discover the ROs available to contact for

	a finer granularity on physical resources view. Also, it is used to know the placement of the other VNFs that compose a Network Service. Moreover, the MFs need to be attached to the NS´s virtual links, and the mapping of these virtual links to the 5Gr-RL virtual networks is available at the 5Gr-SO.
MO-RO	Used by Monitoring Orchestrator to decide where to place the monitoring functions in the physical/virtual infrastructure.

The normal operation of the orchestration of monitoring functions requires a series of interactions between the architecture components presented above, which are described in the following workflow:

1. A vertical request to monitor certain network slices and services, along with a set of KPIs to visualize.
2. The 5Gr-VS forwards to the MO an identifier of the services/slices to monitor. (MO-VS)
3. The MO requests to the SO the locations of the relevant ROs to monitor the requested service, identifying the ROs of the different domains where the monitoring functions will be placed (RAN, transport, etc.) (MO-RO interface)
4. The MO requests to the RO the chain of computing, storage and network resources that compose the services/slices to monitor. (MO-RO)
5. The MO calculates the type of monitoring functions that need to be placed in the different physical resources.
6. The MO deploys the monitoring functions on the MP, from which the MP is going to aggregate the information later on, e.g. creating a data bus with specific monitoring functions as publishers (MO-MP).
7. The MO places the different monitoring functions in the specific locations calculated in 4. (MO-MF)
8. The monitoring functions send the raw data and metrics retrieved from the monitored element to the MP (MF-MP).
9. The MP aggregates the information received from the monitoring functions, processes the information and sends the results back to the 5Gr-VS for the visualization (MP-VS) and/or to some module in the SO that leverages the monitoring results for SLA enforcement, etc.

4.2.1.4. Innovation 4 (I4): Control and Management. Control-loops stability

It is envisioned that the closed-loop not only takes place inside each of the layers of the 5Growth stack to carry out their internal automated control of service and/or resources, but also across the different layers and different administrative domains which interact with each other according to predefined SLAs, policies/rules, mapping models, etc. At the architecture level, the control loops can be placed per layer (see Figure 27), across layers (see Figure 28), and across domains (see Figure 29). In the 5Gr-RL layer, we can even have closed-loop per technological or network domain. To enable these closed-loops, different interactions may be needed at the architecture level: (i) interaction with the monitoring platform; (2) interaction with the AI/ML platform; (3) interactions between the layers within a single administrative domain; and (4) interaction between the administrative domains, in case of federation.

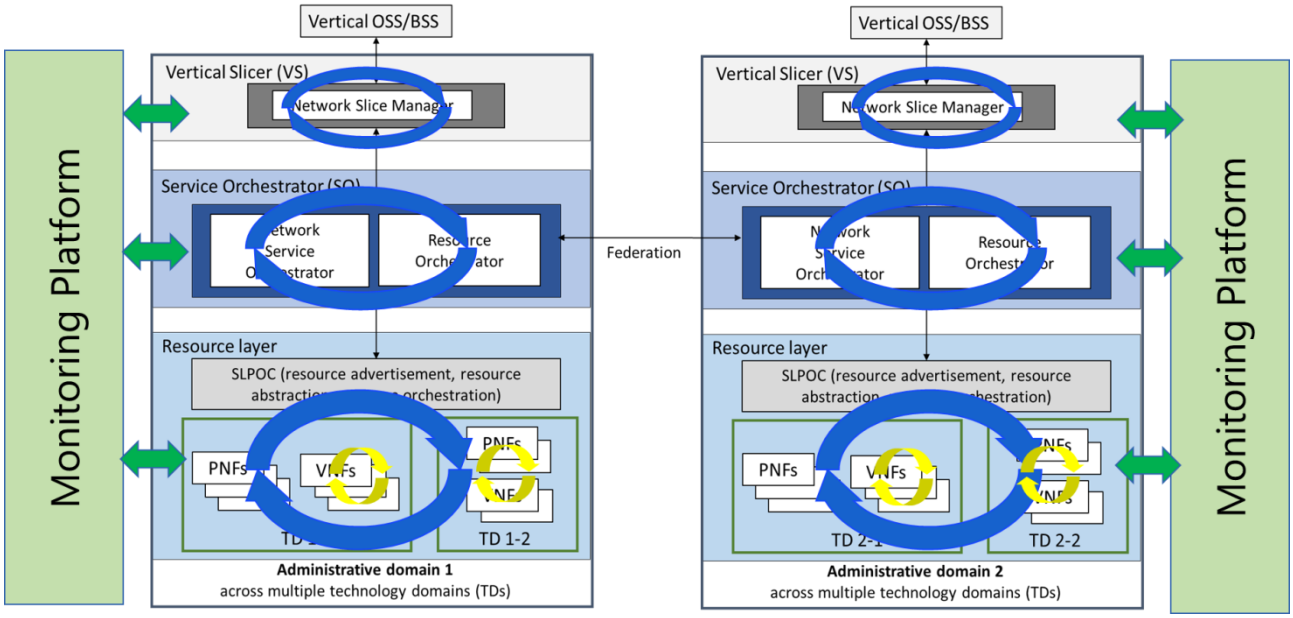

FIGURE 27: CONCEPTAUL ARCHITECTURE OF THE CLOSED LOOPS PER LAYER

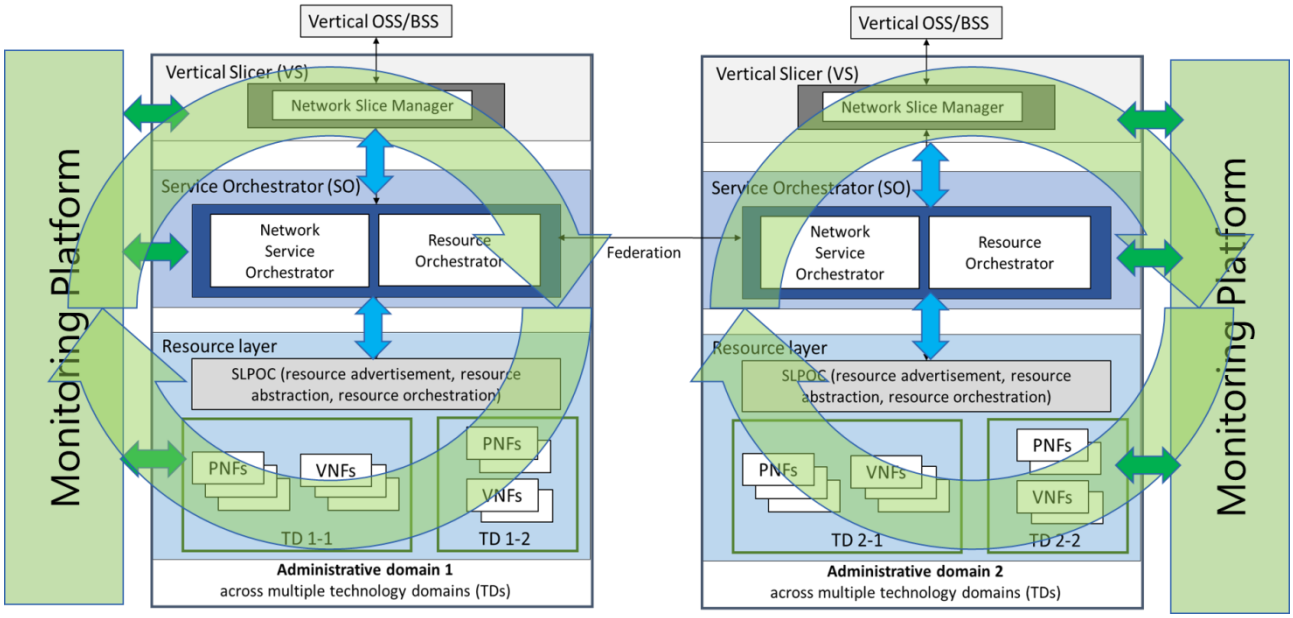

FIGURE 28: CONCEPTUAL ARCHITECTURE OF THE CLOSED LOOPS ACROSS LAYERS

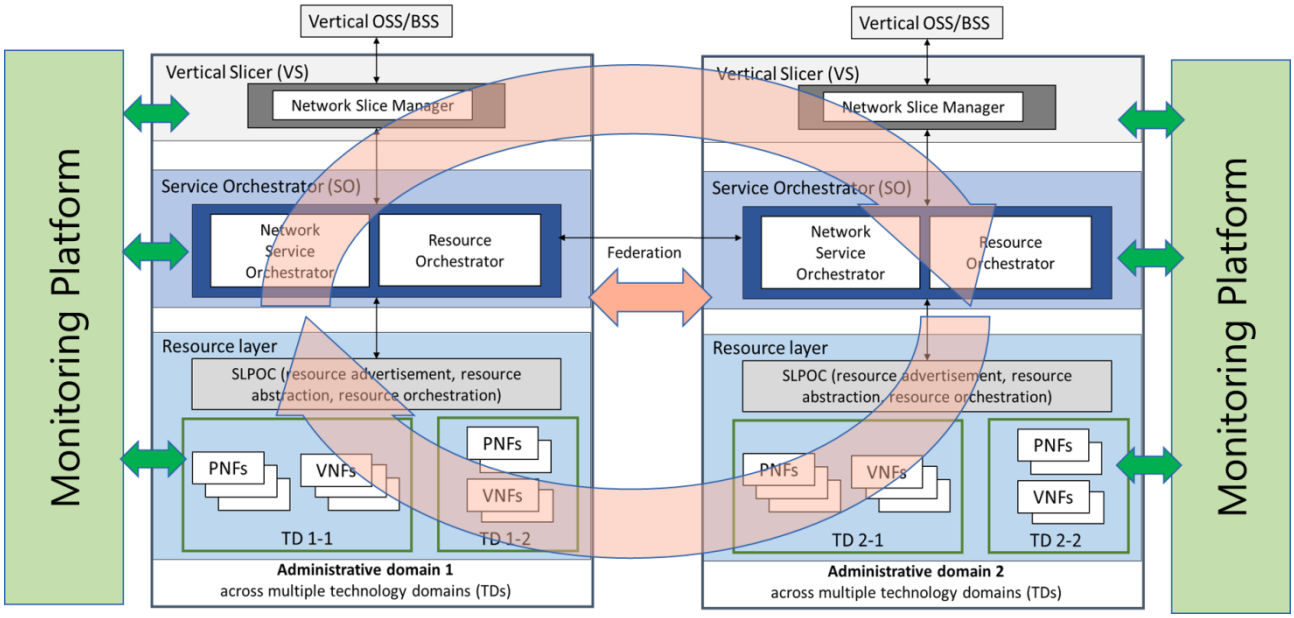

FIGURE 29: CONCEPTUAL ARCHITECTURE OF THE CLOSED LOOPS ACROSS ADMINISTRATIVE DOMAINS

To enable a stable and closed-loop through the system, it is essential to design the 5Growth architecture along with novel mechanisms/models, and define additional entities within each of the building blocks, as well as specify new interfaces across the different layers. This will enable the required communication and interactions between the layers for closing the control loop in the system as well as ensuring interactions between administrative domains.

To this aim, 5Growth will explore the following dimensions.

Vertical Slicer (VS) Control Loop

- Enhance the *SLA & Policy Management module* inside the 5Gr-VS to manage the SLAs at the vertical service level, and derive related policies for the SLA control throughout the system, including storing SLAs (business and technical) and handling SO Control-loop manager events, etc.
- Enhance the *Arbitrator module* for arbitration across vertical services SLAs.
- Enhance *Translator module* for appropriate SLA definitions and translations between business (at the service level) and technical SLAs (at the network level).
- Add *new interface* to the AI/ML platform, which provides recommendation or inputs (like predictions of traffic demand, etc.), to the *SLA & Policy Management module* to make decisions.
- Add *new interface* to the monitoring platform, which provides monitoring data, telemetric, and real-time data analytics for notification of events and alarms, to the *SLA & Policy Management module* to make decisions.
- Enhance the *VS NBI* interface to enable interactions with the vertical customers (e.g. alert the vertical whenever SLAs cannot be enforced automatically and manual intervention is required), as well as with the monitoring platform to allow vertical -or VS- driven service scaling.

- Enhance the *VS SBI* interface with the SO Control Loop Manager, which will transform SLAs into network service SO control loop policies (scaling, self-healing, re-configuration, etc.) and technical requirements.

Service Orchestrator (SO) Control Loop

- Enhance the SO *SLA manager* module to ensure the E2E network service SLAs:
  - Develop SLA-driven service management automation mechanism (based on the design from I8: Smart Orchestration and Resource Control).
  - Calculate Network Service KPIs from monitoring platform data (input: I2, Section 4.2.1.2)
  - Store SO control loop network service policies and KPIs.
  - Enforce the correct policies to each technological domain 5Gr-RL orchestrator, or even take network service federation decisions.
- Enhance the *SO EBI/WBI* interface with the peering SOs for E2E network service SLA management.
- Enhance the *SO NBI* to notify vertical service control loop manager whenever network service requirements cannot be satisfied with current installed policies.
- Add *new interface* to the AI/ML platform, which provides recommendation or inputs (like predictions of traffic demand, detecting anomalies or root causes, etc.), to the SO *SLA manager* module to make proper decisions.
- Enhance the *interface* to the monitoring platform, which provides monitoring data, telemetric, and real-time data analytics for notification of events and alarms, to the SO *SLA manager* module to make decisions.

Resource Layer (RL) Control Loop

- Per-domain Loop
  - WIM resource management through SDN for RAN and network domain (transport and/or edge/core), including
    - Programmable traffic management module, responsible for injecting control behavior to a programmable data plane to support vertical requirements (within and across slices), behavior learning, and stability control.
    - Data plane customization per slice is required to support network level KPIs/ SLAs via real time control and traffic management, for example by means of Virtual Queues for Performance Isolation and Traffic Management per slice.
    - Data plane arbitration entity and related control interfaces to handle arbitration (e.g., scheduling) of data-plane traffic across services (slices).
  - VIM resource management within a data center.
- Across-domain Loop to provide E2E resource control across (radio) access, transport, core and cloud domains, if applied. Here the 5Gr-RL control loop manager for every technological domain (RAN, Edge, Transport, Core, Cloud), may interface with the E2E domain orchestrator. Notice that sometimes the 5Gr-RL control loop manager can be part of the E2E domain orchestrator.

- For the above both loops, we need to define new interface(s) to the AI/ML platform, which provides recommendations or inputs (like predictions of traffic demand, detections of anomalous behaviors, etc.), to the VIM/WIM controllers to make proper resource allocation/update decisions.

Interaction between VS-SO-RL layers is needed to ensure the closed-loop throughout the whole 5Growth system across different layers. The interactions are specified in the design of the related NBI/SBI interfaces and related information model between the layers. This may include the definition of SLA modelling through the system, defined policies or rules (e.g. auto-scaling rules), and static mapping tables/models agreed between the layers (e.g. service level to resource level) according to the predefined SLAs between the layers.

Interaction between SO-SO is needed to ensure the closed-loop between the administrative domains for the End-to-End service LCM. The required interactions will be specified in the extension of the 5Gr-SO EBI/WBI interfaces. This may include the automatic process for dynamic network service on-boarding, provisioning and update (e.g. scaling) according to the agreed SLAs and services offered according to the signed contracts between the administrative domains.

Stability control needs to be provided in the system. It can be achieved through a proper design of the SLA control module at each layer and the close interactions between the layers; the design of smart control mechanisms defining related thresholds, control actions based on policies and rules, and right access of data which is fed to the different control-loops in 5Gr-VS, 5Gr-SO and 5Gr-RL is available at a different volume, variety, veracity, and volume.

4.2.1.5. Innovation 5 (I5): Control and Management. AI/ML Support

The purpose of the AI/ML platform is to support different layers and applications in the 5Growth architecture when they are running algorithms that require training from external data. As shown in Figure 30, the platform interacts with the layers in the 5Growth architecture, and with the Monitoring Platform. It is assumed that a layer may want to use its own algorithms to execute decisions. It thus provides the specificities of the algorithm to the AI/ML platform, requesting from it a mathematical model (specific for the algorithm it wishes to use). The requesting entity may not expect an immediate response from the AI/ML platform, since the latter operates on a different time scale. Indeed, the returned model must be generated from training data (5Growth Task 2.3) to which the AI/ML platform needs to have access through the Monitoring Platform by establishing a feedback loop (the AI/ML receives from the MP training data as well as performance metrics that can be used to compute rewards used by the AI/ML algorithms). Note that the MP is not necessarily a monolithic block as shown in Figure 30, but could be split into functional blocks related to each layer. The training data should obviously be meaningful to the layer for which the algorithm is being trained, and it should be specified through some abstraction or internal representation shared with the Monitoring Platform. The AI/ML platform thus outputs to the requesting layer a model (e.g., number of layers in a neural network, its internal connections, etc.) that aids the decision processes of that layer, such as a policy, or an orchestration decision by the 5Gr-SO or 5Gr-RL, Finally, it is to be remarked that AI/ML not only interacts with 5Gr-VS/5Gr-SO/5Gr-RL, but may also provide service to any applications or forecasting platform (such as the

one foreseen in I10, Section 4.2.2.3). So, in Figure 30, the AI/ML platform is shown as offering services through its NBI to other components (VS/SO/RL, forecasting, applications etc.).

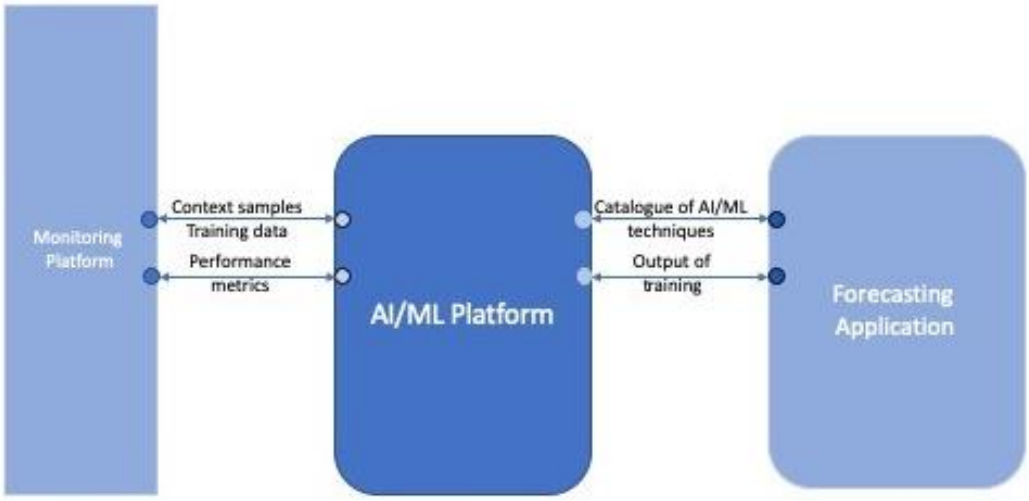

FIGURE 30: 5GROWTH ARCHITECTURE INTEGRATING THE AI/ML PLATFORM

It is thus envisioned that the AI/ML Platform will expose the following NBI, which can be used as in the example interaction shown in Figure 31:

- Catalogue of AI/ML techniques
- Output of the training

The AI/ML will also need Interfaces toward MP for the exchange of:

- Context Samples and Training data
- Performance metrics

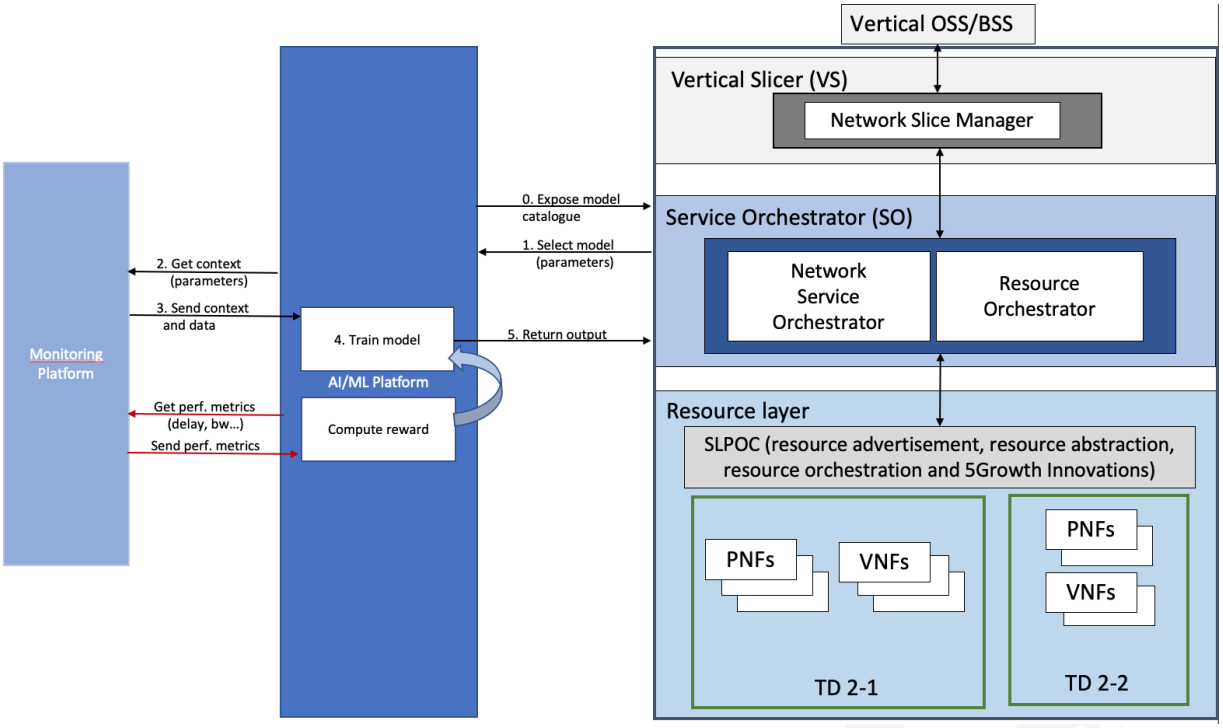

FIGURE 31: EXAMPLE INTERACTION BETWEEN PLATFORMS AND LAYERS

4.2.1.6. Innovation 6 (I6): End-to-End orchestration. Federation and Inter-domain

The work related with federation/inter-domain is a key part of the 5Growth architecture discussion and, as such, it is part of T2.1. However, it will also have implications in the rest of tasks of WP2.

Figure 32 presents the three architecture building blocks of 5Growth (5Gr-VS, 5Gr-SO, 5Gr-RL) that may be involved in federation/multi-domain relations with other providers. The monitoring building block will also be affected, but we focus on these three for the initial design. All possible options under consideration are depicted in Figure 32. They include service and resource federation and other possible multi-domain interactions.

From a federation/multi-domain perspective, 5Growth service provider domains can be classified into consumer and provider domains. The 5Growth service provider receiving the request from the vertical (leftmost one in Figure 32) becomes the only one interacting with it and having a complete view of the E2E vertical service. In case federation or multi-domain support is needed, this 5Growth service provider will become the consumer domain of other provider domains (rightmost ones in Figure 32). The available options are explained below.

On the one hand, the term federation is used when the multi-domain interaction occurs between entities that have exactly the same functionality in the consumer and provider domains. When the Communication Service Federation Function/Communication Service Management Function (CSFF/CSMF), which is part of the 5Gr-VS, interacts with other CSFF/CSMFs, we refer to communication service federation. When service federation is offered through inter-SO interaction, we refer to NFV network service federation. On the other hand, we refer to Hierarchical multi-domain service support when one entity that is higher in the consumer domain stack hierarchy interacts with another entity lower in the stack of the provider domain, in this case, consumer (Communication Service Management Function) CSMF to provider NSMF (Network Slice Management Function).

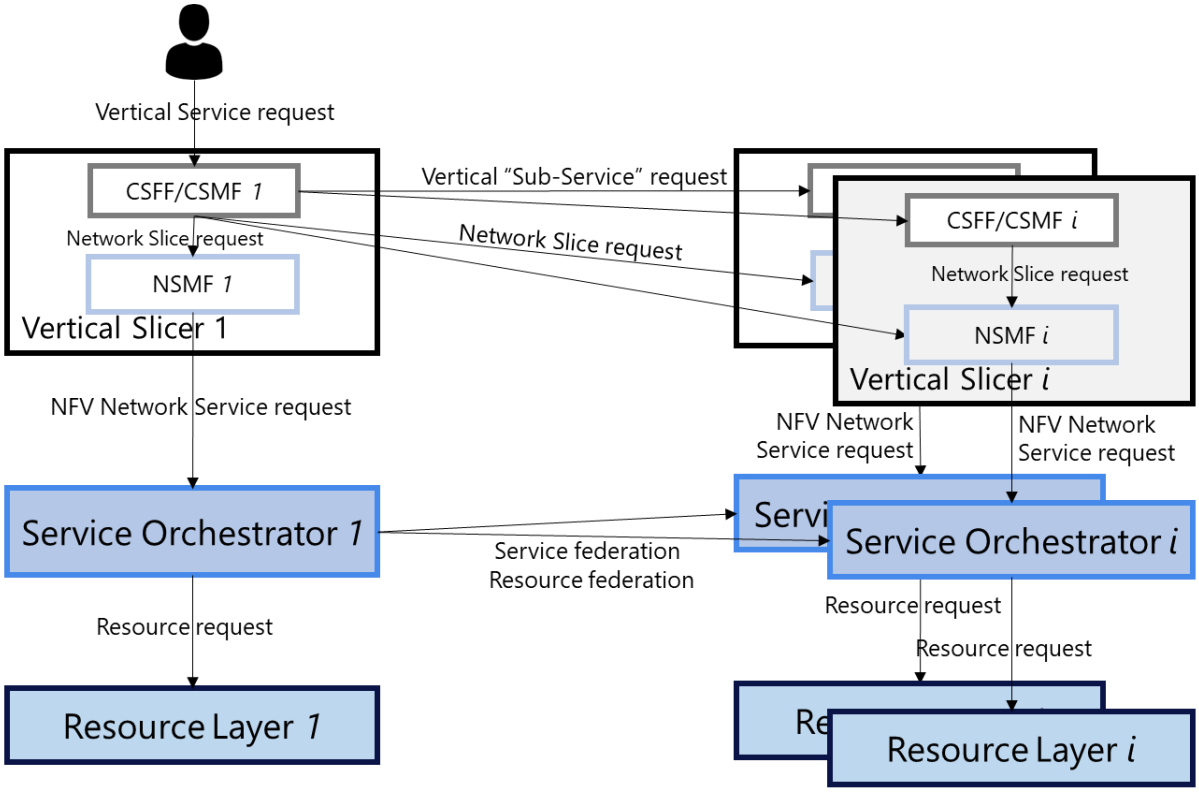

FIGURE 32: FEDERATION/MULTI-DOMAIN ARCHITECTURE.

Service Federation

Two service federation options are considered.

Communication service federation

In this case, communication service is understood according to the 3GPP notation [1] and refers to service requested by the vertical. In this option, an extended Communication Service Management Function (CSMF) of the consumer domain is needed, since the interaction with the peer domain should be added (this could be referred to as Communication Service Federation Function, or CSFF). This function, possibly based on the vertical request, decides how to split the communication service into subservices and what subservice must be requested to the CSFF of the respective provider domains. In the figure, this consumer-provider interaction corresponds to the Vertical sub-service request arrows.

In this option, the CSFF is in charge of managing multi-domain support, since it is the only building block in the architecture with knowledge of the E2E vertical service. This also implies that resource layer information for stitching the vertical sub-services (if needed) is handled at the CSFF. Advertisement of vertical (sub-)service blueprints (VSB) is also required so that potential consumer domains are aware of the communication service offerings of provider domains.

Furthermore, the CSFF-to-CSFF interface will mimic the vertical-to-CSFF interface, since the vertical subservice request will follow the same information model with enhancements of the VSBs towards federation support.

NFV network service federation

In this model, the vertical is assumed to focus on the vertical service logic and fully delegates service deployment to the 5Growth service provider as long as the SLAs are respected. Therefore, the vertical slicer will gather the requirements and vertical service topology from the vertical and map it into a network slice, which will eventually translate into an NFV network service request towards the service orchestrator (SO). The SO acts as an NFVO and, as such, comprises service and resource orchestration functionalities.

In this option, the NFV network service (NFV-NS) request may come in the form of a composite network service and the 5Gr-SO will deploy each nested network service in one provider domain by interacting with provider 5Gr-SOs. Alternatively, the NFV-NS request may come as a regular network service and the 5Gr-SO itself may decide how to split it into nested NFV-NSs and request their deployment to provider 5Gr-SOs. The interface to request nested NFV-NSs follows the same information model as that between the 5Gr-VS and the consumer 5Gr-SO with modifications for federation support.

After all nested NFV-NSs are deployed, the consumer 5Gr-SO is in charge of stitching them into a composite E2E NFV-NS by exchanging service and resource information with the peer 5Gr-SO and by requesting the allocation of paths to the underlying resource layers. That is, the consumer 5Gr-SO is in charge of resource allocation at the consumer domain resource layer and the provider 5Gr-SO is in charge of resource allocation at the provider domain resource layer under the coordination of the consumer domain (e.g., selection of IP address ranges).

Hierarchical multi-domain service support

As in the Communication service federation option described above, the CSMF of the consumer domain is the only building block with an E2E view of the vertical service. However, in this case, there is a hierarchical relationship with the NSMFs of provider domains, since what the CSMF requests to provider domains are parts of the E2E network slice, which, from the point of view of the provider domain, are requested as network slices (and not communication services).

In the same way as in the other option, the CSMF is in charge of managing multi-domain support, since it is the only building block in the architecture with knowledge of the E2E vertical service, and so, also deciding in what provider each portion of the end-to-end slice is deployed. This also implies that resource layer information for stitching the vertical sub-services (if needed) is handled at the CSMF. However, the advertisement required in this case is for network slice templates so that potential consumer domains are aware of the network slice offerings of provider domains. Furthermore, the CSMF-to-NSMF interface will have to support such interaction.

Resource Federation

In resource federation, there is an initial phase during which consumer domains advertise/discover resource availability (computing, storage, network) of peering domains. Based on this information, the consumer domain decides what resources from what provider domains to use and generates the corresponding NFVIaaS service requests. After completion of the corresponding resource allocation procedure, the control of provider-domain resources is fully delegated to the consumer

domain, which handles them as if they were consumer domain resources. Therefore, any network function or network path could be deployed in the provider domain in the same way it is deployed in the consumer domain.

In 5Growth, resource federation is handled at the service layer (i.e., among service orchestrators). The following functionality is involved. First, resource orchestrators in each domain request resource availability information from the local resource layer and this information is stored in the NFVI resource repository of the service orchestrator. Second, each service orchestrator applies the resource abstraction procedures to the resource information it plans to offer to other domains of the federation. In this way, it selects the appropriate exposure level according to domain policies. Third, the consumer domain selects the resources from the domains it wants to use and requests them in the form of and NFVIaaS request. Fourth, this request is handled by the provider domain, abstracted and sent to the local 5Gr-RL to allocate the resources. Finally, the resource federation will migrate the management and control of the selected resources to the consumer domain so that these resources can be fully handled from the consumer domain while the resource federation agreement is in place. As a result, when a service request is received in the consumer domain, the resource orchestrator of the service orchestrator handles federated resources in the same way as local ones. Therefore, these resources are also provided to the placement algorithm, which can place VNFs there to be connected with VNFs located in consumer-domain resources.

Transversal techniques: DLT-based federation

In each federation case, multiple domains interact between each other in different stages of advertising, discovery, negotiation, establishing connections and usage. To that end, 5Growth will also explore Distributed Ledger Technologies (DLT)-based techniques that are transversal to all above federation options in the sense that they offer an efficient way to advertise and discover services or resources as well as dynamically negotiate new terms (i.e., on-the-fly) (Figure 33). DLTs (e.g., Smart Contracts [63] on top of permissioned Blockchain) establish Distributed Applications (DApp) for advertising, discovery and negotiation of all vertical service blueprints, network slice templates, NFV network services and resources.

The solution provides all domains with an updated view of offerings or pending requests for federation, in addition to basic features, such as discovery, advertising, or negotiation, can also provide more advanced ones, such as broker-role, dynamic pricing, applied abstractions, etc.

Using the DLT-based solution, the consumer domain entity, in charge of the E2E service, decides to what provider domains it should request services or resources.

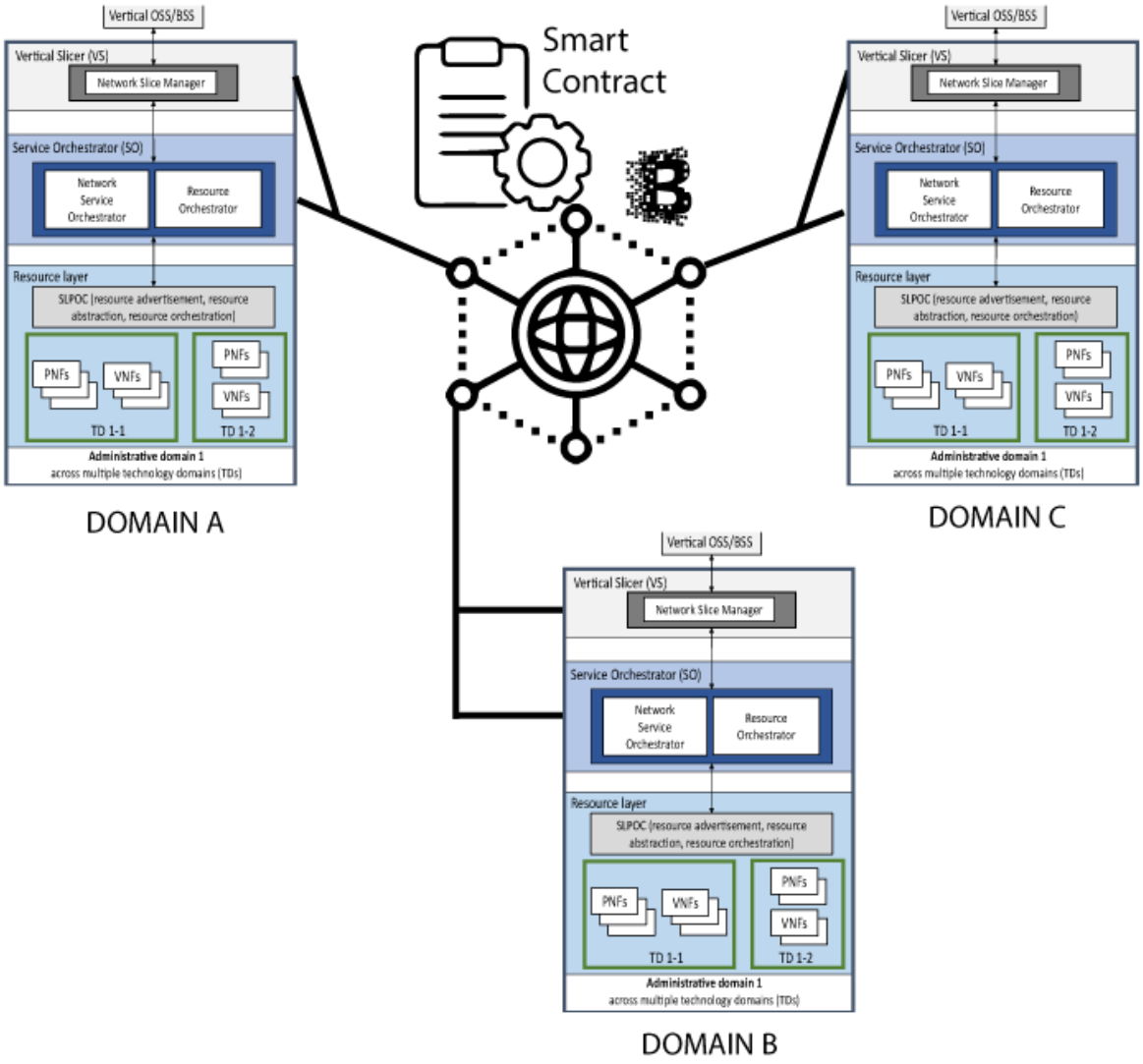

FIGURE 33: DLT-BASED FEDERATION

4.2.1.1. Innovation 7 (I7): End-to-End orchestration. Next Generation RAN

As mentioned in Section 3.2.7, 5Growth will pay special focus on O-RAN's[19] reference architecture, shown in Figure 34) to support novel open, AI-powered RANs. In turn, Figure 35 shows how O-RAN reference architecture can be integrated into 5Growth main architecture.

[19] https://www.o-ran.org/

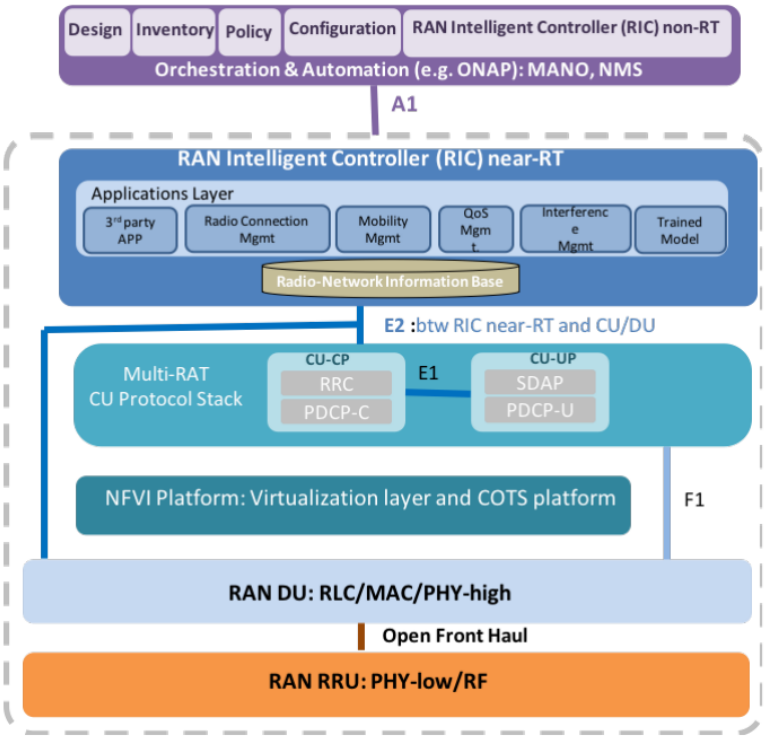

FIGURE 34: O-RAN ALLIANCE REFERENCE ARCHITECTURE [18]

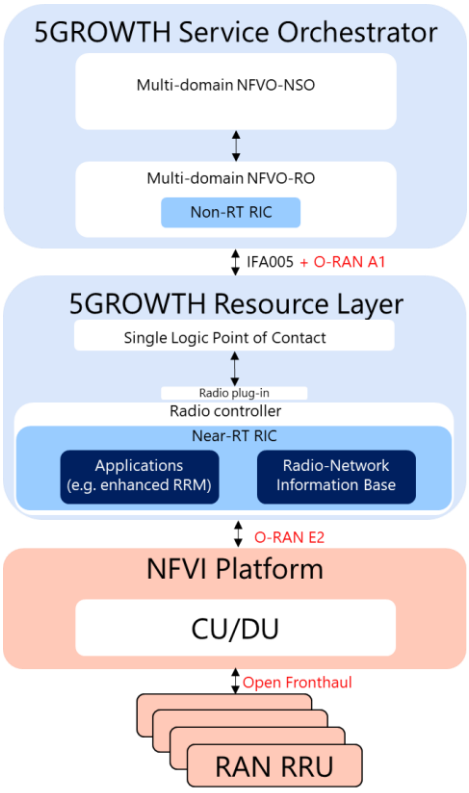

FIGURE 35: 5GROWTH ARCHITECTURE EXTENSIONS TO SUPPORT NEXT-GENERATION RADIO ACCESS NETWORKS

The main new functional blocks are the following.

- Non-RT (Real-Time) RIC (Radio Intelligent Controller). This RIC seats in the Orchestration and Automation layer – the highest layer of the O-RAN stack. This functionality can be part of 5Growth Service Orchestrator and makes decision at the second-level granularity (>1s). Non-RT RIC functions include service and policy management, RAN analytics (relying on 5Growth monitoring platform) and model-training (relying on 5Growth´s AI/ML platform) for the near-RT (<1s) RAN functionality. Trained models and real-time control functions produced in the non-RT RIC are distributed to the near-RT RIC for runtime execution.
- Near-RT RIC: Near-RT RIC is the highest control entity within the eNB/gNB (Resource Layer) and provides Radio Resource Management (RRM) functionality with embedded intelligence (optionally accommodating legacy RRM). Near-RT RIC shall be compatible with legacy RRM, enhancing traditional functions such as per-UE load-balancing, Resource Block (RB) management, interference detection and mitigation, etc. In addition, it shall provide new functions with embedded intelligence, such as QoS management, connectivity management, seamless handover control, and provide data-plane abstractions to the Orchestration and Automation layer.
- Radio-Network Information Base (R-NIB). This is a database placed within the near-RT RIC storing near real-time (<1s granularity) state of the underlying network via E2 interface and commands from non-RT RIC via A1.

- Multi-RAT (Radio Access Technology) CU (Centralized/Cloud Unit) protocol stack and platform: The function of the Multi-RAT protocol stack supports 4G, 5G and other protocol processors, enabling next-generation RATs. The basic functions of the protocol stack(s) shall be implemented according to the control commands issued by the near-RT RIC (e.g. handovers). Virtualization delivers a highly efficient execution environment for CU and near-RT RIC, providing the ability to distribute capacity across multiple network elements with, e.g., security isolation and virtual resource allocation.
- DU and RRU Function. These include real-time L2 functions, baseband processing and radio frequency processing.

The main interfaces are the following:

- Open F1/W1/E1/X2/Xn interfaces. These are standard interfaces from 3GPP, which may be enhanced to support interoperation among multiple vendors.
- A1 interface: A1 is the interface between the Orchestration layer containing the non-RT RIC and the eNB/gNB containing the near-RT RIC. A1 shall enable network management applications in non-RT RIC that are able to receive and act on highly reliable data from the CU and DU (Distributed Unit) in standardized format. Messages generated from AI-enabled policies and ML-based training models in non-RT RIC are passed to the near-RT RIC for runtime executing through this interface. A1 provides the ability to modify the RAN behaviour by deploying different models optimized to individual operator policies and objectives.
- E2 interface: E2 is the interface between the near-RT RIC and the Multi-RAT CU protocol stack and the underlying RAN DU. The baseline design of this interface corresponds to the interface between legacy RRM and RRC in traditional systems. Then, the E2 interface shall evolve to provide more intelligence, including feeding data to the near-RT RIC to facilitate radio resource management and send configuration commands to the CU/DU.

4.2.2. Algorithms Innovations

4.2.2.1. Innovation 8 (I8): Smart orchestration and resource control

A conceptual view of this innovation architecture is shown in Figure 36.

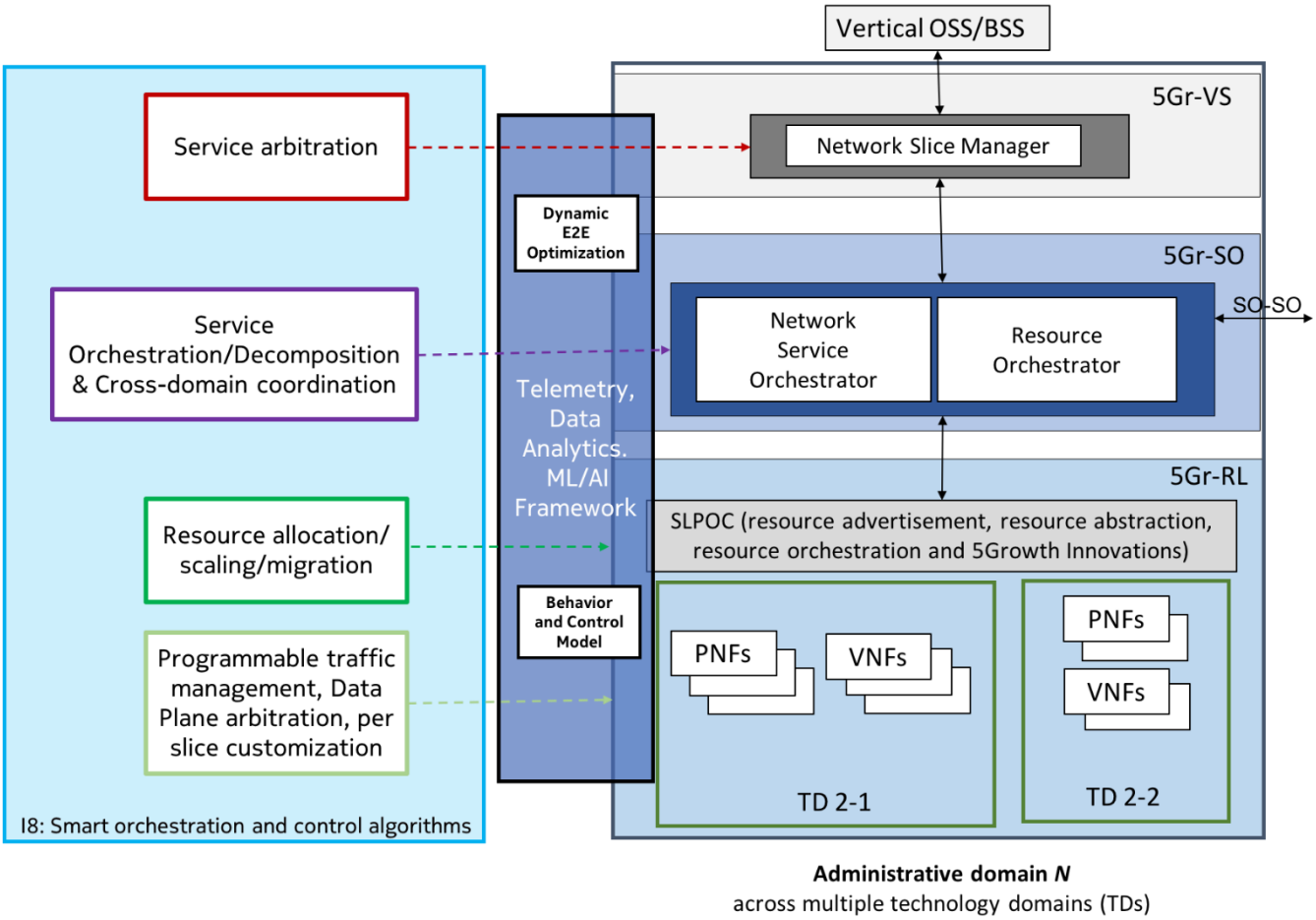

FIGURE 36: 5GROWTH INNOVATIONS TOWARDS SMART ORCHESTRATION AND RESOURCE CONTROL

5Growth Vertical Slicer (5Gr-VS): The 5Gr-VS will be the common entry point for all verticals into the 5Growth system. In accordance with the 5G-TRANSFORMER architecture, the 5Gr-VS will offer a catalogue of vertical service blueprints, based on which the verticals can generate their service requests. The role of the 5Gr-VS is to trigger the actions allowing the 5Growth system to fulfil and support the requirements of a given service request. The 5Gr-VS will translate the service requirements to slice related requirements and decide whether the service is included in an already existing slice or a new one should be created. The final result of this process will be a request sent by the 5Gr-VS towards the 5Gr-SO to create or update the NFV network services (NFV-NS) that implement the slices. The particular component is conscious of the business needs of the vertical and their SLA requirements.

In the baseline architecture, the vertical slicer is responsible for addressing intra-vertical service arbitration; that is resolve intra-vertical contention by prioritizing those services of the vertical that are more critical. Such arbitration is based on global budgets for resource utilization, as defined in the corresponding SLAs. The arbitration component in the 5GT-VS maps priorities of services and SLA requirements to resources' cardinality changes. These allow the 5GT-SO to take decisions for the appropriate allocation of resources to the most important vertical service instances, in case of actual resource shortage.

The innovation in 5Growth will revolve around introducing novel on-line AI/ML algorithms, collecting monitoring data of vertical service instances and returning fine-grained arbitration. The goal is to support both intra-vertical and inter-vertical service arbitration, which takes into account the expected resource margin (for both compute and networking resources) with respect to the SLA that a specific service will likely incur. This will allow the 5Gr-SO to optimize, scale and migrate network services accordingly.

5Growth Service Orchestrator (5Gr-SO): In accordance with the baseline architecture, the main duty of the 5Gr-SO will be to provide end-to-end orchestration of the NFV-NS within either a single or across multiple administrative domains by interacting with the local 5Gr-RL and, if needed, with other 5Gr-SOs managing remote domains (i.e., federation). In the latter case, the 5Gr-SO decides upon the appropriate decomposition of the NFV-NS across the different administrative domains, while it hides the federation details from the 5Gr-VS and therefore the respective vertical. The functionality of the 5Gr-SO is broken down to the network service orchestrator (NFVO-SO) and the resource orchestrator (NFV-RO). As described in Section 2.2.2, the former is responsible for (potentially) the decomposition of the complex NFV-NS and its lifecycle management while the latter takes resource allocation decisions at a higher level of abstraction than the corresponding decisions taken at the 5Gr-RL (e.g., VIM or WIM level). The 5GT-SO also embeds the Virtual Network Function Managers (VNFM) to manage the lifecycle of the VNFs composing the NFV-NS.

The 5Gr-SO will enhance the baseline SO in the following ways:

- The abstracted model and information studied in the 5G-TRANFORMER project, provided by the 5GT-MTP to the 5GT-SO, will be extended in the context of 5Growth to align with the new capabilities supported by the 5Growth architecture (e.g., telemetry and monitoring).
- 5Growth will propose novel NFV-NS resource allocation approaches that will enable not only satisfying the imposed SLAs when deploying the network service but also to attain an efficient use of all the resources handled at the 5Gr-SO for the entire NFV-NS lifecycle (i.e., (re)optimization, scaling, migration, etc.). To this end we will introduce on-line, algorithms for multi-domain service orchestration, supporting the different requirements imposed by the verticals (e.g., end-to-end latency, reliability etc.). This problem entails the selection of the compute resources at different NFVI as well as network resources over the (abstracted) infrastructure providing the NFVI connectivity. In other words, the problem amounts to service chain decomposition between the different administrative domains and SFC (Service Function Chaining) embedding per selected domain. Novel adaptive and real time solutions will be proposed powered by real-time telemetry, analytics and AI/ML to favour cost-efficient decisions pertaining to the lifecycle of the network service, so that targeted SLAs are neither unsatisfied nor degraded. Emphasis will be placed in resource orchestration for next-generation virtualized RANs, jointly controlling the allocation of computing, memory, and radio resources when multiple Distributed Units (DUs) share common infrastructure.

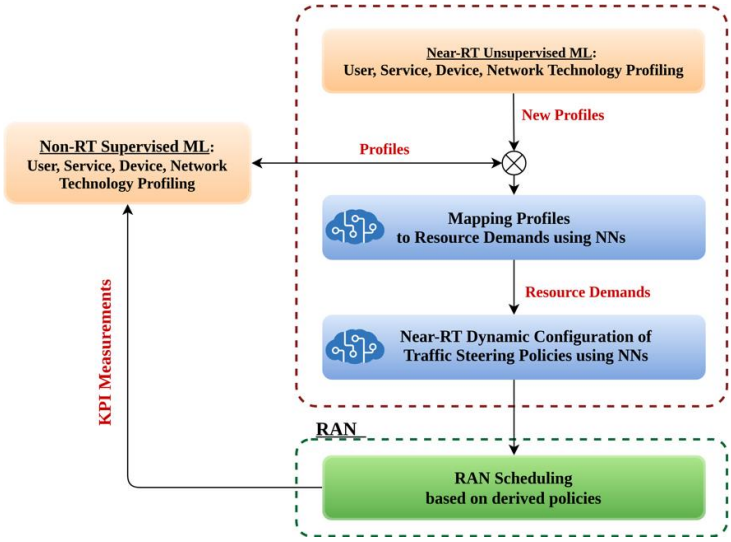

FIGURE 37: 5GROWTH PROFILING USING ML

5Growth Resource Layer (5Gr-RL): In a similar fashion with the 5GT-MTP in the baseline architecture, the 5Gr-RL 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 and transport network, computing and storage infrastructure. Having multiple technology domains under its control (e.g., data centres, mobile network, wide area network), the 5Gr-RL SLPOC (Single Logical Point of Contact) acts as an end-to-end resource orchestrator, i.e. coordinating multiple controllers (i.e., VIMs and WIMs) governing a number of technological domains. This in turn allows the 5Gr-RL SLPOC collecting granular information from such controllers, that eventually could be abstracted (according to a number of levels pursuing different criteria e.g., for scalability reasons) and then passed to the 5Gr-SO. The orchestration primitives in 5Growth will evolve beyond simple schemes used to date (e.g., for auto-scaling), as elastic network services require optimal multi-domain/-level automated self-management. To do this, the project needs to understand also the dynamic behavior of a network slice. Therefore, in this context, 5Growth will:

- Propose mechanisms with the required abstraction model of the underlying (heterogeneous) resources, involving compute and network, and consumed by 5Gr-SO.
- Investigate user, service, device and network technology profiling using Non-Real Time supervised and near-Real Time unsupervised ML algorithms. A Neural Network model will exploit these profiles to extract resource demands for the underlying network components. The resource demands will be eventually used as input for the proposed network optimization approaches (i.e., pertaining to resource allocation, scheduling). Non-RT ML algorithms will update the pre-defined profiles and create new policies to adapt the resource management accordingly, based on the actual traffic. The aforementioned process is depicted in Figure 37, focusing on the RAN.
- Investigate the statistical relationship/correlation of traffic flows and VNF load (e.g., using Regression Analysis) from data related to deployed SFCs during field trials.
- Devise on-line resource allocation algorithms complementing/expanding 5Gr-SO SFC embedding decisions (in terms of both compute and networking) to achieve an efficient resource allocation. As we are targeting optimized end-to-end resource allocation across

wireless, packet and optical switching domains, the proposed algorithms need to ensure not only the adequate adaptation among those different transport layers, but also the technological restrictions and constraints imposed by each one. In such a heterogeneous data plane scenario, a crucial aspect to deal with the efficient resource utilization is related to favour the transport of multiple network services (i.e., higher-layer data flows) over coarse bandwidth tunnels provided by lower layer technologies (e.g., packet over optical channels). In other words, the algorithms will exploit the benefits of statistical multiplexing. Finally, adaptation/re-optimization mechanisms (analogous to the 5Gr-SO) would be also handled at the 5Gr-RL. The purpose is to continuously ensure the network services' SLAs, in the corresponding closed control loop at the 5Gr-RL, autonomously to the 5Gr-SO operations (i.e., decisions and triggering). To do that, gathered monitored information at the 5Growth RL would be used as inputs to AI/ML mechanisms to be conceived.

5Growth will investigate service performance assurance at the resource layer, by introducing (re)programmable SLA-aware traffic management algorithms at the data plane. The objectives are:

- Enable data plane customization per network slice, in order to support network level KPIs/ SLAs via real-time control and traffic management. Programmable data plane solutions such as P4 [69] and supported architectures (e.g., Portable Switch Architecture (PSA)), provide an excellent way to define the packet forwarding behavior of network devices. However, most programmable devices still have non-programmable traffic managers. In the context of 5Growth, we will contribute to ongoing efforts on making traffic management logic in (HW) switches programmable, and also propose a solution that *supports customized per-slice traffic management* in the data plane that is portable between hardware and software systems. Moreover, 5Growth will also propose a solution that supports the *orchestration of traffic management per slice*, reconfiguring the data plane to meet network KPIs and SLAs defined at the slices and by verticals. Such as solution will include an open programmable interface, integrated into 5Growth, which translates SLA network requirements into configuration options which will be programmatically committed to the data plane (e.g. updating the QoS scheduler parameters and/or algorithm, flow prioritization by upgrading or downgrading flows in the slices).
- 5Growth will investigate traffic management scheduling algorithms across slices, to arbitrate in data plane resource usage at the transport, edge and DC domains, by leveraging on the programmable interfaces developed in the context of 5Growth and behaviour profiles based on the monitoring data, to assist the scheduling. Focusing on potential RAN bottleneck, we will use low-level hardware scheduling primitives along with the open-source OpenAirInterface (OAI) gNB[20] or srsLTE[21], programmed using some high-level Domain Specific Language (DSL) like P4.

[20] 5G-NR roadmap for OAI available at https://ieeexplore.ieee.org/document/8727207
[21] https://github.com/srsLTE/srsLTE

4.2.2.2. Innovation 9 (I9): Anomaly detection algorithms

A first high-level conceptual architecture of the Anomaly Detection module is illustrated in Figure 38:

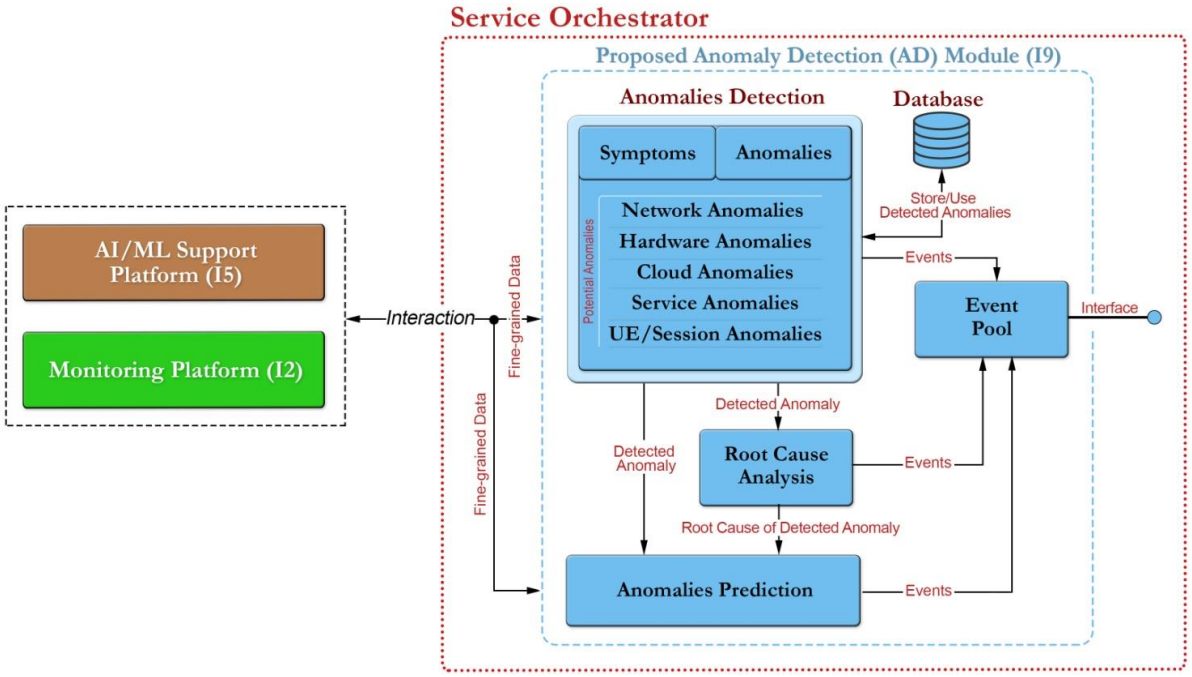

FIGURE 38: INITIAL CONCEPTUAL ARCHITECTURE OF THE ANOMALY DETECTION (AD) MODULE (I9).

The primary components are described below. Please note that the Monitoring Platform and the AI/ML Support Platform represents the innovations I2 and I5 (Sections 4.2.1.2 and 4.2.1.5) respectively, which are going to be extended towards the requirements of the AD (Anomaly Detection)

> NOTE: It is important to note that if it is decided, as the project progresses, that the AI/ML Support (I5) platform is responsible for exposing all the monitoring data to the rest of the modules, then the Anomaly Detection Module (I9) is going to have a direct interaction with AI/ML Support platform (I5) instead of interacting with the Monitoring Platform (I2).

The **Monitoring Platform** and **AI/ML Platform** are not internal sub-modules of AD and are presented in more detail in Section 4.2.1.2 and 4.2.1.5, respectively.

Anomalies Detection Module:

Anomaly Detection Module is broken down into two (2) sub-modules and is responsible for the analysis of the aggregated and fine-grained data produced by the monitoring module in order to:

- Identify possible service, network, UE/session as well as cloud related anomalies (symptoms)
- Detect the actual anomalies from the list of symptoms that would potentially lead to system malfunctioning or lack of resources.

Root Cause Analysis Module:

Root Cause Analysis Module is responsible to give an understanding of the detected anomaly. Fusing the prior knowledge with ML techniques will resolve the generation of these anomalies and will define the causes so as to reduce the human-involvement by automating the design of detection patterns.

Anomalies Prediction Module:

Anomaly Prediction Module is responsible of predicting already observed types of anomalies using RNNs (Recursive Neural Network) models, based on historical data and the output of Monitoring platform (I2), Anomaly Detection and Root Cause Analysis Module so as to proceed to a fast system recovery.

Event Pool:

The event pool will be responsible for storing new and more complex types of events/alerts by collecting and aggregating the outputs of Anomaly Prediction, Anomaly Detection and Root Cause Analysis Module in order to be exploited by the final consumers (e.g., resource/mobility management network functions, the Vertical Slicer, 3rd party applications, etc.).

4.2.2.3. Innovation 10 (I10): Forecasting and inference

This innovation will provide functionalities that can be exploited by the 5Growth architecture at different layers. Figure 39 depicts the most likely I10 interfaces. The I2-I10 interface allows the Forecasting and Inference module to access the time series collected by the Vertical-oriented Monitoring System developed in I2. The received data will be utilised to perform prediction exploiting either AI/ML models available in the AI/ML platform through the I2-I5 interface or internal models (e.g., based in time series analysis approaches). The forecasting and inference results will be exposed to the Smart Orchestration and Resource Control developed in I8 through the I8-I10 interface.

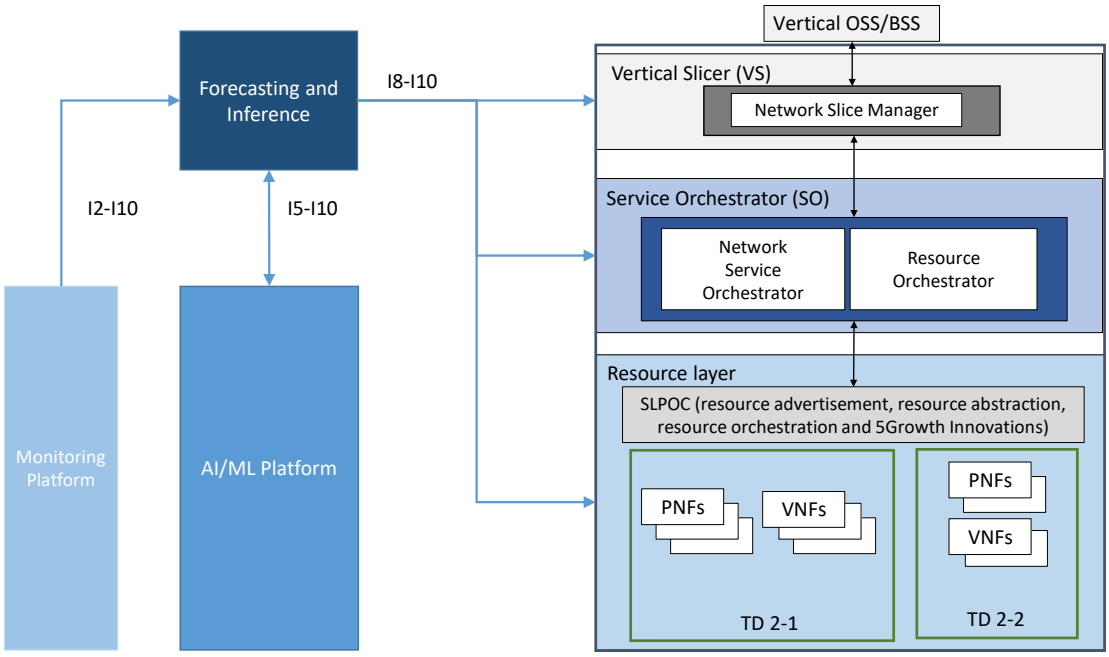

FIGURE 39: INPUTS AND OUTPUTS TO THE I10 INNOVATION FUNCTIONAL BLOCK

Forecasting and inference will be based on several algorithms based on classical time series analysis, such as single, double, triple exponential smoothing and on AI/ML techniques such as Long Term Short Term Memory (LSTM). They will be compared in terms of execution time, training time, and required resources.

Classical time series analysis algorithms provide simple formulas to forecast time series analysis trends. Thus, they are relatively simple and fast to be executed.

AI/ML techniques are capable of analyzing a massive amount of data but they might require a large amount of resources to run and present longer execution time and longer training.

Based on sample time series analysis performance evaluation the most suitable algorithm for predicting specific parameters will be selected. For example, if the considered parameter presents a fast variations (i.e., large bandwidth in the frequency domain) forecasting algorithms presenting a fast execution time will be selected. This performance could be traded off with forecast accuracy. Conversely, if the considered parameter does not present fast variations (i.e., small bandwidth in the frequency domain) more precise but slower execution time algorithms will be selected.

4.2.3. Framework Innovations

4.2.3.1. Innovation 11 (I11): Security and auditability

This innovation introduces the applicability of non-repudiation mechanisms and Moving Target Defense/Cyber Mimic Defense (MTD/CMD) methods in the 5Growth platform architecture.

On the one hand, the non-repudiation mechanisms, which are aimed at ensuring platform auditability, can be integrated at multiple layers and applied at both horizontal and vertical interactions, even within the same administrative domains. Specifically, non-repudiation

mechanisms shall allow the generation of available, clear, original (unchanged), verifiable and accurate audit trails reflecting the:

- Communications between the 5Growth platform and ICT-17 management systems (audit trails of type #1), enabling the interplay of a non-public network domain with one or more public network domains at the orchestration and control level. The messages exchanged between these two administrative domains, optionally including the notifications related to the requested operations, need to support non-repudiation.
- Communications between the 5Growth platform and the industry vertical, who takes the role of the platform's customer (audit trails of type #2). These communications result in interactions across two non-public domains. Non-repudiation mechanisms similar to the one described above are also applicable in this scenario.
- Communications between modules within the 5Growth platform, for those cases where the involved modules are provided and maintained by different actors (audit trails of type #3). Unlike the two above scenarios, these exchanges enable vertical interactions without leaving the logical boundaries of the platform's domain. These interactions are not only applicable between modules from different subsystems (i.e. Vertical Slicer (VS) subsystem, Service Orchestrator (SO) subsystem and Resource Layer (RL) subsystem), where multi-actor environments are relatively common, but also between modules from the same subsystem. A typical example of this scenario can be found in the 5Gr-RL subsystem, when the Resource Orchestrator (RO) and the Virtualized Infrastructure Manager (VIM) are administrated by different stakeholders. Ensuring non-repudiation in the communication between these two modules allows keeping trusted records of virtualized resource management request-responses, which is of key relevance to support the verification of quota reservation and resource consumption in multi-tenancy environments.
- Telemetry data, including collected performance measurements and fault alarms (audit trails of type #4). The integration of non-repudiation mechanisms for these metrics provides means to support the verification and orchestration of Service Level Agreements (SLAs).

Figure 40 shows the integration of non-repudiation mechanisms required to keep the abovementioned audit trials in the 5Growth platform.

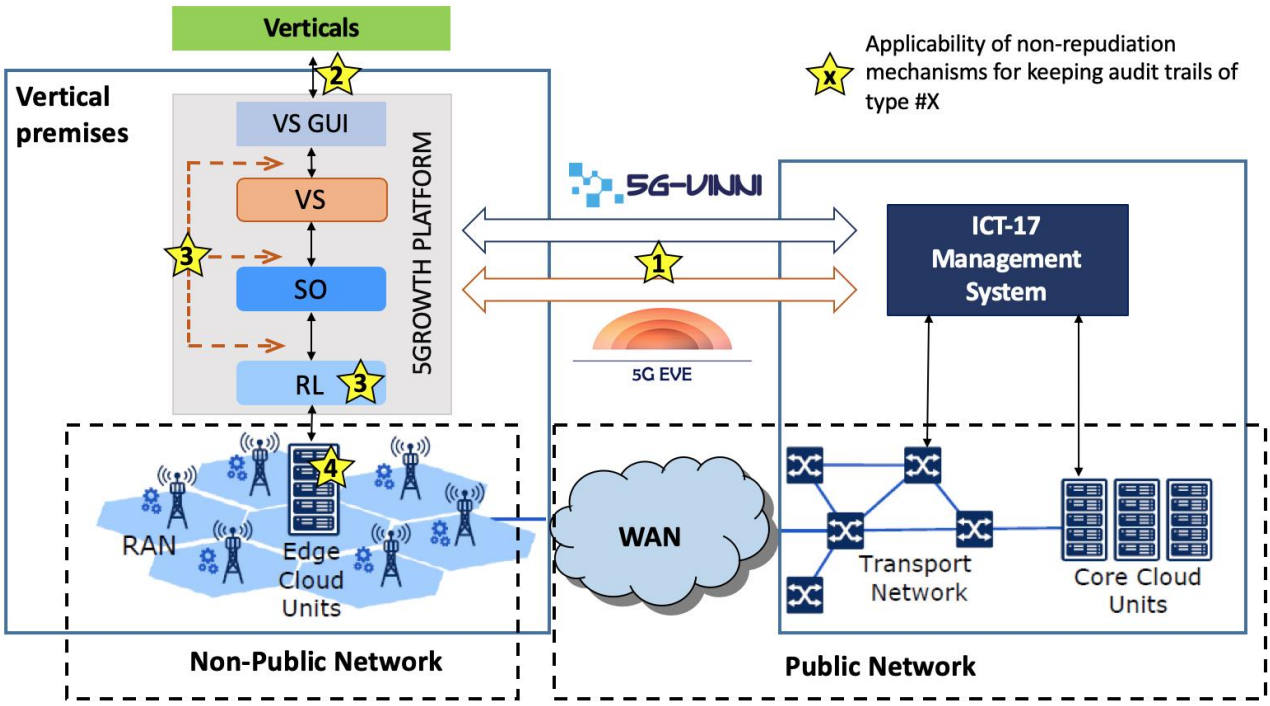

FIGURE 40: AUDATIBILITY IN THE 5GROWTH PLATFORM

The interfaces involved in the types of audit trails described above shall have to incorporate the elements to provide the appropriate fingerprinting mechanisms, as well as the trusted store mechanisms. Ideally, a common fingerprinting module will be made available for deployment at these interfaces, together with the procedures to create, structure and access trusted stores.

On the other hand, Figure 41 shows how the MTD and CMD mechanisms can be integrated into the 5Growth platform architecture.

An assessment of the integration potential of MTD mechanisms into the main components (e.g., 5Gr-VS GUI, 5Gr-VS, 5Gr-SO, or even the 5Gr-RL abstraction) of the 5Growth platform will be realized, with the aim of providing an additional layer of security against the adversarial exploitation of vulnerabilities in these services. The same mechanisms may also be extended to the verticals in two ways: (1) as a requirement to interact with the 5Growth platform, meaning that only pre-authorized clients could interact with its interfaces, and (2) as a Security-aaS enhancement to the services deployed by the vertical, defending the vertical functions (not just the platform itself).

The same MTD mechanisms may also be extended to the interaction between the 5Growth platform and the ICT-17 management systems. However, further research is necessary to assess its feasibility, as the inclusion of such a mechanism would also create an effort on the ICT-17 side. Additionally, the degree of exposure of the 5Growth platform to attacks via the ICT-17 interfaces must also be assessed, so that mitigations can be in place through other layers of security.

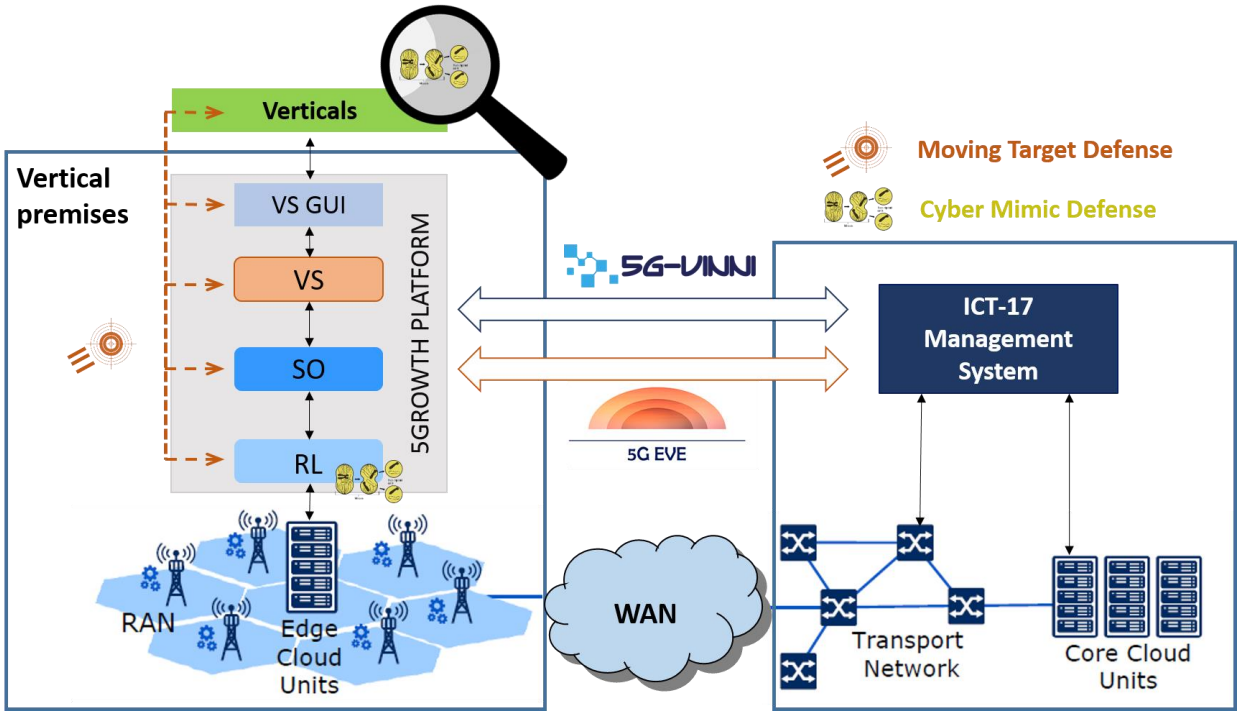

FIGURE 41: MTD AND CMD IN THE 5GROWTH PLATFORM

The CMD mechanisms will play a more targeted role within the logical perimeter of the 5Growth platform. Due to their high effort nature, these mechanisms shall only be used when very stringent assurances of correctness are required. For instance, high-value targets that attract the more sophisticated attackers, such as public-utilities or public-safety services. Within the 5Growth platform, such services will exist inside the verticals (only specific functions). The resource layer of the platform is also a suitable candidate to deploy these mechanisms when required. In this case, the CMD mechanisms can be used to provide assurances to the 5Growth platform that the resolved rules are in fact being enforced at their enforcement point. This mechanism is essential to thwart supply chain attacks against the 5Growth platform or the verticals that rely upon the platform.

4.2.3.2. Innovation 12 (I12): 5Growth CI/CD

The CI/CD (Continuous Integration and continuous delivery) concept is a process pipeline covering automation of all software lifecycle stages - developing, integrating, building, testing and deployment.

Figure 42 shows the full high-level CI/CD pipeline from the moment of committing new code till deploying changes to production environment.

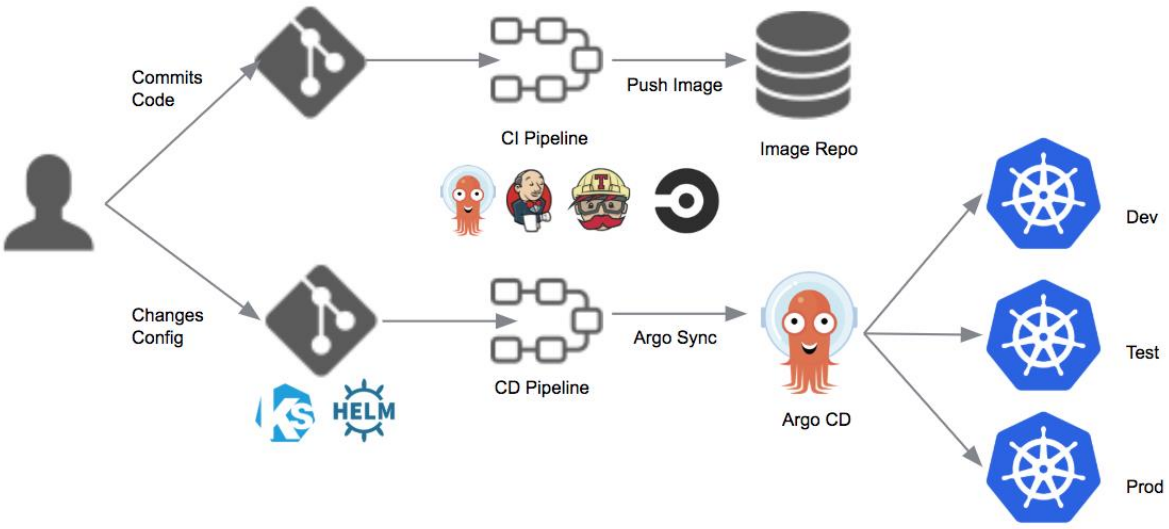

FIGURE 42: CI/CD PIPELINE FLOW WITH ARGOCD ON TOP OF KUBERNETES

Automation starts to work right after a developer pushes the changes to repository. First step - is to detect new changes by a CI tool (ArgoCD[22], Jenkins[23], Travis[24], CircleCI[25], etc.), usually with the help of webhooks. When the CI tool detects changes, the CI pipeline starts. In most cases, the CI pipeline consist of a few tasks, the most important are:

- Pooling the new code;
- Building the application from the pooled source code;
- Making tests – The amount of tests may vary, but unit tests (to determine whether regression happened) and integrations tests (to determine whether new code can be integrated) are necessary (for the moment it is not clear who will be responsible for providing those tests, but automated testing is vital for the concept).
- Depending on the test results, next steps are reporting failures (mail, skype, slack, etc.) or promotion (packaging, pushing to image or package repo for future use as a new version) in case of success.

In case of Kubernetes continuous deployment is significantly simpler compared to VM based clouds, as most of the operations Kubernetes will handle itself, it is just needed to describe the desired state. Still CD pipeline is strongly dependent on a tool - the following is a description of ArgoCD flow.

The ArgoCD was chosen because it is lightweight Kubernetes-native continuous delivery system. Its main idea is based on GitOps paradigm for CD, which states that Git must be a single source of

[22] https://argoproj.github.io/projects/argo-cd

[23] https://jenkins.io

[24] https://travis-ci.org

[25] https://circleci.com

truth. After the component config changes are pushed into the component config repo, ArgoCD syncs the app state with the changes, i.e., changes are declarative.

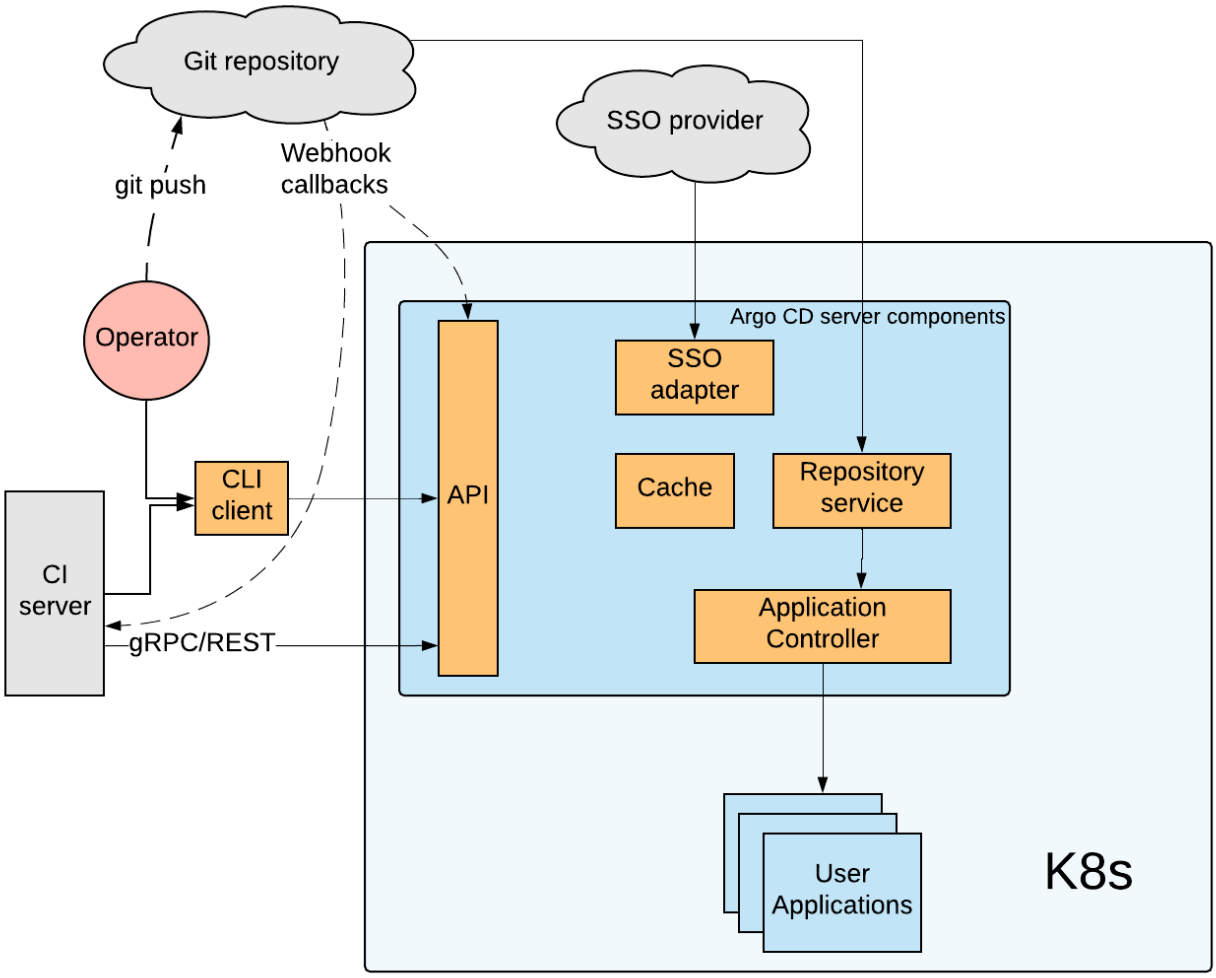

FIGURE 43: ARGOCD ARCHITECTURE

Due to being kubernetes-native ArgoCD is in charge of Kubernetes entities - deployment, services, pods, etc; and has all advanced kubernetes features - healthchecking, autoscaling, rolling upgrade or roll-back to any application state committed in the git repository. ArgoCD supports multiple manifests template formats: Helm[26], Kustomize[27], Ksonnet/Jsonnet[28] or plain YAML[29].

As shown on the architecture figure (Figure 43) ArgoCD consists of the following components:

[26] https://helm.sh

[27] https://kustomize.io

[28] https://ksonnet.io

[29] https://yaml.org

- API server - which is responsible for application management and status reporting, invoking of application operations (e.g. sync, rollback, user-defined actions), credential management, auth/authz management, listener/forwarder for Git webhook events;
- Application Controller - which adjusts the application's actual state with the desired target one;
- Repository Server - maintains a local cache of the Git repository;
- Argo CD Client - a CLI application to interact with the argocd server;
- SSO (Single Sign On) adapter - that implements integrations with external SSO providers (Github, Google Cloud, LDAP etc.);
- Cache - is a Redis[30] instance that caches frequently accessed objects.

[30] https://redis.io

5. Conclusions

In this deliverable we have performed a gap analysis between our reference baseline 5G platform from the 5GPPP phase 2 project 5G-TRANSFORMER, and the function requirements of the 9 vertical pilot use cases to be implemented and demonstrated in the project. This has led us to design a set of 12 innovations and develop a 2-release plan for the development of the 5Growth platform. This document reports the initial design of these 12 innovations, presenting their main objectives and high-level ideas of both conceptual and architectural level suggesting required extensions from the baseline 5G-TRANSFORMER platform on the new/enhanced interfaces and new function modules to support these innovations.

The first release (Release 1, Month 12) will provide 5Growth baseline features (those from 5G-TRANSFORMER) plus initial versions of first selected set of innovations prioritized due to their relevance to WP1 use cases and WP3 pilots. Specifically, I1 (RAN segments in network slices) and I2 (Vertical-service monitoring) will provide a much enhanced support to the verticals; and I4 (Control-loops stability) and I8 (Smart orchestration and resource control algorithms) will help in automate the process of optimization, adaptation, cost reduction and zero-touch management of the vertical services as required by the novel stringent use cases of interest for 5Growth.

The second release (Release 2, Month 24) aims to provide further support to new technologies such as next-generation (virtual and open) RAN solutions and AI/ML based resource control and smart service orchestration workflows and algorithms. The continuous development and integration (CD/CI) of these innovations into 5Growth platform will go along the design and implementation of the remaining innovations: Monitoring orchestration (I3), AI/ML support (I5), federation and inter-domain (I6), next-generation RAN (I7), anomaly detection mechanisms (I9) and forecasting and inference algorithms (I10). Innovations CI/CD (I12) and security & auditability (I11) are two integral components of 5Growth which will be part of the framework for design and development of all innovations in the project.

This 2-release platform plan and clustering of innovations help to prioritize the most relevant features required for the appropriate development of vertical pilots in WP3, while, at the same time, it lets us balance work across the different tasks and innovation working groups to parallelize efforts and optimize progress. Ultimately, an integrated 5Growth platform with all 12 innovations, individually tested through unit tests (UTs) led by T2.2, T2.3 and T2.4 and integration tests (ITs) coordinated by T2.1, will be provided as the final outcome of WP2 to serve the vertical pilots selected for 5Growth.

6. References

[1] 5G-TRANSFORMER, D1.3, 5G-TRANSFORMER refined architecture, May 2019. http://5g-transformer.eu/wp-content/uploads/2019/05/D1.3_5G-TRANSFORMER_Refined_Architecture.pdf

[2] 5G-TRANSFORMER, D2.3, Final design and implementation report on the MTP: Design, May 2019. http://5g-transformer.eu/wp-content/uploads/2019/05/D2.3_Final_design_and_implementation_report_on_the_MTP_report.pdf

[3] 5G-TRANSFORMER, D3.3, Final design and implementation report on the Vertical Slicer, May 2019. http://5g-transformer.eu/wp-content/uploads/2019/05/D3.3_Final_design_and_implementation_report_on_the_Vertical_Slicer_report.pdf

[4] 5G-TRANSFORMER, D4.3, Final design and implementation report on service orchestration, federation and monitoring platform, May 2019. http://5g-transformer.eu/wp-content/uploads/2019/05/D4.3_Final_design_and_implementation_report_on_service_orchestration_federation_and_monitoring_platform_report.pdf

[5] 5Growth. "Business Model Design." Deliverable D1.1, September 2019.

[6] ETSI ISG NFV-IFA 005, "Network Function Virtualisation (NFV), Management and Orchestration; OR-VI reference point- Interface and Information Model Specification", v2.1.1, Apr. 2016

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

[8] 5GEx deliverable D2.1, public version "Initial System Requirements and Architecture, available at https://drive.google.com/file/d/0B4O_JVjsvab9VXItUDJqMUR6d28/view

[9] 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

[10] 3GPP TR 28.801, V15.1.0, Telecommunication management; Study on management and orchestration of network slicing for next generation network, January 2018.

[11] 3GPP TS 23.501, V16.0.2, System architecture for the 5G System (5GS), April 2019.

[12] 3GPP TR 23.714, V14.0.0, Study on Control Plane (CP) and User Plane (UP) separation of Evolved Packet Core (EPC) nodes, June 2016.

[13] 3GPP TS 28.530, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Management and orchestration; Concepts, use cases and requirements

[14] (Release 16)", v16.0.0, September 2019

[15] 3GPP TS 28.541, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16)", v16.2.0, September 2019

[16] 3GPP TS 22.261, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service requirements for the 5G system; Stage 1 (Release 16)"

[17] 5GCity, D4.1: Orchestrator design, service programming and machine learning models, 2018.
[18] O-RAN: Towards an Open and Smart RAN– O-RAN White Paper, October 2018, https://www.o-ran.org/
[19] NHTSA, "Preliminary regulatory impact analysis - FMVSS No. 150 Vehicle-to-Vehicle Communication Technology for Light Vehicles," December 2016.
[20] G. Avino, M. Malinverno, C. Casetti, C. F. Chiasserini, F. Malandrino, M. Rapelli, G. Zennaro "Support of Safety Services through Vehicular Communications: The Intersection Collision Avoidance Use Case".
[21] 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.
[22] TeleManagement Forum. "TeleManagement Forum Information Framework (SID): GB922_Addendum_4SO_Service_Overview_R9-5_v9-7".
[23] ETSI GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Management and Orchestration", v1.1.1, 2014.
[24] ETSI GS NFV 003, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", v1.2.1, Dec. 2014.
[25] 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.
[26] 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.
[27] 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.
[28] 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.
[29] ETSI GS NFV-IFA 010, "Management and Orchestration; Functional requirements specification", v2.4.1, 2018.
[30] ETSI GS NFV-IFA 011, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification" v3.2.1, April 2019.
[31] ETSI GS NFV-IFA 012, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Os-Ma-Nfvo reference point - application and service management use cases and recommendations", v3.1.1, October 2018.
[32] ETSI GS NFV-IFA 013, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Os-Ma-Nfvo reference point – Interface and Information Model Specification" v3.2.1, April 2019.
[33] ETSI GS NFV-IFA 014, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification" v3.2.1, April 2019.
[34] ETSI GR NFV-IFA 022, "Management and Orchestration; Report on Management and Connectivity for Multi-Site Services", v0.8.2, 2018.
[35] ETSI GS NFV-IFA 028, "Management and Orchestration; Report on architecture options to support multiple administrative domains", v3.1.1, 2018.

[36] ETSI GR NFV-EVE 012 V3.1.1, "Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework", 2017.
[37] ETSI GS MEC 001 V1.1.1, "Mobile Edge Computing (MEC): Terminology", March 2016.
[38] ETSI GS MEC 003, "Mobile Edge Computing (MEC); Framework and Reference Architecture", v1.1.1, March 2016.
[39] 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.
[40] ETSI GR MEC 017, "Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV Environment", v1.1.1, February 2018.
[41] ITU-T TR Y.3011, "Framework of Network Virtualization for Future Networks, Next Generation Networks-Future Networks", International Telecommunication Union-Telecommunication Standardization Sector (ITU-T), Tech. Rep., 01 2012, v14.0.0.
[42] 3GPP TR 23.799, "Study on Architecture for Next Generation System", Third Generation Partnership Project (3GPP), Tech. Rep., 12 2016, v14.0.0.
[43] ETSI GS ZSM 001 V1.1.1 (2019-10), Zero-touch network and Service Management (ZSM); Requirements based on documented scenarios
[44] A. de la Oliva *et.al.*, "5G-TRANSFORMER: Slicing and Orchestrating Transport Networks for Industry Verticals", accepted by IEEE Communications Magazine, 2018.
[45] C. Casetti et.al, Network Slices for Vertical Industries, 1st Workshop on Control and management of Vertical slicing including the Edge and Fog Systems (COMPASS), Barcelona, IEEE, 2018.
[46] K. Antevski, et al., "Resource Orchestration of 5G Transport Networks for Vertical Industries", IEEE PIMRC 2018, Workshop 5G Cell-Less Nets, September 2018.
[47] 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.
[48] Salvat, Josep Xavier, et al. "Overbooking network slices through yield-driven end-to-end orchestration." Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies. ACM, 2018.
[49] 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.
[50] Ayala-Romero, Jose A., et al. "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs." The 25th Annual International Conference on Mobile Computing and Networking. ACM, 2019.
[51] Horizon 2020, 5GPPP: 5G-Crosshaul project, Available at: http://5g-crosshaul.eu/
[52] D. King and A. Farrell, "A PCE-Based Architecture for Application-Based Network Operations", IETF RFC 7491, March 2015.
[53] M. Björklund, Martin, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", IETF RFC6020, October 2010.
[54] A. Bierman et al, "Restconf Protocol", IETF RFC 8040, January 2017.
[55] L. Qiang et al., "Technology Independent Information Model for Network Slicing", draft-qiang-coms-netslicing-information-model-02, Work in progress, Expires July 30, 2018.
[56] RFC 2119, "Key words for use in RFCs to Indicate Requirement Levels", IETF, 1997.
[57] Description of Network Slicing Concept by NGMN Alliance, NGMN 5G P1 Requirements & Architecture Work Stream End-to-End Architecture, version 1.0, January 2016.
[58] NGMN Alliance, "Description of Network Slicing Concept", v.1.0.8, September 2016.

[59] NGMN Alliance, NGMN 5G White Paper, February 2015.
[60] Open Source MANO. https://osm.etsi.org/
[61] J. Baranda et.al., "Resource Management in a Hierarchically Controlled Multidomain Wireless/Optical Integrated Fronthaul and Backhaul Network", IEEE NFV-SDN conference, November 2017.
[62] 5GEx, https://5g-ppp.eu/5gex/
[63] Nick Szabo -- Smart Contracts: Building Blocks for Digital Markets, 1996, www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_cont, [Online].
[64] 3GPP TS 29.060 V15.1.0 (2017-12), General Packet Radio Service (GPRS); GPRS Tunnelling Protocol (GTP) across the Gn and Gp interface.
[65] IETF, RFC7348, Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, August 2014.
[66] Multi-access Edge Computing, http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing
[67] IETF RFC3198, Terminology for Policy-Based Management, November 2001.
[68] L. Cominardi, "Multi-domain federation: scope, challenges, and opportunities," Workshop in 3rd IEEE Conference on Network Softwarization, Bologna, Italy, July 2017.
[69] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rex- ford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, and David Walker. 2014. P4: Programming Protocol-independent Packet Processors. SIG- COMM Comput. Commun. Rev. 44, 3 (July 2014), 87–95.
[70] J.G.Herrera and J.F.Botero, "Resource allocation in nfv: A comprehensive survey," IEEE Transactions on Network and Service Management, vol. 13, no. 3, pp. 518–532, 2016.
[71] Dietrich D., Papagianni C., Papadimitriou P., Baras J.S. (2017) Near-Optimal Placement of Virtualized EPC Functions with Latency Bounds. In: Sastry N., Chakraborty S. (eds) Communication Systems and Networks. COMSNETS 2017. Lecture Notes in Computer Science, vol 10340. Springer, Cham.

7. Annex I — Preliminary Results

In the following, we present preliminary results for one of the algorithms currently designed and implemented within the scope of **Innovation I8 (Smart Orchestration and Resource Control)**.

System Model

Resource allocation for Service Function Chains (SFCs) (or equivalently VNF-FGs) can be mathematically modelled as a graph-in-graph embedding problem. The directed graph G_F associated with an SFC (Figure 44: SFC request) consists of a set V_F vertices representing the workloads g^i and a set E_F of directed edges representing the traffic flows $g^{(i,j)}$ that run from VNF i to VNF j (if they are hosted on different substrate nodes e.g., servers). The workloads g^i and traffic flows $g^{(i,j)}$ are modelled as random variables that mutually expose some statistical dependence. Service chains are requested at specific time instants (often modelled as a Poisson process) and remain active for the lifetime of the slice (also modelled as a random variable).

The directed graph G_S associated with the substrate (or equivalently the NFVI, Figure 44: Substrate) consists of a set V_S of vertices representing the network nodes (e.g., routers, servers, gateways) and the set E_S of edges, representing the directed links between these nodes. A residual compute capacity r_u and transport capacity $r_{(u,v)}$ is associated with each vertex and edge respectively and these residual capacities evolve over time.

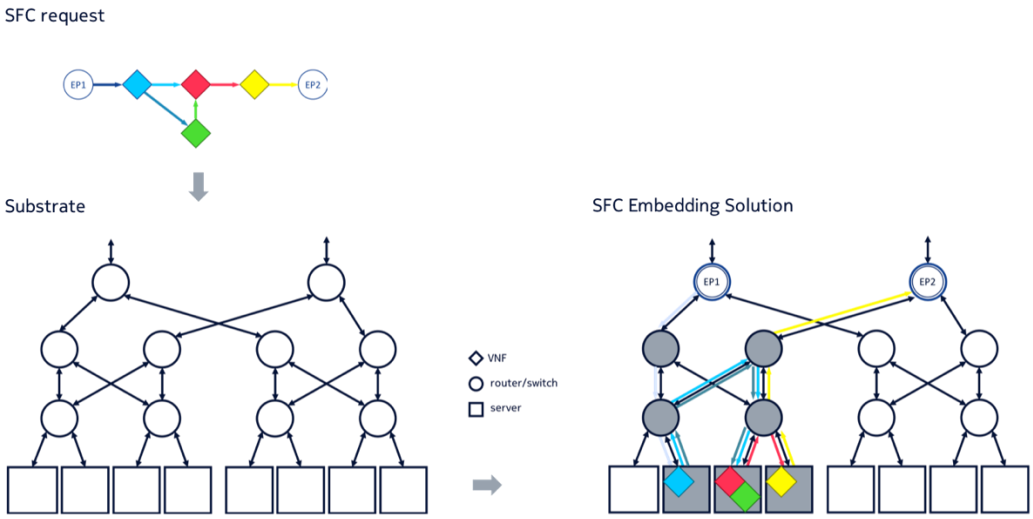

FIGURE 44: SFC EMBEDDING PROBLEM

Problem Description

At request instants each requested SFC needs to be embedded in the substrate in such a way that its operational constraints (e.g., end-to-end delay) are respected (Figure 44: SFC Embedding Solution). When a new SFC is embedded in the substrate, the remaining resource r_u and $r_{(u,v)}$ decrease; at departure instants when a previously accepted SFCs is terminated, they increase; and in between request instants they are governed by the SFCs in the set of accepted requests and the statistical fluctuations of their associated workloads and traffic flows over time.

In order to describe the embedding it is useful to introduce the following decision variables:

- x_u^i expresses that workload g^i of SFC request G_F is hosted on substrate node u or not,
- $x_{(u,v)}^{(i,j)}$ expresses the fraction of the traffic flow $g^{(i,j)}$ from workload i to workload j that is supported by the directed link (u, v).

Because each workload needs to run on one and only one server and because if a flow is either generated or transported into a certain node, it must be terminated or transported out, the following restrictions apply:

$$\sum_{u \epsilon V_S} x_u^i = 1 \text{ (Placement constraints)}$$

$$x_u^i + \sum_{(v,u) \epsilon E_S^o(u)} x_{(v,u)}^{(i,j)} = x_u^j + \sum_{(u,v) \epsilon E_S^i(u)} x_{(u,v)}^{(i,j)} \leq 1 \text{ (Flow constraints)}$$

Where $E_S^o(u)$ and $E_S^i(u)$ is the subset of the set E_S of all edges that end, respectively start, at node u. These are referred to as the valid embedding constraints.

Additionally, capacity constraints express that the remaining resources r_u and $r_{(u,v)}$ cannot become completely depleted at any time. In addition to these, other constraints, e.g., delay, reliability, security etc. depending on the use case may need to be respected too.

Often the set of real variables $f_{(u,v)}^{(i,j)}$ is introduced where $f_{(u,v)}^{(i,j)} = g^{(i,j)} x_{(u,v)}^{(i,j)}$ to replace the **x** variables, expressing the amount of traffic from workload i to workload j that is supported by the directed link (u, v). The flow constraints are reformulated as:

$$\sum_{(u,v) \epsilon E_S^i(u)} f_{(u,v)}^{(i,j)} - \sum_{(v,u) \epsilon E_S^o(u)} f_{(v,u)}^{(i,j)} = g^{(i,j)}(x_u^i - x_u^j)$$

Current Practice and Discussion

Existing approaches employ traditional optimization to address SFC embedding. As described in the previous paragraph, the problem is usually formulated as a Mixed Integer Linear Program (MILP), tailored to the specific objective that is pursued (e.g., QoS, reliability etc.). Each time a request arrives we solve a MILP (that maximizes a certain objective, under the constraints defined above) to determine the decision variables x_u^i and $f_{(u,v)}^{(i,j)}$. If no such solution exists, the request is rejected.

Since the problem is NP-hard [70], sub-optimal (meta) heuristics and approximation algorithms have been devised to make the problem computationally tractable, especially considering that SFC embedding needs to be addressed in real time ("online" problem). As the problem formulation depends on the use case objective (e.g., availability constraints take a different form than end-to-end delay constraints) the corresponding approaches are also case-specific.

In literature, most approaches dealing with the online problem make decisions based on a snapshot of the remaining capacities in the NFVI observed at the request time, and usually it is assumed that these remaining capacities are known with high precision, while the (future) evolution of the workloads for in-service (or expiring) SFCs over time is not considered (static resource allocation). Usually, the maximum or average of resources that an SFC may require over its lifetime is considered, which leads to over-provisioning of resources and as a result, rejecting incoming SFC requests that could have otherwise been accepted. In reality, the evolution over time of neither the

resources a requested SFC will require nor the residual available resources of the NFVI are known. The problem is amplified when such information is provided at a high level of abstraction. In addition, making embedding decisions based on a snapshot of the remaining resources at request time is not optimal over time. Due to the finite lifetime of (most) SFCs, resource availability is likely to increase during the SFC inter-request time. This trend can be learned and taking it into account can lead to better decisions.

RL-based SFC embedding

In such an environment with uncertain resource supply and resource demand, Reinforcement Learning (RL) is ideally suited to tackle the SFC embedding problem. 5Gr-RL gradually steers the decision-making process in the right direction based on feedback it gets on how good the embedding decision was. The approach can be used to address the "online" problem, supporting decision-making in real (polynomial) time.

Each time request arrives, the current policy of the 5Gr-RL algorithm decides if and how to embed the SFC in the NFVI. By rewarding embeddings that do not violate any constraints and penalizing embeddings that do, the 5Gr-RL algorithm gradually learns the best embedding policy. At the start (or when circumstances change) the quality of the embedding is not suboptimal, but gradually it gets closer to the optimal policy.

Indicatively, two different simulation scenarios are evaluated. For the first simulation scenario, we compare the efficiency of the 5Gr-RL approach to the benchmarks MILP approach described earlier, using as traffic envelope the maximum inbound traffic demand per service chain (MILPmax) and the average inbound traffic demand per service chain (MILPavg). We compare them on the basis of the SFC request rejection ratio that is the ratio of rejected requests divided by the total number of requests, and the resource violation ratio is the ratio of the monitoring instances at which any of the resources is violated to the total number of monitoring instances (implicitly considering SLA violations).

Figure 45 depicts the evolution of the two metrics for the embedding algorithms. The RL-based approach converges after approximately 1500 requests. In steady state there are still fluctuations due to the stochastic nature of the requests and the exploration capability of the RL approach. MILPmax has no violations by design but exhibits the highest rejection ratio as it takes into account the maximum inbound traffic demand per service chain. MILPavg on the other hand presents a high violation ratio, due to the underestimation of the resource demands, at similar rejection ratio level to the 5Gr-RL. 5Gr-RL manages to keep the resource violation ratio low, without considering capacity constraints for the embedding problem and having limited information on the infrastructure resources, as opposed to the MILP that is provided with the remaining compute and transport capacity in full precision.

RL MILPmax MILPavg

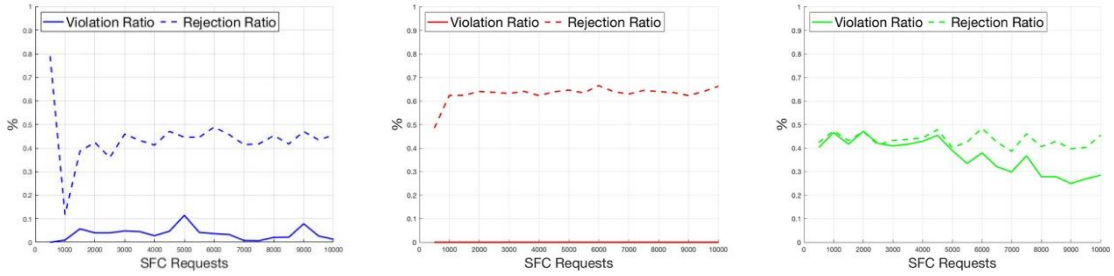

FIGURE 45: RESOURCE VIOLATION - REQUEST REJECTION

For the second simulation scenario, we study the ability of the 5Gr-RL approach to adapt fast to changing conditions such as a surge in workload/traffic demands. To assess this aspect, we increase the requested workload halfway through the simulation; for the 5000 remaining SFC requests the corresponding inbound traffic is increased approximately by 30%. Figure 46 shows that the proposed RL-based approach is able to adapt and converge to a new "steady state" fast according to the policy imposed, while the MILPmax approach is not able to cope with these changing conditions. The resource violations in the MILPmax case could only be avoided by resetting the traffic envelope for the incoming requests at the second half of the simulation.

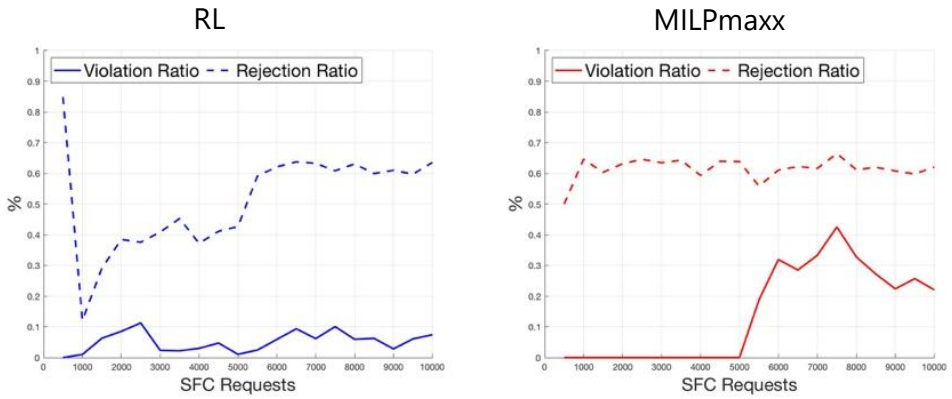

FIGURE 46: RESOURCE VIOLATION - REQUEST REJECTION UNDER CHANGING CONDITIONS

8. Annex II — Glossary

This document defines the key terminology used in 5Growth. The definitions of terms are structured according to their area, such as virtualization-related or business-related. The terms defined here are the most relevant ones, especially those that have different definitions by various standardization developing organizations and it is largely inherit by 5G-TRANSFORMER project.

8.1. General Terms

Data model: [67] a mapping of the contents of an information model into a form that is specific to a particular type of data store or repository. A "data model" is basically the rendering of an information model according to a specific set of mechanisms for representing, organizing, storing and handling data. It has three parts:

- A collection of data structures such as lists, tables, relations, etc.
- A collection of operations that can be applied to the structures such as retrieval, update, summation, etc.
- A collection of integrity rules that define the legal states (set of values) or changes of state (operations on values).

Information model (IM): an abstraction and representation of the entities in a managed environment, their properties, attributes and operations, and the way that they relate to each other. It is independent of any specific repository, software usage, protocol, or platform. [67]

Policy: [1] policy can be defined from two perspectives:

- A definite goal, course or method of action to guide and determine present and future decisions. "Policies" are implemented or executed within a particular context (such as policies defined within a business unit).
- Policies as a set of rules to administer, manage, and control access to network resources.

NOTE: [67] These two views are not contradictory since individual rules may be defined in support of business goals.

Service: the behavior or functionality provided by a network, network element or host. To completely specify a "service", one must define the "functions to be performed ..., the information required ... to perform these functions, and the information made available by the element to other elements of the system". Policy can be used to configure a "service" in a network or on a network element/host, invoke its functionality, and/or coordinate services in an interdomain or end-to-end environment. [67]

8.2. Network function virtualization related

The central concepts around network function virtualization and network services are based on the definitions of ETSI NFV.

Network Function (NF): functional block within a network infrastructure that has well-defined external interfaces and well-defined functional behaviour. [24]

Network Service (NFV-NS): composition of Network Functions and defined by its functional and behavioural specification. [24]

NOTE: "The Network Service contributes to the behaviour of the higher layer service, which is characterized by at least performance, dependability, and security specifications. The end-to-end network service behaviour is the result of the combination of the individual network function behaviours as well as the behaviours of the network infrastructure composition mechanism." [24]

NOTE: A network service can be seen as a set of VNFs or PNFs, connected by VLs as defined in a VNFFG.

Network Service Descriptor (NSD): it is used in the 5Growth project as network slice template that describes the deployment of a Network Service including service topology (constituent VNFs and the relationships between them, Virtual Links, VNF Forwarding Graphs) as well as Network Service characteristics such as SLAs and any other artefacts necessary for the Network Service on-boarding and lifecycle management of its instances. [24]

NOTE: The NSD includes a number of deployment flavours, each referencing deployment flavours of all or a subset of the NFV-NS's constituent VNFs and Virtual Links. The NSD also provides a list of pointers to the descriptors of its constituent VNFs (i.e. VNFDs) and additional information on the connectivity between them together with the traffic forwarding rules.

Network Service Instance (NFV-NSI): refers to an instance of a network service (NFV-NS).

NOTE: In our case the process for instantiating a Network Service instance is initiated from 5GT-VS by sending a request to the 5GT-SO. This request typically contains a pointer to a NSD, a flavour selector and additional input parameters (e.g., IP addresses to be assigned to some of the network functions) and constraints (e.g. location where to deploy all or some of the network functions). The flavour mechanism not only enables selecting a subset of VNFs and virtual links to be instantiated but also the actual flavour for each of the selected objects and the number of instances to be created for each selected VNF.

NFVI as a Service (NFVIaaS): the tenant (e.g., a vertical or an MVNO) is offered a virtual infrastructure including associated resources (networking/computing/storage) under its full control in which it can deploy and manage its own NFV network services on top of it. It is assumed that the vertical will deploy its own MANO stack. This is probably the most usual service consumed by M(V)NOs, given that they have the knowledge and need to customize their communication service offering to their own customers. Resources could be virtual cores, storage, virtual nodes and links, etc.

NOTE: The tenant can deploy and connect VMs on these resources under its own control.

NOTE: NFVIaaS includes the provision of network slices or network slice subnets as a service.

Network Service as a Service (NSaaS): provide to a tenant the possibility to define and instantiate a network service.

NF forwarding graph (NF FG): graph of logical links connecting NF nodes for the purpose of describing traffic flow between these network functions [24].

Physical Application (PA): implementation of a VA via a tightly coupled software and hardware system.

NOTE: analogous to PNF.

NOTE: may include devices such as cameras, smart city sensors, etc.

Physical Network Function (PNF): implementation of a NF via a tightly coupled software and hardware system. [24]

VA Forwarding Graph (VA FG): forwarding graph among VA, VNF, PA, PNF nodes.

Virtual Application (VA): more general term for a piece of software which can be loaded into a Virtual Machine. [24]

Virtual link (VL): set of connection points along with the connectivity relationship between them and any associated target performance metrics (e.g. bandwidth, latency, QoS). [24]

Virtualised Network Function (VNF): implementation of an NF that can be deployed on a Network Function Virtualisation Infrastructure. [24]

Virtualised Network Function Component (VNFC): internal component of a VNF providing a defined subset of that VNF's functionality, with the main characteristic that a single instance of this component maps 1:1 against a single Virtualisation Container. [24]

Virtualised Network Function Descriptor (VNFD): configuration template that describes a VNF in terms of its deployment and operational behaviour, and is used in the process of VNF on-boarding and managing the lifecycle of a VNF instance. [24]

VNF Forwarding Graph (VNF FG): NF forwarding graph where at least one node is a VNF. [24]

VS as a Service (VSaaS): provide to a vertical the possibility to define and instantiate a vertical service.

NOTE: similar to NSaaS, with vertical instead of network service.

8.3. Network slice related

Network slice (**NS**): a network slice is a complete logical network with specific services offered to customers over a shared compute, storage and network infrastructure. E.g. a network operator can build a network slice including an Access Network (AN) and a Core Network (CN) to enable communication services.

Network slice instance (NSI): a set of network functions and the resources for these network functions which are arranged and configured, forming a complete logical network to meet certain network characteristics [1].

NOTE: A Network slice instance is the realization of a network slice.

NOTE: In ETSI NFV parlance a network slice instance would typically be deployed as a NFV Network Service instance (NFV-NSI). Different slices can map to instances of the same type of NFV-NS with different deployment flavours or instances of different types of NFV-NS. In an NFV framework, creating a network slice will typically involve filling a NSD and requesting the NFV Orchestrator to instantiate a NFV-NS according to the contents of its NSD and selected deployment flavour.

Network slice subnet instance (NSSI): a set of network functions and the resources for these network functions which are arranged and configured to form a logical network (sub-network) [1].

NOTE:

- A NSI may include one or more NSSIs, which can include one or more VNFs or PNFs.
- A NSSI can be shared by multiple NSIs.
- Both NSI and NSSI can be mapped to a nested NFV Network Service.

8.4. Vertical service related

Vertical: 5Growth stakeholder belonging to a specific industry or group of customers and consuming 5Growth services (defined in Section 8.6). MVNOs are considered a special type of vertical in 5Growth.

NOTE: The existence of network slices is transparent to the vertical and it is fully under the control of the 5Growth Service Provider how to handle them, including, for instance, mapping services into network slices.

Vertical Service (VS): from a business perspective, it is a service focused on a specific industry or group of customers with specialized needs (e.g., industry 4.0, energy, transportation). The 5Growth system is designed to fulfil the requirements of vertical services coming from the 5Growth Service consumers (Section 8.6).

From a technical point of view, it is a composition of general functions as well as network functions and defined by its functional and behavioural specification.

NOTE: The vertical service behaviour is the result of the combination of the individual VA behaviours as well as the behaviours of the network infrastructure composition mechanism.

NOTE: A vertical service is similar to a network service, but provides more general functionality than network functionality.

Vertical Service Blueprint (VSB): a parameterized version of a VSD, where parameters have to be provided to provide a complete VSD, which is ready to be instantiated.

NOTE: There can be a wide range of parameters. The parameters can be used to express requirements of the vertical service, but also management related parameters such as file locations of virtual machine images or the priority of a service. A subset of parameters to

express requirements are: Bitrate of VAs and the connecting links, round-trip time among two VAs, geographical area to be covered by the vertical service.

Vertical Service Descriptor (VSD): a description of the deployment of a vertical service including service topology (constituent VAs and the relationships between them, Virtual Links, VNF Forwarding Graphs) as well as vertical service characteristics such as SLAs and any other artefacts necessary for the vertical service on-boarding and lifecycle management of its instances.

NOTE: A VSD may still contain instance specific parameters to be provided at instantiation time. This is similar to parameters provided at instantiation time of VNFs.

8.5. Multi-access edge computing related

The central concepts around edge computing (EC) are based on the definitions of ETSI MEC [37] and recent draft integrating NFV and MEC [40]. Following the renaming of mobile edge computing to multi-access edge computing, the definitions have been changed accordingly.

Multi-access edge application (MEA): application that can be instantiated on a multi-access edge host within the multi-access edge system and can potentially provide or consume multi-access edge services. [37]

Multiple-access Edge Application Orchestrator (MEAO): it has the same functions as MEO, excepting that it should use the NFVO to instantiate the virtual resources for the MEA as well as for the MEP.

Multiple-access Edge Host (MEC Host): it provides the virtualization environment to run MEC applications, while it interacts with the mobile network entities, via the MEP platform, to provide MES and offload data to MEA.

Multiple-access Edge Orchestrator (MEO): the MEO is in charge of the orchestration and the instantiation of MEA.

Multiple-access Edge Platform Manager (MEPM): it is in charge of the lifecycle management of the deployed MEA. The MEPM is in charge of the MEP configuration, such as the MEC application authorization, the traffic type that needs to be offloaded to the MEC application, DNS redirection, etc.

Multiple-access Edge Platform Manager – NFV (MEPM-V): the virtualized version of the MEPM delegates the LCM of MEA to one or more VNFMs, and keeps the MEP configuration.

Multi-access edge platform (MEP): collection of functionalities that are required to run multi-access edge applications on a specific multi-access edge host virtualisation infrastructure and to enable them to provide and consume multi-access edge services, and that can provide itself a number of multi-access edge services. [37]

Multi-access edge service (MES): service provided via the multi-access edge platform either by the multi-access edge platform itself or by a multi-access edge application. [37]

8.6. Business logic/stakeholder related

5Growth Service (TS): services offered by a 5Growth Service Provider (SP) to 5Growth Service Consumers, such as verticals, through the 5Gr-VS northbound interface or to other SPs through the east-west interface (EBI/WBI). As for the former, it includes the following four types of services: 5Growth Managed Vertical Service (MVS), 5Growth Unmanaged Vertical Service (UVS), NSaaS (sect. 8.2), and NFVIaaS (sect. 8.2). Additionally, SPs can also consume services offered by peering SPs. This interaction is done through the east-west interface (EBI/WBI) of 5GT-SOs, and so, it is not a slice-related interaction but an NFV network service-related one. There are two types of services offered in this way: federated services and federated resources (sect. 8.7). A service offered by a 5Growth Service Provider can include a bundle of such services.

5Growth Managed Vertical Service (MVS): vertical services that are fully deployed and managed by the SP and consumed as such by the vertical (i.e., without any interface available to modify the service logic, but only for getting operational information, at most).

5Growth Unmanaged Vertical Service (UVS): vertical services that are deployed by the SP (i.e., instantiating VNFs and their connectivity), but their logic is partly or fully managed by the vertical. This includes the configuration of VNF internals to control the logic of the vertical services at service level, e.g., the algorithms for EVS (Extended Virtual Sensing) for the automotive use case. In this case, the lifecycle management of the NFV-NS and its VNFs are still retained by the SP.

5Growth Service Consumer (SC): uses 5Growth services that are offered by a 5Growth Service Provider. Note that a 5Growth Service Provider can also be a SC of another service provider.

5Growth Service Provider (SP): provides 5Growth services. Designs, builds and operates its 5Growth services.

5Growth Mobile Transport and Computing Platform Operator (TMOP): in charge of orchestrating resources, potentially from multiple virtual infrastructure providers (VISP) and offered to the SP. In that sense, it acts as an aggregator of resources. The virtual infrastructure features transport and computing resources, potentially including those of datacentre service providers with which the TMOP has an agreement. Designs, builds, and operates the computing and network aggregated virtual infrastructure services. It has agreements with Virtualization Infrastructure Service Providers (VISPs) (see below for a definition).

Virtualization Infrastructure Service Provider (VISP): provides virtualized infrastructure services. Designs, builds and operates its virtualization infrastructure(s) [1]. VISP-T provides virtual transport infrastructure and VISP-C virtual computing infrastructure.

Data Centre Service Provider (DCSP)[31]: provides data centre services. Designs, builds and operates its data centers. [1]

[31] The difference between DCSP and VISP-C is that the former is closer to the raw resources (host servers) offering simple services of raw resource consumption. Additionally, these resources are located in a

8.7. 5Growth specific terms

Abstracted Resource / Resource abstraction: limited description of a resource with intention to hide certain parameters (such as quantity, vendors, location of the resource, etc.) and secure enough to be shared with other administrative domains.

Abstracted Service / Service abstraction: limited description of a service with intention to hide certain parameters (such as used resources, virtual links, interconnections etc.) and secure enough to be shared with other administrative domains.

Administrative domain: is a collection of resources operated by a single organization. The internal structure (collection of resources) operates with significant degree of mutual trust among them. The domain's resources interoperate with other administrative domains in a mutually suspicious manner and they are viewed as a cohesive entity from the outside.

Available Resources for Federation: set of resources offered by a provider domain under pre-agreed terms and conditions; available resources potentially to be used by a consumer domain, with certain pre-agreed terms and conditions.

Available Services for Federation: available services offered by a provider domain to other potential consumer domains, under pre-agreed terms and conditions.

Consumer domain: administrative domain that demands resources or services from other administrative domains.

Federated Resources: resources managed by a consumer domain, but owned by a provider domain (operator). The consumer domain is allowed (by the provider domain) to manage and use the resources based on pre-agreed terms and conditions (SLAs). In this case, the consumer SP uses NFV (abstracted) virtual resources offered by the peer SP. This may be the case when an end-to-end NFVIaaS service is built by combining virtual resources belonging to multiple SP administrative domains.

Federated Services: services managed by a consumer domain, but owned by a provider domain. The consumer domain is allowed (by the provider domain) to manage and use the services based on pre-agreed terms and conditions (SLAs). In this case, the consumer SP uses NFV network services offered by the peer SP. This may be the case when an end-to-end service is split into constituent services that are deployed in multiple SP administrative domains.

Federation: is a mechanism for integrating multiple administrative domains at different granularity into a unified open platform where the federated resources and/or services can trust each other at a certain degree [68].

centralized location (datacentre). The latter offers access to a variety of virtual infrastructure resources created by aggregating multiple technology domains and by making them accessible through a single API for all of them. For instance, VISP-C may offer not only centralized datacentre resources, but also distributed computing resources available throughout the network.

Local Repository: database (in an administrative domain) that holds information for available resources for federation, catalogue of services/abstracted services, provided by other provider domains.

Provider domain: administrative domain that offers resources or services to other administrative domains.

Technology domain: is a collection of resources that are part of a single technology (system) and belong to a single administrative domain. The internal structure is defined and operated according to the technology definitions and standards. One or more technology domains can be part of an administrative domain.

Service catalogue: composed set of services and/or service abstractions offered by a provider domain to other potential consumer domains using mutual taxonomy and agreed usage terms (SLAs). The composed service catalogue is shared among pre-agreed peering administrative domains.

Service Orchestrator (5Gr-SO): the service orchestrator component of the 5Growth system.

Vertical Slicer (5Gr-VS): the vertical interfacing component of the 5Growth system.

Resource Layer (5Gr-RL): the infrastructure layer component of the 5Growth system. It hosts all the compute, storage and networking physical and virtual resources where network slices and end-to-end services are executed.