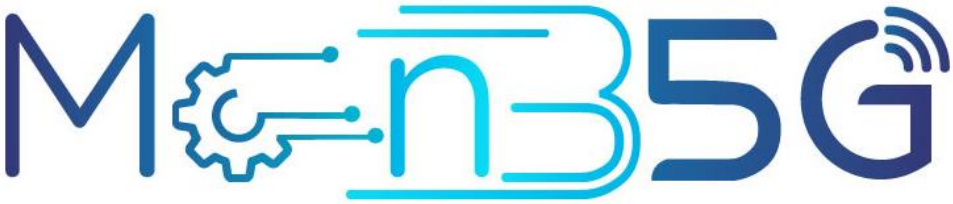

Deliverable D2.4
Final release of the MonB5G architecture (including security)

Document Summary Information

Grant Agreement No.	871780	**Acronym**	MonB5G
Full Title	Distributed Management of Network Slices in beyond 5G		
Start Date	01/11/2019	**Duration**	36 months
Project URL	https://www.monb5g.eu/		
Deliverable	D2.4: Final release of the MonB5G zero touch slice management and orchestration architecture		
Work Package	WP2		
Contractual due date	31/10/2021	**Actual submission date**	30/10/2021
Nature	Report	**Dissemination Level**	Public
Lead Beneficiary	ORA-PL		
Responsible Author	Sławomir Kukliński (ORA-PL)		
Contributions from	Sławomir Kukliński (ORA-PL), Lechosław Tomaszewski (ORA-PL), Robert Kołakowski (ORA-PL), Jolanta Fabisiak (ORA-PL) Adlen Ksentini (EUR), Sabra Ben Saad (EUR), Aiman Nait Abbou (AAL), Othmane Hireche (AAL), Chafika Benzaid (AAL), Tarik Taleb (AAL), Amina Boubendir (ORA-FR), José Jurandir Alves Esteves (ORA-FR), Anne-Marie Bosneag (LMI), Cao Than Phan (BCOM), Christos Tselios (CTXS), Zhao Xu (NEC), Lanfranco Zanzi (NEC), Francesco Devoti (NEC), George Guirgis (EBOS), George Tsolis (CTXS), Hatim Chergui (CTTC), Luis Sanabria-Russo (CTTC), Sarang Kahvazadeh (CTTC), Ioannis Chochliouros (OTE)		

Disclaimer

Copyright message

Contents

List of acronyms and abbreviations

Acronym	*Description*
3GPP	Third Generation Partnership Project
5GS	5G System
5GC	5G Core
ACT	Actuator
ACT-F	Actuator Function
ACT-S	Actuator Sublayer
AE	Analytic Engine
AE-F	Analytic Engine Function
AE-S	Analytic Engine Sublayer
AI	Artificial Intelligence
ANSSI	French Network and Information Security Agency
AP	Access Point
APEX	Adaptive Policy EXecution
API	Application Programming Interface
ASO	Autonomic Security Orchestrator
AUC	Area Under Curve
AUSF	Authentication Server Function
BS	Base Station
BSS	Business Support System
CD	Continuous Delivery
CFS	Customer-Facing Service
CHOP	Configuration, Healing, Optimization, and Protection
CI	Continuous Integration
CISM	Container Infrastructure Service Managers
CIS	Container Infrastructure System
CISI	Container Infrastructure Service Instance
CLA	Closed-Loop Automation
CM	Containerised Machine
CN	Core Network
CNF	Cloud-native Network Function
COE	Container Orchestration Engine
CP	Control Plane
CPU	Central Processing Unit
CSF	Cybersecurity Framework

Acronym	*Description*
CSMF	Communication Service Management Function
CTI	Cyber Threat Intelligence
CU	Centralised Unit
CVaR	Conditional Value at Risk
D-ASO	Domain Autonomic Security Orchestrator
DB	Data Base
DDoS	Distributed Denial of Service
DE	Decision Engine
DE-F	Decision Engine Function
DE-S	Decision Engine Sublayer
DHCP	Dynamic Host Configuration Protocol
DL	Deep Learning
DMO	Domain Manager and Orchestrator
DN	Data Network
DNS	Domain Name Server
DPI	Deep Packet Inspection
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
DSF	Domain Shared Function
DSF	Dynamically Shared Function
DSP	Digital Service Provider
DU	Distributed Unit
E2E	End to End
E2E-ASO	End-to-End Autonomic Security Orchestrator
ECA	Event Condition Action
EE	Energy Efficiency
EEM	Embedded Element Manager
EM	Element Manager
eMBB	Enhanced Mobile Broadband
ENI	Experiential Networked Intelligence
eTOM	Enhanced Telecom Operations Map
ETSI	European Telecommunications Standards Institute
FCAPS	Fault, Configuration, Accounting, Performance, Security
FL	Federated Learning

Acronym	*Description*
GANA	Generic Autonomic Network Architecture
gNB	gNodeB (next generation NodeB)
GS	Group Specification
GSM	Global System for Mobile Communications
GSMA	GSMA Alliance
GST	Generic network Slice Template
HTTP, http	HyperText Transfer Protocol
HTTPS, https	HyperText Transfer Protocol Secure
ID	Infrastructure Domain
IDM	Infrastructure Domain Manager
IDMO	Inter-Domain Manager and Orchestrator
IDS	Intrusion Detection System
IDSM	Inter-Domain Slice Manager
IFA	Interfaces and Architecture
ILP	Integer Linear Programming
IMSI	International Mobile Subscriber Identity
IOMF	Infrastructure Orchestrated Management Functions
IoT	Internet of Things
IP	Internet Protocol
ISF	Infrastructure Security Function
ISM	In-Slice Management
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union – Telecommunication Standardisation Sector
KPI	Key Performance Indicator
L-ASO	Local Autonomic Security Orchestrator
LCM	Lifecycle Management
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
MaaS	Management as a Service
MAC	Medium Access Control
MAE	Mean Absolute Error
MAN-F	Management Function
MANO	Management and Orchestration
MAPE	Monitor-Analyse-Plan-Execute
MEC	Multi-access Edge Computing

Acronym	*Description*
MEO	MEC Orchestrator
MitM	Man-in-the-Middle
ML	Machine Learning
MLaaS	MonB5G Layer as a Service
mMTC	Massive Machine Type Communications
MNO	Mobile Network Operator
MS	Monitoring System
MSE	Mean Square Error
MS-F	Monitoring System Function
MS-S	Monitoring System Sublayer
NBI	NorthBound Interface
NEST	Network Slice Type
NF	Network Function
NFV	Network Function Virtualisation
NFVI	NFV Infrastructure
NFVO	Network Function Virtualisation Orchestrator
NGMN	Next Generation Mobile Networks
NIST	National Institute of Standards and Technology
NS	Network Service
NSD	Network Service Descriptor
NSI	Network Slice Instance
NSMF	Network Slice Management Function
NSO	Network Service Orchestrator
NSP	Network Service Provider
NSSI	Network Sub-Slice Instance
NSSMF	Network Slice Subnetwork Management Function
NSST	Network Slice Subnet Template
NST	Network Slice Template
O-RAN	Open RAN
OAI	Open Air Interface
ODL	OpenDayLight
ONAP	Open Network Automation Platform
OPEX	Operational Expenses
OSM	Open-Source MANO
OSS	Operations Support System

Acronym	*Description*
PA	Policy Administrator
PaaS	Platform as a Service
PAP	Policy Administration Point
PBM	Policy-Based Management
PDP	Policy Decision Point
PDU	Protocol Data Unit
PE	Policy Engine
PEP	Policy Enforcement Point
PIP	Policy Information Point
PNF	Physical Network Function
PoC	Proof of Concept
PSF	Physical Security Function
PXE	Preboot Execution Environment
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RMSE	Root Mean Square Error
RMSLE	Root Mean Squared Logarithmic Error
RNN	Recurrent Neural Network
ROC	Receiver Operator Characteristic
RT	Real-Time
RTT	Round-Trip Time
RU	Radio Unit
SAP	Service Access Point
SBA	Service-Based Architecture
SDN	Software-Defined Networking
se-NSI	security enhanced Network Slice Instance
se-NSST	security enhanced NSST
SECaaS	Security as a Service
SECaaS	Security Service Platform
SecPaaS	Security Platform as a Service
SFC	Service Function Chaining
SFL	Slice Functional Layer

Acronym	*Description*
SLA	Service Level Agreement
SM	Sleep Mode
SM	Slice Manager
SMF	Session Management Function
SML	Slice Management Layer
SMO	Service Management and Orchestration
SMP	Slice Management Provider
SOD	Slice Orchestration Domain
SON	Self-Organizing Network
SRF	Slice Request Form
SRT	Short-term Risk across Time
SSLA	Security SLA
SVM	Support Vector Machine
TMF	TeleManagement Forum
TDD	Time Division Duplex
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TMN	Telecommunications Management Network
TVRA	Threat, Vulnerability and Risk Assessment
TSDB	Time Series Data Base
UC	Use Case
UDM	Unified Data Management
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UP	User Plane
UPF	User Plane Function
URLLC	Ultra-Reliable Low-Latency Communication
USL	Universal Scalability Law
V2X	Vehicle to Everything
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
VSF	Virtual Security Function
WP	Work Package

Acronym	*Description*
WWW, www	World-Wide Web
ZSM	Zero-touch network and Service Management

Executive summary

Future 5G networks are projected to support massive numbers of network slices working concurrently, which, together with the already high complexity of the network slicing solution, makes the tasks related to management and orchestration problematic. The elevated requirements for coverage, bandwidth and latency, as well as inter-domain operation, further exacerbate the complexity of network management, making already devised, standard, human-centric managing solutions insufficient and ineffective. The currently widespread centralised approach to network management also negatively impacts the separation and security of network slices, as well as the complexity of the central managing entity. Furthermore, centralisation also increases the overhead related to slice management data that has to be sent to the management system during the slice operation. Zero-touch management is perceived as one of the "key concepts" that can significantly contribute to the simplification of the human-based tasks for network slice management and orchestration. With the extensive usage of AI-driven mechanisms, its goal is to provide self-managed networks with little to no human interaction. The concept is currently heavily researched by the standardisation bodies.

This deliverable describes the final MonB5G zero-touch slice management and orchestration framework that aims to facilitate the deployment of a massive number of slices in different administrative and technological domains. An initial version of the architecture has already been described in MonB5G deliverable D2.1.

The proposed MonB5G architecture addresses the aforementioned issues by enabling functions distribution and providing strong separation of runtime management of network slices' and between orchestration domains. The concept facilitates self-managed slices composed of self-managed functions, further extended to slices created in multiple orchestration domains. The issue of management complexity is addressed by using AI at multiple levels of the management hierarchy to achieve specific management goals and to minimise interactions between architectural entities, e.g., by means of hierarchical closed-loop control, intents and aggregated Key Performance Indicators (KPIs). To that end, we have used a distribution of AI-driven management functions at multiple levels of the management hierarchy (node, slice, domain, and inter-domain). At all these levels, management-related operations are performed in a way, leading to the reduction of information exchange compared to the centralised solutions. The exchanged information between the management levels is becoming abstracted (intent-based), and finally, it leads (at the inter-domain level) to KPIs exchange.

The use of the AI-driven In-Slice Management concept (ISM), i.e., embedded slice management, has reduced the number of slice external interfaces (the management plane of a slice has become a slice template) and provides a perfect separation of the slices' management plane that cannot be achieved in the 3GPP approach to network slicing management. The implementation of slice management as a part of a slice (i.e., a set of Virtual Network Functions) provides better scalability of slice management performance and allows for the programmability of slice management services during slice lifetime. Moreover, the approach has made it possible to use the intent-based interfaces for the slice management by slice tenant or slice management provider. Following the ITU-T/TMN FCAPS approach, we treat the security as a management service (not as an external add-on). An emphasis is put on the security, management automation/programmability, and energy efficiency aspects of network slicing.

The MonB5G framework allows for the implementation of management as dynamically deployed PaaS (i.e., a set of orchestrated functions that are devoted to a specific slice template). Such an approach reduces MonB5G compliant slice footprint but provides weaker isolation of slices' management plane than the native ISM approach.

In terms of orchestration, instead of a single orchestrator-based approach, we have used a multi-domain orchestration with domain-specific orchestrators that are agnostic to network slices and can be seen as resource orchestrators. In MonB5G, the slice orchestrator is mainly focused on domain resources and is linked with its OSS/BSS that performs appropriate, resource-oriented functions. It is agnostic to slices orchestrated by it. Moreover, atop of the orchestrator, each domain has resource-oriented OSS/BSS that

is focused on runtime management of domain resources, involved in admission control, etc. Such placement of the domain OSS/BSS facilitates abstracted orchestration interfaces and provides loose coupling between the orchestrators (OSS/BSS is used as an interaction point with the E2E orchestrator) Moreover, such a modular approach enables the modification of the orchestrator without a need for the modification of other components of the architecture.

An important novelty of the MonB5G approach is the capability of ISM to trigger its slice modification request (typically based on slice-specific analysis), thus enabling proactive management operations and leveraging the agnosticism of the slice orchestrator. In the 3GPP and ETSI NFV MANO approach, such operation is triggered by the centralised OSS/BSS.

The MonB5G approach allows for resource brokering and energy-efficient operations. For that purpose, we have modified the existing interfaces between the infrastructure and other components of the architecture. We have also added the programmability of the infrastructure management by providing the capability to orchestrate its services using the MonB5G framework to the infrastructure provider.

The usage of AI in MonB5G architecture brings multiple benefits. First, the AI-enabled functions allow for local processing of information, therefore reducing the information exchange within the framework. It deals with both the monitoring and reconfigurations using the intent-based approach. AI is also used to analyse and interpret high-level performance-related information, i.e., KPIs. Moreover, the incorporation of the AI methods enables more accurate predictions regarding future demands and, as a result, makes reconfigurations less frequent and more accurate.

The split into separately managed and orchestrated domains reduces the overall management traffic. To achieve further reduction, the well-known, KPI-based approach to exchange for monitoring information between domains has been followed together with the intent-based approach for configuration changes.

Last, but not least, the presented approach includes a multi-actor business model in opposite contrast to ETSI MANO single actor-based approach.

All the mentioned MonB5G architecture capabilities contribute to the scalability and the flexibility of management and orchestration of slices.

In this document, the detailed description of the MonB5G architecture internal components and framework operations is presented, accompanied with instantiation proposals and implementation suggestions using commonly used open-source tools. Moreover, the scalability of the proposed approach is emphasized. This document presents a step towards the development of the architecture that could enable the implementation of network slicing solutions on a massive scale in commercial use cases, as well as create new business opportunities in the field of 5G (and above) network slicing and management and orchestration of telecommunication networks.

1 Scope

This deliverable reports on the activity of the MonB5G project's Work Package 2 (WP2), describing the final concept of Zero-Touch Distributed Slice Management Architecture. The preliminary version of the architecture has been described in deliverable D2.1. The present document includes the update of state-of- the- art regarding management and orchestration, standardisation efforts regarding network slicing, autonomic management, and the most important projects and their achievements related to slice management and orchestration in regard to D2.1.

According to our best knowledge, the proposed MonB5G architecture concept is the first one that addresses the scalability and robustness of network slicing management and orchestration by using a distributed and programmable management architecture. AI-enabled management operations are adopted at different levels of the management hierarchy. This novel approach to slice management has also been incorporated, i.e., AI-driven slice management functionalities can be embedded as part of a slice, providing higher elasticity in the creation and deployment of diverse slice types. The framework also provides a strong separation of concerns, which contributes significantly to complexity reduction and easier administration of slices, especially in the case of multi-domain slices deployed over different infrastructure domains belonging to several owners. Altogether, the above-mentioned features enable making a significant step towards self-managed network slices.

1.1 Deliverable structure

The structure of this deliverable can be summarized as follows:

- Section 2 describes the motivation that drives the creation of a new network slice management and orchestration approach.
- Section 3 is devoted to the description of the final MonB5G architecture with respect to the overall principles of the frameworks, as well as details regarding its internal components, mechanisms and operation.
- Section 4 presents the architecture instantiation proposals for different technological domains (cloud, RAN).
- Section 5 describes the operation of the MonB5G architecture in exemplary E2E scenarios, focusing on the creation of data pipelines, cooperation of monitoring entities, performing distributed analytics, as well as example procedures related to slice management and orchestration.
- Section 6 outlines the commonalities between MonB5G architecture and the ETSI Zero-Touch Service Management approach. Moreover, the scalability of the MonB5G architecture in the context of management and orchestration operations is described.
- Section 7 presents the remarks on the implementation of the MonB5G architecture.
- Section 8 concludes the document.

2 Motivation

5G introduces the use of virtualisation technology as a means to offer customised communication service capabilities over the same infrastructure by partitioning it into individualized slices [1]. In the future, 5G networks are projected to support massive numbers of network slices with different performance requirements, functionality and timespans [2][3][4] working concurrently, which together with the already high complexity of the network slicing solution, makes the tasks related to management and orchestration problematic. Realisation of the full benefits of virtualisation, cloud and edge computing requires the adoption of an autonomic management and control framework. An autonomic management framework is an important enabler for innovative business models. It represents not only a benefit for Mobile Network Operators (MNOs), but also enables customers to dynamically request and negotiate services and preferences for service customisation and personalisation. The telco cloud is expected to be programmable E2E, where the customer service requirements are supported by an appropriate underlying network slice with the necessary and sufficient allocation of compute, storage, and network resources. Cognitive and programmable capabilities enable "zero-touch" operation and maintenance of the network through automation for network and service planning, deployment, maintenance and optimization phases, delivering self-CHOP [5] (Configuration, Healing, Optimization, and Protection) qualities in a forward-looking system. The network and user equipment cooperate within the autonomic management and control framework for closed-loop automation and optimisation of system performance and behaviours while allowing in-deployment flexibility. Isolation, facilitated by underlying automated network slicing procedures, provides for improvements in security, privacy and fault tolerance, which are aligned with the self-CHOP characteristics of an E2E autonomic management and control framework. The characteristics of a self-CHOP enabled system reflects a fundamental shift from the relatively cumbersome silos of manual operational and configuration procedures to agile, programmable, and autonomic capabilities for automating system operation for performance and service optimisation in real time or near real time.

MonB5G aims to provide a new architecture to achieve scalable and automated management and orchestration of high numbers of parallel network slices envisioned in 5G and beyond. The MonB5G pillars are: (i) a highly distributed management and orchestration system, deployed over several entities involved in the lifecycle management (LCM) of network slices, namely MANO, NSMF, MEO, and the slice itself; (ii) data-driven mechanisms, based on distributed machine learning algorithms, to enable self-management and self-configuration of network slices, towards reaching the principle of scalable zero-touch network management.

2.1 Issues concerning management and orchestration

The network slice management differs from classical network management schemes, as it requires administrating not a single domain but multiple network domains. Current MANO solutions mainly focus on centralised approaches, which introduce scalability problems, especially in the presence of multiple administrative and/or technical domains as those envisioned in the network slicing context. In this scenario, the communication between the orchestration entity and the distributed networking entities will be characterized by significant delay and non-negligible traffic overhead, thereby preventing the implementation of standard polling-based monitoring processes. Any resource management decision should be compared to an almost real-time global view of the mobile infrastructure in order to avoid misconfigurations. To guarantee up-to-date monitoring information during the resource allocation process and allow online reconfiguration operations in response to unexpected network dynamics, an efficient and lightweight decisional process involving closed-loop feedbacks must be in place. The slice setup would likely affect resources spanning across multiple data centres and networking domains. The current MANO framework lacks mechanisms for managing all these attributes properly. The high variability dictated by the mobile network eco-system requires the orchestration process to be both location- and context-aware. Namely, it should be able to determine how the different networking functionalities operate, based on their location and exploit monitoring information to obtain detailed

reports on the current status of the multi-domain resource availability and/or utilisation, e.g., in terms of storing and processing data.

A network slice can be seen as the composition of a set of sub-slices belonging to different technological domains (e.g., RAN, transport, cloud, edge, core network). To overcome the centralised approach, each technological domain may be assigned with one (or more) management element, or agent, logically closer to the pool of resources to be orchestrated, thus enabling faster detection and reaction of domain-specific problems. The idea of having a hierarchical orchestrator is gaining momentum as an enabler for the flexible distribution of management tasks among entities belonging to different network domains. A hierarchical orchestration would enable adaptive function delegation supporting different levels of centralisation of monitoring, analysis, and decision-making tasks, based on the current operation status and the necessary degree of reconfiguration.

Given the hierarchical structure, management decisions could be taken at lower levels limiting the monitoring overhead and reducing the reaction time. When needed, monitoring information may be directly extracted and locally aggregated by the distributed agent, which will also be in charge of reacting to unexpected scenarios enforcing reconfiguration policies. If not able to resolve the situation, the agent will perform an initial inference task and provide the upper layers with more refined information (rather than raw monitoring data) to ease the problem solving and reduce the communication resource consumption. In this regard, the number of layers and timescale of information exchange among adjacent levels (horizontal and vertical) are particularly critical aspects to consider due to their impact on the overall capability of the system to promptly react to changes (e.g., in case of performance degradation or faulty operation of a network slice, as well as quickly identifying security-related issues). Therefore, finding the best combination of a number of layers and function delegation in the orchestrator hierarchy is crucial. As a starting point, a sub-optimal solution with a static number of available layers could be employed. However, we envision AI as the "key enabler" of such distributed management to devise an automatic solution able to optimally disaggregate and distribute the different management tasks to each hierarchical layer according to the current and future network status.

It should be noted that a distributed solution does not come with challenges itself. The use of controlled deep Reinforcement Learning[1] (RL) and other machine learning techniques for distributed management is challenging with regard to the distribution of the algorithm itself. An important point is how to perform optimal decisions considering the dependency that data have considering heterogeneous technologies/systems within each domain (e.g., taking optimal VNF placement decisions on the Edge domain in such a way to not generate degradation of E2E latency, but without complete information about Cloud and Edge domain state). Another significant issue is creating an appropriate strategy to dispatch sub-slices. In some cases, it may be beneficial to host some VNFs on the cloud to offload Edge data centres, while in other cases e.g., in the latency-critical scenarios, cloud deployment might not be a relevant option. Having an intelligent strategy to dispatch sub-slices and VNFs to the appropriate domain, together with the capacity to predict how requirements may evolve over time, is of utmost importance to ensure an optimal allocation of resources in each technological domain.

Scalability is a typical problem for network management. In the era of virtualisation, the management operations can be split between the orchestration (MANO) and classical management functions, where management does not have to cope with hardware. However, the network slicing paradigm can lead to serious management scalability problems. As a network slice can be seen as a network instance, the management has to cope not with one but with the unknown a priori number of networks to be managed. So far, in the NFV MANO approach, a single Operations Support System (OSS) is in charge of that, which raises several significant issues.

---

[1] Reinforcement Learning (RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences. For further information also see, e.g.: https://en.wikipedia.org/wiki/Reinforcement_learning.

First, a centralised approach to management causes very weak isolation of management planes of all the slices, which is a fundamental principle for slices isolation [6]. The lack of proper slices' management planes separation can raise severe security issues. Moreover, the access of tenants to the operator can be heavily impacted in terms of reliability and management performance as all slice-related operations are performed via one common entity. The performance can be further affected by the new network slice deployments as the dedicated slice management counterparts (slice runtime management plane) have to be added to OSS/BSS. A high degree of OSS/BSS centralisation also requires close attention in terms of OSS/BSS access, slice tenant authorization, etc., so as to preserve its secure operation. Moreover, the dynamic modifications of OSS/BSS have to be synchronized with the slice deployment, which increases the complexity of the whole process. Strong centralisation can also impact the slices that require rapid modifications as the round-trip time (RTT) increases proportionally to the distance between network nodes and OSS/BSS. In some cases, the additional delay cannot be neglected. Until now, the aforementioned issues related to OSS/BSS centralisation have not been addressed by the 3GPP, by any means.

The NFV MANO framework also severely suffers from the lack of a detailed specification of OSS/BSS functions. Until the ETSI NFV Release 3 [7], the functionality of OSS/BSS had not been defined at all. The currently specified OSS/BSS functions – user subscriptions handling, policy-based management of slices and services, Key Performance Indicators (KPIs) monitoring for Service Level Agreement (SLA) enforcement, accounting, etc. – are not sufficient, however, for advanced templates, i.e., beyond the 3GPP network. In the case of slicing, there are missing mechanisms for slice description and slice selection. Moreover, the approach to network slicing proposed by 3GPP is very complex in terms of functionalities of the system components, and their interactions. Dedicated network slicing components residing in 5G System (5GS) did not lead to framework simplification as the other components of the architecture have to be slice aware. Proper separation of concerns in the architecture could reduce the overall huge overhead of the network slicing. Further simplification could be achieved by the separation of resource and service orchestration and management for the purpose of making the orchestrator slice agnostic.

As already mentioned, the tenants' access to slice management is performed via a centralised OSS/BSS. In fact, the 3GPP management system architecture [8] allows the slice operator (tenant) to obtain selected management data and to subscribe to slice management operations [9]. Private management systems are not allowed and there exists no definition of the Tenants Portal. This entity is crucial in the context of slice-related operations in multi-tenant network slicing frameworks, as it serves for interactions between tenants and the system operator for the purpose of network slice service exposure, negotiation, ordering, and LCM [10]. Moreover, the NFV MANO model does not support business interfaces allowing for interactions between multiple business stakeholders (infrastructure providers, orchestrator operators, template providers, VNF providers).

The NFV MANO orchestration is automated *per se*, but some scalability issues pertain. No multi-orchestrator solutions are natively supported, and no individual blocks of the orchestrator have their performance monitored. The lack for cooperation with external third-party orchestrators can cause significant problems when 5GS is only a part of the deployed E2E solution (e.g., when the transport to remote servers and the deployment of virtual servers is performed out of 5GS [11]), which is a common occurrence as typically, each technological domain requires a dedicated orchestrator. Moreover, there is a limited impact on the behaviour of the NFV MANO orchestrator, e.g., it is not possible to have an impact on the way the VNF placement is carried out, (except for the Network Service Descriptor (NSD) flavour, and in general, automated runtime resource and other reconfiguration options are rather limited, inflexible, and monolithic (e.g., based on preconfigured rules). Proactive resource scaling has to be accomplished using iterative interactions between OSS/BSS and NFVO, which can lead to overconsumption of resources or slice KPI degradation due to operational inertia. So far, no mechanism exists that would allow for combining sophisticated algorithms with the slice-related information that can be obtained from the 5GS (e.g., information about the number of slice users, their location, their service requirements, UE mobility, slice load etc.) for the purpose of future resource consumption prediction and anticipation of resource allocation. Therefore, to support proactive resource allocation, the intelligent

orchestration that consumes the information from the control or application planes of a slice needs to be deployed.

Some issues also arise regarding ETSI NFV MANO and ETSI Multi-access Edge Computing (MEC) interoperation in the context of the VNFs deployment on the edge. Currently proposed integration option includes the co-operation of NFVO and MEC Orchestrator (MEO), which is usually not needed. In fact, in most cases, adding services to service-oriented slices during their runtime (i.e., one of MEC functionalities) is not needed if the service-oriented functions are already included in the slice template. Moreover, the inclusion of the service functions into a slice template provides multiple benefits including better service-network cooperation, higher isolation and security. High degree of separation of the service platforms makes the overall system operation much more complex. The example of an issue caused by the aforementioned approach can be observed in the context of the MEC Platform APIs, which are not slice aware. Moreover, several MEC and 5G Core (5GC) functions overlap such as, e.g., the authentication of users. Therefore, the integration of MEC with network slicing needs thorough and extended adaptation of both solutions, which is still discussed by the standardisation bodies as well as heavily researched by the academia [12][13].

The current approach to management does not fully benefit from the telco ecosystem virtualisation and softwarisation processes. Currently, NFV MANO has several limitations regarding the support for Continuous Integration/Continuous Delivery (CI/CD) approach[2] to the software development. So far, no interface is defined for the interactions between the VNF providers and the VNF operators. Despite some efforts from ETSI [14] to define the CI/CD workflow for NFV MANO, some elements are still missing such as e.g., the procedures for the VNFs validation (in terms of security, 3GPP compliance etc.) created as the result of CI/CD processes.

From an implementation point of view, however, choosing a centralised approach may have the advantages of a low delay in the setup of the slice, helped by the fact that the decision is taken by a single entity with a global view of the network. In order to allow a centralised entity to optimally perform its orchestration tasks, it is also important to define and design a complex monitoring system reporting updated information about the different domain resources as of the availability, consumption and congestion levels. Such information is crucial to reduce the probability of orchestration errors and enable efficient admission and control mechanisms. However, in complex and wide networks as those expected to support the future 5G ecosystem, it is practically unfeasible to transmit, with the adequate level of time granularity, the set of massive KPIs and monitoring metrics related to the high number of slices without introducing significant communication overhead, which will impact on data-plane transmissions, therefore, reducing the overall network efficiency.

On the one hand, the centralised decision entity represents a single-point-of-failure in the network architecture, which, in case of failure, can affect all the network management operations and lead to critical results. On the other hand, a centralised approach may introduce communication security and confidentiality drawbacks, as each domain would need to provide the decision entity, in a continuous way, with sensible monitoring information over the network.

MonB5G aims to provide solutions to the aforementioned scalability challenges and lack of autonomic management and orchestration mechanisms. MonB5G proposes to automate network management and orchestration by using **AI-based algorithms and distributed automated operations**. We do not assume that the solution is flat, but instead propose a **hierarchical approach**, which allows for flexible distribution of management tasks among entities at different levels of the hierarchy, while supporting **different levels**

---

[2] CI/CD is the cornerstone of true digital transformation. With CI/CD, delivery processes for new software versions and services can be automated – dramatically improving time-to-market and service agility. Continuous Integration/Continuous Deployment describes the key stages in an automated software development and deployment flow. This flow typically includes design, coding, testing, integration, delivery, validation and phased deployment activities before operation in a target environment. Also see: https://www.ericsson.com/en/ci-cd?gclid=EAIaIQobChMIxJvf-6XW8wIVT-J3Ch0qSwDQEAAYASAAEgKq_vD_BwE&gclsrc=aw.ds.

of centralisation by the optimal adaptive delegation of monitoring, analysis, and decision-making tasks, based on the current operation status.

2.2 Impact of decentralisation on network management

From the initial version of this deliverable D2.1, we recall that centralisation negatively impacts the separation and security of the network slices, while also increasing the **complexity** of the central managing entity, which in turn increases the **overhead** of the centralised slice managements.

A decentralised approach to network management is not just an evolution, but an essential enabler of 5G and beyond technology. Also, it was described in D2.1 was how a **distributed** network **management system** inherently increases **reliability**, **scalability,** and **security**.

Beyond these initially discussed advantages, decentralisation also enhances **flexibility**, **customisability,** and **adaptability** for all stakeholders in the ecosystem.

The premise of 5G and beyond technology is the enablement of a new level of Quality of Service (QoS) [15] that is tailored for the customers' segments-needs, where the segments as mentioned before are represented by virtual network slices that serve over-the-top streaming services, technology developers, infrastructure providers, etc.

Decentralisation enables the improvement of both **speed** and **flexibility** [16] within these networks (slices) through local control and execution of the needed services, thus *improving "the best case"*. This positively affects all of the network stakeholders, but more specifically, it facilitates **customisation** [17] for the service/content providers and the network slice providers by dynamically reorganizing to meet the varying demands and network conditions [16]. This *local dynamic management* is done automatically through algorithms, policies and goals which in turn facilitates local reconfiguration of the network (service) slices for the providers, and finally providing the needed **Quality of Service** (QoS) level for the *tenants* and *end-users*.

Distributed management also allows *fine-tuning* of slice operations, i.e., its configuration, state and functions, by continuous autonomic management. This enables **adaptability,** which is another advantage of the decentralised management of slices, as it provides continuous system-awareness that is able to cope with temporal and spatial changes in operational context that was not feasible (in real-time and with high granularity level) in centralised management [18].

As an exponentially growing number of businesses, developers, providers and users migrate towards 5G and beyond, the only model to support the new demands is the **distributed network management model**, for its own advantages and the new benefits provided to all the stakeholders.

2.3 Benefits of AI for network slices management and orchestration

5G introduces the use of virtualisation technology as a means to offer customised communication service capabilities over the same infrastructure by partitioning it into slices [19][20]. In this way, it is possible to satisfy the service requirements of different vertical industries [21]. The slices consist of a set of Virtual Network Functions (VNFs) that encapsulate specific sub-services that the slice needs to provide the service functionalities it was designed for. VNFs are mapped to physical nodes of the infrastructure, while the virtual links of the slice are mapped to physical links.

The slicing functionality is achieved by leveraging Software-Defined Networking (SDN) and NFV techniques [20][22][23][24][25]. A slice represents a virtual subset of the physical resources of the infrastructure that has been assigned to a tenant, which is the entity that reserves and pays for the resources of the slice. In this setting, the number of slices deployed simultaneously is expected to be very large and coordinating their deployment over the infrastructure will be extremely difficult for a human being, if not impossible. Since manual management and coordination are unfeasible, automation tools need to be deployed in order to achieve these tasks efficiently.

The use of Artificial Intelligence (AI) and Machine Learning (ML) as a key enabler for future networks has very early been recognised at European[3] and global level [26]. The identified challenges and corresponding opportunities given by AI/ML will affect different network aspects, layers, and functions and even create new requirements for the architecture of future mobile networks. AI and ML are broadly viewed as a class of computer and algorithm assisted intelligence modalities that mimic human intelligence at a task level, characterised by analysing a given set of data or observations, while determining an optimised solution to meet a desired objective. Deep Learning[4] (DL) is an augmentation of ML rendered through the use of multilayer neural network algorithms to flexibly handle a diverse array of complex use cases, associated with structured data[5] or unstructured data[6]. The functions across any layer of the core network, radio access network, the user equipment as well as the management and orchestration level are potential reference points to serve as a source of data (and events) or as a target for control, behaving as an input or an output respectively for a given ML function. The ML function optimises the operation of a particular entity in the 5G system, a larger part of the network or an associated service. The application of a specific learning model, hinges on the nature of the optimisation problem in the 5G system. A common framework of architectural building-blocks that are technology neutral is beneficial for harnessing a given ML function and its related interfaces, for technology-specific realisations.

AI is quickly becoming a key-feature in both network management and operational aspects of mobile networks. The wide availability of monitoring and operational data coming from heterogeneous networking domains allows gathering substantive insights on real-time networking processes. Decisions that previously took slow human interactions, based on traditional network characterisation and optimisation methods, can now be autonomously performed by ML algorithms with a holistic view of the network, enabling software components to directly contribute into decision-making activities related with the mobile network resource management.

AI is a natural candidate to automate the slice management tasks [27] since state-of-the-art research on AI has demonstrated the benefits it can achieve in this context. This not only improves the overall operational efficiency of the infrastructure, but also has significant impact into the reduction of management and energy related costs. Despite the general applicability of ML-based solutions, their practical application often relies on the possibility to access real-time data to perform analytics and diagnostics. Cognitive capabilities embedded within an autonomic management and control subsystem are realised in terms of the various modes of AI. AI and ML offer a variety of extensible methods to meet the connectivity, coverage, capacity, spectrum efficiency, energy efficiency and service demands of a virtualised, decentralised, distributed 5G system. ABI Research illustrated in its recent report that a spike in demand for AI/ML has been created in telco operations, which will enable distributed intelligence, enhanced efficiency and cost effectiveness [28].

A variety of the slice management tasks are investigated with leading edge AI techniques, including:

- **Slice admission control** [25][29][30]**, which in turn includes slice scheduling and slice collocation problems.** The fundamental issue in all these tasks is to ensure that VNFs of the slice and the virtual links between them can be mapped efficiently to physical resources in the infrastructure, making it a resource allocation problem [24]. The allocation of resources to the network slice in a timely and optimal manner is to ensure that the performance constraints of the slice, defined in its SLA

---

[3] See, e.g.: 5G IA (2020): Draft Proposal for a European Partnership under Horizon Europe Smart Networks and Services (version 30, June 2020). Available at: https://ec.europa.eu/info/sites/info/files/research_and_innovation/funding/documents/ec_rtd_he-partnership_smart-networks-services.pdf.

[4] Also see, *inter alia*: https://en.wikipedia.org/wiki/Deep_learning.

[5] Structured data consists typically of formatted, clearly defined and easily readable data types, such as names, geolocation, address, etc.

[6] Unstructured data consists typically of qualitative information that cannot be easily searched, analysed or processed, such as mobility patterns, free text, audio, video, satellite images, etc.

specifications, will be met. But mapping the performance constraints to SLAs and then to actual resource allocations may not be very straightforward, depending on the SLA specifications and the definition of the performance constraints. AI can also be useful in order to generate SLA specifications that represent better the performance needs of the slice and can also be used to drive resource allocation mechanisms [31].

- **In addition to slice admission control, it is necessary to also dynamically manage the slices, by readjusting their resource allocation to match their current demand, in order to optimise the resource allocation of the infrastructure** [1][23][24][32][33][34], thus truly bringing forward the benefits of network slicing. Proactive management of resource utilisation based on AI prediction methods is also a useful aspect. Other aspects of the slice management problem can also benefit from AI, such as monitoring, data acquisition, channel state prediction [35], security and fault detection, which all also need to be considered for 5G deployments [27].
- **AI can also be used for monitoring and prediction**, such as, inferring the operation status of the components of the infrastructure [27][35] traffic load of networks [22][29][33][36][37][38], where traffic forecasting is critical in the design of slice admission control policies and dynamic resource management mechanisms. In [22], the authors employ an Recurrent Neural Network[7] (RNN) for traffic forecasting, in order to drive the dimensioning and positioning of edge datacentres of the 5G infrastructure. In [36], the authors predict traffic using the Holt-Winters (HW) method[8], but only apply prediction on the subset of cells that are pre-classified using an Naive Bayes classifier[9]. This pre-classification is used to establish whether the traffic in the cell is in some way, predictable. Similarly, an HW predictor was used in [29] and [30], in the latter the authors also employed a Reinforcement Learning (RL) approach to modify the parameters of their traffic forecasting predictor in order to reduce SLA violations. In [37], the authors used an Long Short-Term Memory (LSTM)[10]-based traffic predictor to drive bandwidth reconfiguration among the slices. They modelled the bandwidth reconfiguration problem as a fractional knapsack problem [39].
- **Distributed AI allows for local processing and efficient compression of monitoring data that are exchanged between the functional components of the architecture.** Moreover, it also enables abstracted control, by the use of the intent-based reconfigurations.
- Last, but not least, **AI can be beneficial for improving the users' experience according to the services they access.** Given the current regulations for data privacy, it is not possible for service providers to accumulate users' data. Thus, a new AI technique emerged which trains agents for optimisation tasks without directly accessing users' data. This technique, called Federated Learning (FL) [40], works as follows. In the point where the users' data are being generated, an agent learns using the local data it has access to, but the data never leave the users' device. Once it learns, it sends its model parameters to a central agent that aggregates the models of multiple logical agents. This is done so, employed in order to increase the accuracy of the learned models. A lot of state-of-the-art research has been focusing on how to make FL more resource-efficient and more accurate [41][42][43][44][45], findings that are very relevant in the context of MonB5G as well. Moreover, FL has been used for

---

7 A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behaviour.

8 The Holt-Winters forecasting method applies a triple exponential smoothing for level, trend and seasonal components. A Holt-Winters model is defined by its three order parameters, alpha, beta, gamma. Alpha specifies the coefficient for the level smoothing. Beta specifies the coefficient for the trend smoothing.

9 In statistics, naive Bayes (NB) classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.

10 Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

many applications in mobile networks that improve users' experience, namely cyber-attack detection, edge caching and computation offloading, base station association [46], and predicting users' application demands [47].

Recently, reinforcement learning (RL), a family of AI methods for real-time dynamic-decision-making problems [48] has been widely investigated in the context of network management. RL is a promising tool for solving resource management and other optimisation challenges in 5G networks characterised by temporal variation and stochasticity of service and resource availability. Distributed variants of RL, such as Multi-Agent RL[11], can benefit the MonB5G architecture by introducing automation in MANO tasks in a decentralised fashion. Some of the benefits RL can provide are listed below.

- **Intra-slice reconfiguration.** Well studied RL algorithms could efficiently be used to reconfigure the VNF placement and chaining inside a slice dynamically [49]. Based on the capacity and load of connecting links and servers, the congestion level of local and alternative computing resources, and even KPI predictors from the distributed AEs proposed by the MonB5G project, RL could attempt to identify feasible local reconfigurations, affecting only a part of the chain. These reconfigurations could apply a specific policy, without the need for global reconfiguration or migrating the entire chain. This reconfiguration could also utilise proposed KPI estimates.
- **Inter-slice Reconfiguration**. RL implementations could be leveraged to simultaneously reconfigure slices. Inter-slice reconfiguration could benefit the performance of the network in cases of non-feasible reconfigurations of a single slice, or resource utilisation across slices, based on the available shared resources [50]. The large numbers of resources, when already fully allocated, could pose multiple challenges for reconfiguring one slice, because it can affect all the others and their respective SLAs. Also, different slices could overlap in their placement and compete for only a subset of common resources. That phenomenon creates complicated dependencies between slices. By using novel methods from the recent literature, RL methods could be developed to automatically learn and extract the key dependencies between slices. These dependencies could be performed both locally and globally to efficiently allocate radio, transport, computation, and storage resources between a substantial number of slices. RL could also offer a multi-objective approach to minimise the number of reconfigurations and SLA violations, maintaining almost optimal multiplexing gains, and optimising objectives such as cost, profit, or energy, as indicated by the service provider intent policies.
- **Power consumption reduction**. RL could introduce the concept of energy slicing, attempting to guarantee the required energy supply in different network domains and resources to satisfy various levels of SLAs [49]. It could be trained to maximise the energy efficiency in the network, by placing VNFs in the same physical machine when possible and switching off the idle servers.
- **Resource Allocation**. RL algorithm could be leveraged to predict the optimal computational or network resource ratio between slices and dynamically adjust the performance of each one [51]. As proposed, predicted KPIs could enable proactive resource allocation.
- **VNF and slice life-cycle managemen**t (instantiation, scaling, termination). RL could be used to decide when to scale in or out, instantiate, terminate or even duplicate a specific VNF, VNF chain, or a slice, according to its current traffic at the moment. KPIs could be used to proactively perform life-cycle management operations and orchestration.
- **Security and reliability**. RL could prevent specific types of attacks by detecting and acting rapidly, ensuring slice security and isolation. Decentralised data-driven management mechanisms could also be leveraged to adapt to the distributed architecture of MonB5G.
- **Prevent performance and service quality degradation**. RL could dynamically take decisions to maintain performance, either by migrating VNFs or changing the slice resource ratio to optimise the policy set by the operator.

[11] For further details about Multi-Agent RL also see: https://arxiv.org/abs/1911.10635.

These enhancements could consider multi-agent RL implementations, that can be distributed across the entire proposed architecture of MonB5G, acting in all layers.

By automating the slice management tasks, the AI techniques enable scalable network management, handling more slices in the same time scale for the same network resources. The scalability target is fulfilled by the AI-based solutions from three aspects.

First, the AI functions directly manipulate the networks without human intervention, such that the optimisation and control operations can be efficiently decided and deployed in requested time scales (often stringent) at RAN, transport, NFV infrastructures, and E2E network slices. Thus, the AI-based management automation provides latency improvements, and allows handling more network slices in specified time.

Furthermore, dynamic resource allocation fulfilled by AI techniques can avoid or largely alleviate the resource over-provisioning issues. The traditional method often sets a small utilisation threshold. Resources will be scaled vertically or horizontally if the predefined threshold is reached. The simple method works; however, it often causes over-provisioning, and cannot fully exploit the resources of the infrastructure. The AI-based resource allocation functions dynamically optimise the resource consumption based on the actual demand, and thus more network slices can be supported in the same infrastructure.

In addition, AI provides other enablers to further scale network management and orchestration, for example:

- Deep neural networks, especially auto-encoders, are typically used to learn data representations to compress the management data for communication overhead reduction, and in the meanwhile the learned low dimensional representations still preserve the intrinsic properties of the data, which then guarantees the analytics quality of e.g., network/VNF status and load.
- Feature selection techniques are also exploited to reduce communication cost and computational complexity of analytics. The E2E network slices are located at multiple domains, and often compose thousands of heterogeneous features monitored at network and application levels. With feature selection techniques, a finite set of most informative features will be identified to achieve similar or even better performance in network optimisation and analytics tasks, without the need of communicating and computing a massive volume of telemetry data.
- Finally, distributed ML techniques are another vital tool in scaling network management, though it is also employed for the privacy-preserving target. The aforementioned federated learning is a recent extension of the distributed ML methods. It does not make any assumptions on the local data, and thus allows for heterogeneous local data with highly imbalanced data size. A typical scenario of the distributed ML is to assign the training and inference mechanism from centralised architectures of Core/Cloud to Edge that is closer to the users. Thus, the latency introduced by data communication and execution of ML models in the Core/Cloud level can be largely improved. The distributed ML mechanism, including the recent advances of federated learning and distributed deep neural networks, enables network scalability by distributing and performing most of processing locally and rapidly.

Despite the hype of the previous few years, the adoption of AI/ML methods in cellular networks is still at its early stages. A lot of work is still needed to identify the most suitable solutions for the dynamic network management and control via AI/ML mechanisms. Ongoing research activities need to take into consideration diverse aspects, such as the availability and usability of data sets needed for specification and testing of AI/ML solutions, regulatory aspects and practical implementation issues.

The MonB5G project will tailor and develop state-of-the-art AI techniques in WP3, WP4 and WP5 along the specified directions to scale network slice management at all layers.

3 MonB5G architecture

As we have already outlined, a vast number of 5G radio access nodes deployed to satisfy ubiquitous coverage requirements together with increasing end-users' demand for low latency and high bandwidth envisioned for the 5G systems exacerbate the network management complexity with respect to previous mobile network generations, making human-centric managing solutions unfeasible and challenging the capabilities of other standard approaches. The network slicing concept further intensifies the problem. The overview that we provided made in deliverable D2.1 shows that there are few activities related to the distribution of management and orchestration for network slicing. Also, the programmability of the network slicing management has been ignored. We have noticed that most activities follow the ETSI MANO/3GPP approach in which slice management is centralised and common for all slices. As each network slice instance can be treated as an independent and isolated network, the approach brings the complexity of the overall management to an entirely new level.

The issues related to the ETSI MANO approach have already been described in section 2. The aim of the MonB5G architecture is to address these issues and provide a solution for the concurrent provisioning of high numbers of network slices as envisioned in 5G and beyond, through the relatively simple structure of the management system. The main goal of the MonB5G approach is to achieve scalable and automated management of multiple network slices. In this section, the initial outline of the concept will be presented.

3.1 MonB5G architecture principles

The MonB5G framework uses the management system decomposition that follows the ITU-T [52] and the MAPE (Monitor-Analyse-Plan-Execute) paradigms [53] as the basis. In our case, the MAPE concept is implemented in a distributed way by means of multiple AI-driven operations. Moreover, the runtime management of slices is distributed and programmable. We have also modified the NFV MANO approach slightly by distributing some of the orchestration functions.

The key features of the proposed MonB5G framework are the following:

- **A strong separation of concerns**. In the MonB5G framework, OSS/BSS of each orchestration domain is focused on the lifecycle management (LCM) of slices and on resource management of this domain, but it is agnostic to slices (i.e., it is not involved in slice runtime management). In MonB5G each single- or cross-domain slice can be seen as a service with its own management platform (called embedded or In-Slice Management, ISM [10]), which is separated from the domain(s) OSS/BSS. ISM is a part of the slice template and is responsible for the fault, configuration, accounting, performance, and security management (FCAPS) of a slice. That approach provides benefits such as isolation of management planes of slices (feature not provided by ETSI NFV MANO [54] nor 3GPP). Using this approach, the deployment of a slice requires marginal modification of OSS/BSS in order to support each slice management. In the case of multi-domains slices, a special inter-domain component (ISM) is added to the E2E slice template. It interacts with domain-level ISMs to achieve the E2E management of the slice.
- **Distribution of management operations**. The management operations are AI-driven and pursue different goals. The embedded management concerns nodes, slices, orchestration domains and the E2E slice. Using distributed AI allows for local processing of management information processing, thus reducing the exchange of management information between entities. The AI-driven approach also enables the use of intent-based interfaces.
- **Hierarchical, E2E slice orchestration.** In MonB5G architecture, there are multiple domain-level orchestrators (they can be domain-specific) and one master orchestrator. This implies the use of domain-specific slice templates. The use of multiple orchestrators contributes to orchestration scalability.
- **ISM capability of orchestration.** ISM of each slice may act as a service orchestrator, i.e., it may use the Os-Ma-nfvo-like interface [54] to request slice template modifications, and such action is no longer executed by the domain-level OSS/BSS. The request is typically based on the ISM analysis, and the action is related to slice topology update.

- **Scalable and programmable slice management**. Since ISM is part of a slice, and it is implemented as a set of VNFs, the resource scaling mechanism can contribute to ISM (i.e., slice management) performance. Moreover, all the FCAPS functionalities can be dynamically deployed or updated during slice lifetime using the orchestration capabilities of ISM.
- **Enhanced security of slices.** The use of the ISM concepts provides isolation of the management spaces of different slices, therefore contributing to enhanced slice security. It also limits information exchange between slices and OSS/BSS of each orchestration domain.
- **Support for Management as a Service (MaaS).** MonB5G allows the creation of a "management slice" that can be used for runtime management of multiple slice instances of the same template. In such a case, a new business player, called Slice Management Provider, can be involved in slice management.
- **Programmable, energy-aware infrastructure management.** The infrastructure management system proposed by MonB5G can use the architecture to deploy its services dynamically, in a similar way in which slices are deployed. The framework provides extensions to include energy-aware operations on infrastructure resources by the use of modified, energy consumption aware orchestration. For that purpose, the interface between the orchestrator and the infrastructure is provided.
- **Slice Tenant Portal and slice contract negotiation procedures.** The framework introduces the business portal entity that enables the Slice Tenants to request the deployment of slices based on the selected slice templates and perform lifecycle management operations on their slices. Moreover, the capability for slice contract negotiation procedures between the Slice Tenant and respective stakeholders (e.g., Infrastructure Providers) are facilitated.

The abovementioned features are in line with several ETSI ZSM requirements [55] – the list of the essential requirements of ZSM that are satisfied by the MonB5G management and orchestration framework is provided in section 6.1.

MonB5G introduces logical entities for monitoring, analytics, and decision making that are decomposed into **distributed, interacting components** executed at various levels: at the OSS/BBS level, inside the virtualised infrastructure, and embedded in slices. By local data processing and decisions, our design aims to: (a) minimize the exchange of (big) data between components to keep management scalable, and; (b) significantly reduce the reaction time of data-driven management decisions that could be handled locally. **Reducing the monitoring load** is critical for **carrier-grade performance** for a sliced beyond 5G network, a challenge that has not yet been adequately addressed. This approach requires, however, proper coordination of the "local" management subsystems.

The autonomic network management based on feedback loops that is proposed in MonB5G brings significant benefits, but also faces many problems. The most important problems are related to response times (associated with the round-trip time between network elements) and system stability (the managed system is a nonlinear one - the feedback-based control may lead to instabilities and chaotic behaviour). To this end, we propose a hierarchical control scheme with fast local control loops and slow wider-scope ones. Leveraging time-scale decomposition at different levels of the proposed system, we achieve to limit the interference among different feedback-based decisions. We also assume a rich multi-objective environment, where various goals (e.g., energy consumption, statistical multiplexing, slice isolation, etc. vs. performance) may have different weights, and the proposed algorithms should be able to automatically learn to prioritize accordingly. Moreover, we have also introduced the architecture components that are responsible for providing the feedback loop control stability evaluation and restoring.

3.2 Architecture outline

The **design philosophy** of MonB5G is to provide hierarchical, feedback-loop-based control for fault, configuration, accounting, performance, and security (FCAPS) management, and slice orchestration, featuring **different control loops with different scopes, goals, and timescales**, at the following levels:

- **Global OSS/BSS level**. At this level resides a centralised component with full-scope slice management and orchestration decision capabilities and takes global actions for network-wide, cross-slice, and cross-domain optimizations. The functions at this level are implementation-dependent – in more distributed approaches, OSS/BSS becomes simpler.
- **Technological/Orchestration Domain level:** Each technological domain (e.g., cloud infrastructure, edge, RAN, etc.) may operate its own instances of monitoring, analytics and decision functions. Undertaken AI-driven decisions aim at managing specific domain resources in the presence of coexisting slices, as well as carrying out optimisations to save on energy consumption without degradation of slices performance.
- **Slice level:** For each slice (called further Slice Functional Layer – SFL) we have proposed embedded AI-driven management, and advanced implementation of ISM, called Slice MonB5G Layer (SML). It can be a part of a slice template, or it can be provided in the form of a MaaS. The Slice MonB5G Layer is logically decomposed into Monitoring System Sublayer (MS-S), Analytic Engines Sublayer (AE-S) and the Decision Engines Sublayer (DE-S). Such decomposition enables the independent design of components of each of the mentioned sublayers that offer their services to other sublayers. All the sublayers may have AI-driven behaviour.
- **Node (Virtual Network Function/Physical Network Function/Cloud Native Network Function) level:** At this level, the control loops are implemented as part of modified Element Manager (EM), called Embedded Element Manager (EEM) that belong to SFL. EEM can take some node/function-focused, AI-driven decisions based on node-level monitoring. EEMs interact with SML as slaves, e.g., SML functions may leverage EEM endpoints for gathering data or actuation. The EEM information processing contributes to the reduction of the monitoring traffic and may also provide mechanisms for efficient and secure actuation.

The proposed framework assumes that the infrastructure may also need programmable management. To that end, we have proposed a separate infrastructure OSS/BSS (called **Infrastructure Domain Manager** – **IDM**) that manages the supporting infrastructure. Domain Manager and Orchestrators (DMOs) on the request of IDM can dynamically deploy management functions that cooperate with IDM to achieve efficient infrastructure management in terms of energy-saving and slice cost.

The MonB5G architecture is composed of **static and dynamically deployed components.** Altogether they provide support for operations related to slicing orchestration, fault management (self-healing), self-configuration, performance optimization (including energy-saving) and security-related operations of slices. The AI-driven In-Slice Management approach provides separations of management functions of each slice and gives the ability to manage slice(s) to the slice tenant in a simplified way. Moreover, the MonB5G management services can be deployed dynamically; therefore, they can also be orchestrated in a similar way to slices.

For all components of the MonB5G architecture, (i.e., static and dynamic), the same management philosophy (cf. Figure 1) has been applied. In this approach, the management system is composed of:

- **Monitoring Subsystem Sublayer** (MS Sublayer) which is responsible for collecting, aggregation and processing of the monitored data. The MS output can be consumed by other components of the architecture;
- **Analytic Engines Sublayer** (AE Sublayer) which is composed of multiple Analytic Engines that are focused on different goals;
- **Decision Engines Sublayer** (DE Sublayer) that is composed of multiple engines that are responsible for taking decision related to reconfigurations;
- **Actuators Sublayer** (ACT Sublayer, ACT-S) is responsible for converting the DE decisions into multiple atomic reconfiguration-related operations that simplify the reconfiguration and reduces the traffic between DE and reconfigured node(s). Moreover, the use of ACT makes DE decisions abstracted, and allows using intents for the communication between DE and ACT.

Components of the abovementioned layers cooperate to implement multiple feedback-loop operations.

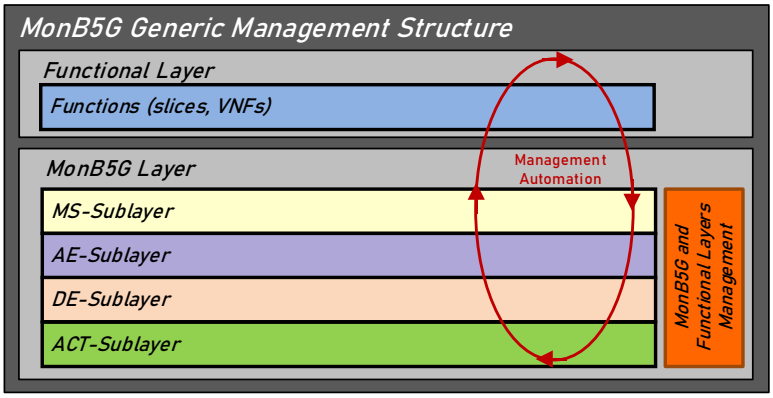

Figure 1. Generic view of MonB5G slice structure.

The MonB5G framework is composed of three layers:

- **Business Layer.** It consists of the business entities operating the framework, providing slice management services to slice tenants, or owning a slice (slice tenants).
- **Management and Orchestration Layer.** It consists of the core functions of the framework responsible for management and orchestration of slices, slices LCM and exposure of management interfaces to specific business entities.
- **Infrastructure Layer.** It consists of the infrastructure, infrastructure providers and functions enabling communication with Management and Orchestration Layer and enabling optimisation of usage of infrastructural resources.

The forthcoming sections provide a detailed description of each static and dynamic component of the framework belonging to Management and Orchestration and Infrastructure Layers. The description of business entities and their interactions is presented from a high-level perspective. A more detailed description can be found in [56].

3.3 Static components of the architecture

The static components of the architecture together with the business entities are presented in Figure 2, where, for the sake of clearness, slices are omitted. In the following section, a description of the static components and their main functionalities will be provided.

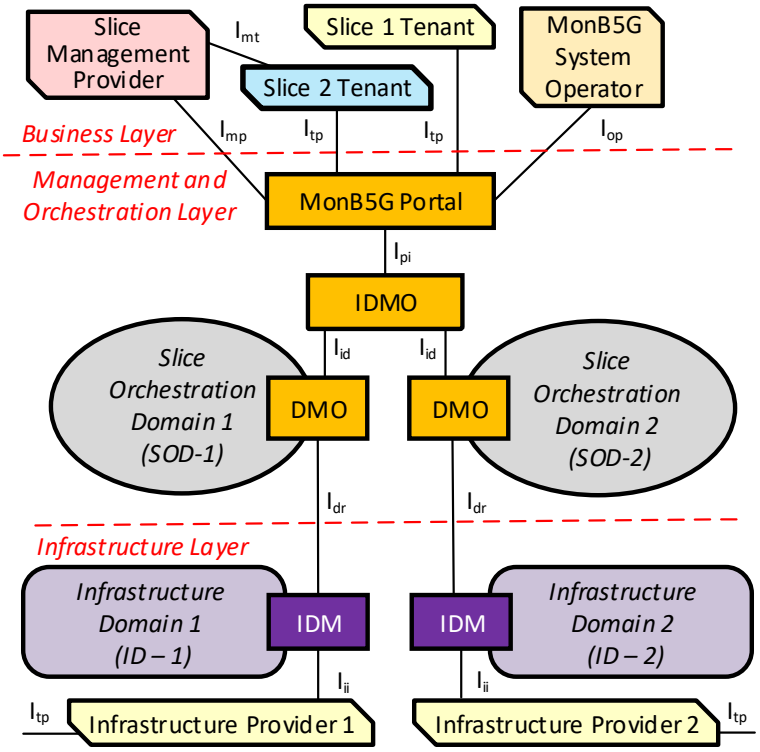

Figure 2. Static components of the MonB5G architecture.

3.3.1 MONB5G PORTAL

The MonB5G portal is used by Slice Tenants, Slice Management Providers, and Infrastructure Providers to request operations regarding slice LCM, i.e., slice deployment, slice modification and slice termination. The portal is also used by the MonB5G System Operator. It exposes the capabilities offered by the MonB5G framework (available slice templates, etc.) and partakes in negotiations related to the business dimension of the contract. The portal is also used to pass all the accounting and billing-related information. The internal structure of MonB5G Portal is presented in Figure 3.

The MonB5G Portal is composed of the following components:

- **Access Management** – an entity responsible for policy enforcement regarding users' access to MonB5G framework features, policy management and users' authorisation.
- **System Health Monitoring** – a component responsible for providing real-time high-level monitoring data showing the current state of the network for the MonB5G System Operator. In case of critical failures or instabilities of the system, the MonB5G System Operator, based on the accumulated monitoring data, can bring the framework back to stable conditions manually.
- **MonB5G Subscribers Database** – the database containing information of all entities having rights to access functions provided by the MonB5G system.
- **IDMO Connector** – the component responsible for communication with Inter-Domain Manager and Orchestrator (IDMO) described in section 3.3.2. The exchanged information includes, among others, slice LCM-related requests, contract negotiation, and high-level system monitoring data.
- **Slice LCM API** – the interface enabling the slice tenant to issue slice LCM-related requests, including slice templates selection from the Validated Templates Database, slice instantiation, slice termination, etc.

MonB5G Portal interacts with the Validated Templates Database for providing the slice tenant with the slice templates that can be instantiated by the framework. The slice templates can be selected from the ones provided in the Slice Template Repository or can be composed using the templates from SML

Repository and SFL Repository. The process of slice template validation and SML and SFL parts compliance check is out of the scope of the framework operation, i.e., it is performed by respective stakeholders before adding the template to the database.

MonB5G Portal exposes three northbound interfaces that expose the MonB5G framework capabilities to MonB5G System Operator (I_{op}), Slice Tenants (I_{tp}) and Slice Management Providers (I_{mp}).

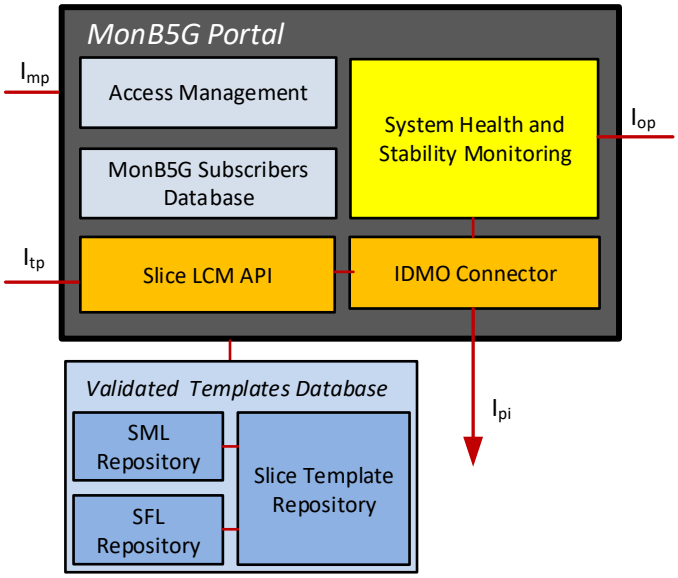

Figure 3. Internal structure of MonB5G Portal.

The MonB5G System Operator can interact with the MonB5G system via the MonB5G portal using I_{op} management interface. It provides monitoring capabilities as well as configuration capabilities of the overall framework. The MonB5G System Operator is responsible for controlling the health, security, and stability of the operation of the whole network. They can access the highest level KPIs that reflect the quality of operation and act accordingly to the obtained measures. All the operations are done via the I_{op} interface.

The I_{tp} interface, which will be typically implemented as a web-based interface, enables Slice Tenants to select and request slice-related operations. It can also be used by the Infrastructure Providers to ask for orchestration of infrastructure-oriented management functions. The procedure is performed in a similar way as slice-related requests of Slice Tenants.

Slice Management Providers may use MaaS platform, called MonB5G Layer as a Service (MLaaS), to offer management of multiple instances of slices based on the same template (as the slice runtime management is slice-specific). LCM of MLaaS is done via the I_{mp} interface. This approach will be described in detail in section 3.4.4.

To perform negotiations related to the business dimension of the contracts, MonB5G Portal interacts with IDMO via the southbound, I_{pi} interface. The exchanged information concerns aspects like availability of resources, existing policies, the resource demand, and other data that enables allocation of a certain number of resources to the requester. After the successful establishment of the contract, the I_{pi} interface is used for LCM of negotiated slices.

3.3.2 INTER-DOMAIN MANAGER AND ORCHESTRATOR

The Inter-Domain Manager and Orchestrator (IDMO) is at the heart of the system. This entity plays a crucial role in slice preparation and deployment phases by negotiation of deployment policy with a slice requester (Slice Tenants, Slice Management Providers or Infrastructure Provides). A schematic picture of IDMO with its internal components is presented in Figure 4.

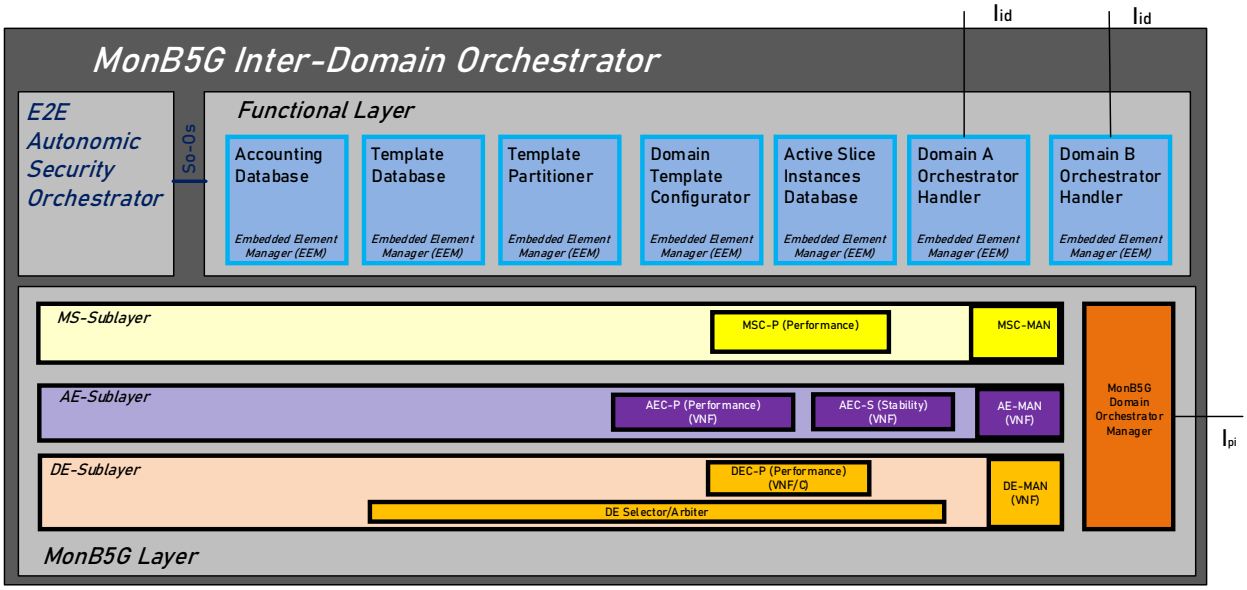

Figure 4. IDMO internal structure.

The structure of IDMO is decomposed into Functional and MonB5G Layers. The MonB5G Layer (management of IDMO) is AI-driven and uses sets of Monitoring System (MS)/Analytic Engine (AE)/Decision Engine (DE) as other components of the management architecture. The approach is described in detail in section 3.4.2. IDMO interacts with DMOs (see further) via I_{id} interface by using domain handlers to deploy the E2E-slices based on the information obtained from DMOs. It can be seen as an E2E orchestrator (umbrella orchestrator, according to [57]).

The Functional Layer of IDMO consists of the following functional entities:

- **Accounting Database.** It keeps all the accounting information, also historical data. The accounting process considers (among others) resource consumption and SLA fulfilment. The data are collected till the termination of each slice.
- **Template Database.** It keeps all templates that can be used by the MonB5G system. It is assumed that the database is dynamically updated. A template in this database is, in general, an E2E template that can be composed of domain-specific templates and used to create the E2E slice.
- **Template Partitioner.** It is responsible for partitioning of a template in case the template will be deployed in multiple domains of the same type. Such partitioning can be motivated by a lack of resources in a single domain for the deployment of the whole template, in the case where different domains cover different geographical areas, for the sake of security (only part of a template is deployed in each domain) or by economic reasons. The decision about partitioning is taken through the cooperation with the Resource Broker.
- **Template Configurator.** Each domain template before its deployment must be configured appropriately before its deployment. This process deals with the initial configuration of slice parameters but also with the configuration of interactions between sub-network slices (parts of the E2E slice that are deployed in a single domain). The modified slice template includes mechanisms added to slice template by IDMO for slice stitching to obtain the E2E slice and proper modification of the E2E slice management plane (correlation of events and KPIs from different domains that are used for slice deployments).
- **Active Slice Instances Database.** Active Slice Instances Database consists of information about deployed slice instances, their identifiers and status (configuration, KPIs, faults, resource consumption).
- **Resource Broker.** It is involved in cases when there are multiple Infrastructure (resource) Providers – IDMO is aware of all infrastructure domains involved in the system and the status of

their resources. The Resource Broker can be involved in negotiations related to the selection of cost-effective deployment of a slice instance. The Resource Broker interacts with Infrastructure Providers indirectly via DMO. In cooperation with Template Partitioner, it may find an optimal slice deployment strategy across multiple orchestration domains and Infrastructure Providers. The broker may consider the energy consumption related to slice instance deployment.

- **Handlers for all Orchestrators.** They are responsible for the interactions with orchestrators of each domain that is used by the MonB5G System operator. Their role is to handle all the processes related to slice LCM (by controlling the progress of each operation) and collection of each domain events (SLA violations, faults, etc.).
- **E2E Autonomic Security Orchestrator (E2E-ASO).** It is responsible for the maintenance of secure operation of a slice from the E2E perspective. The details regarding E2E-ASO are described in section 3.5.1.1.

In the "legacy" implementation of network slice orchestration and management, IDMO may play a role of Communication Service Management Function (CSMF) and Network Slice Management Function (NSMF); for MonB5G compliant slice templates, IDMO is an orchestration part of NSMF only (CSMF and NSMF are described in more details in [8]).

MonB5G architecture also envisions the scalability of IDMO's functional layer by enforcing redundancy. Multiple functional layers may be present in the architecture to provide resilience against failures or information loss. (IDMO) Leader and Followers may be subject to quorum quotas so as to ensure Leader IDMO data and other capabilities are preserved by a minimum number of Followers. IDMO clusters may then follow resilient and well-studied methods for Leader selection and overload prevention[12], allowing administrators (or MonB5G AI/ML-based components) to grow or shrink the number of IDMO Followers according to the number of managed slices.

3.3.3 DOMAIN MANAGER AND ORCHESTRATOR

Domain Manager and Orchestrator (DMO) is responsible for orchestration and management of each of Slice Orchestration Domain (SOD) slices. The internals of DMO are presented in Figure 5. DMO can be seen as a combination of resource-oriented OSS/BSS, an orchestrator (a MANO orchestrator is shown in the picture – in other technological domains, other orchestrators may be used) and a Domain Autonomic Security Orchestrator (D-ASO), described in detail in section 3.5.1.2. The behaviour of all components is optimised using AI. OSS/BSS is tailored to cope with domain-specific management. It is focused on slice lifecycle management (including slice admission control) and resource management (FCAPS of resources). OSS/BSS of DMO can be used for all external interfaces of DMO. It must be noted that IDMO does not interact directly with the orchestrator but with OSS/BSS of each SOD. Therefore, the IDM-IDMO interface can be defined in a similar way (abstracted) for different orchestration technologies. DMO is focused on SOD operations concerning resources (resource allocation to slices, slice LCM, resources FCAPS) and is agnostic to slices, i.e., it is not involved in slice runtime management, including initial slice configuration. Therefore, DMO generally deals with the software dimension of slices (LCM, resource scaling) or allocation of PNFs to slices. In contrast, the runtime management is handled by the management components embedded in slices (i.e., ISM). Similar to IDMO, all DMO operations are AI-driven. Therefore, the internal structure of DMO is also composed of Functional and MonB5G Layers. The operations related to resource management as well as the exchange of infrastructure-related data (about energy consumption) are done via I_{dr} interface. Part of DMO is resource-oriented OSS/BSS and another part is an orchestrator. In most cases, such an orchestrator will be an ETSI MANO compliant orchestrator (ETSI OSM has a version with OSS/BSS).

[12] Etcd Learner architecture and procedures: https://etcd.io/docs/v3.3/learning/learner/.

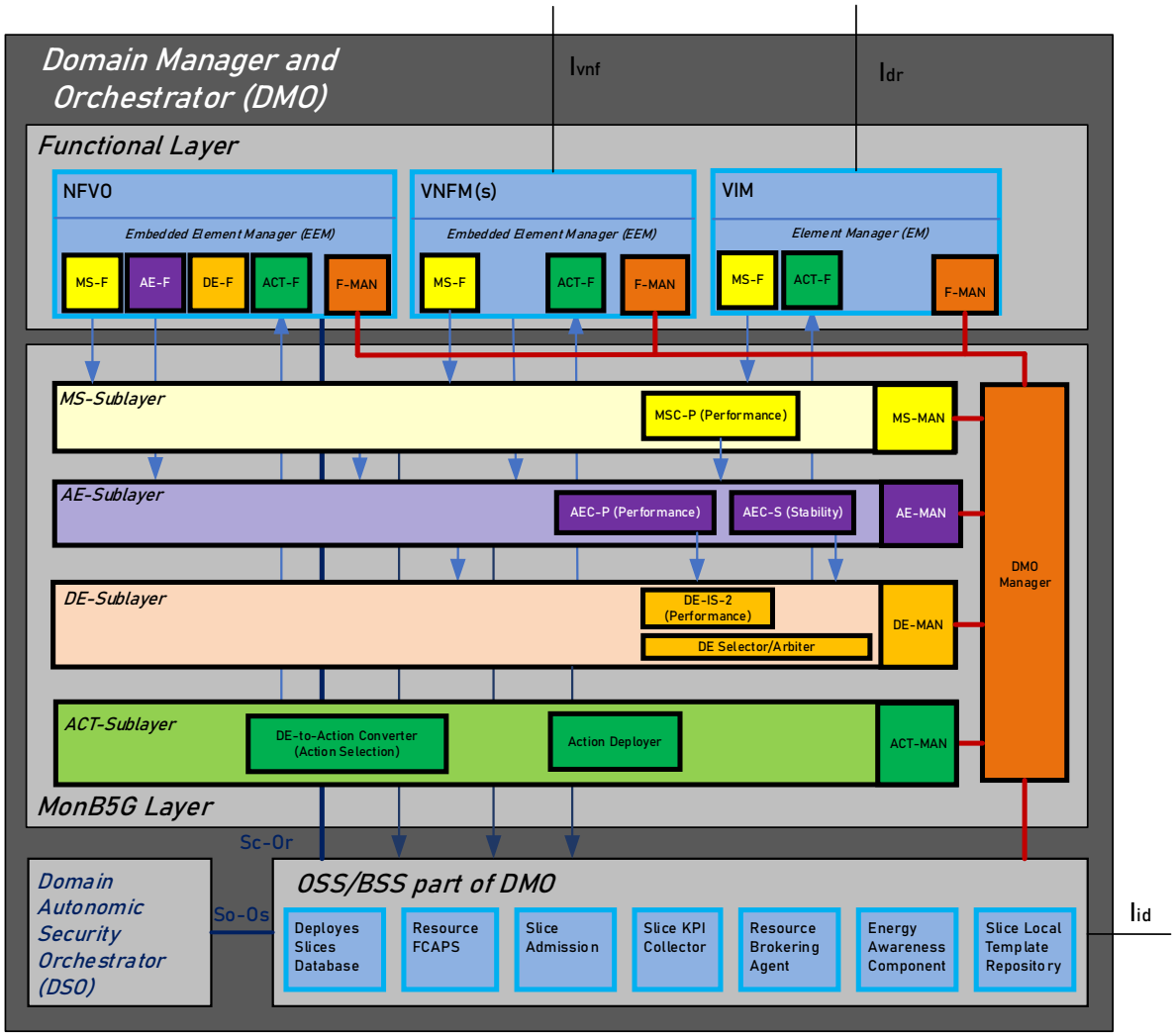

Figure 5. Internal architecture of the Domain Manager and its interactions (MANO case).

In contrast to the 3GPP approach in which management of all slices is part of OSS/BSS, DMO is not involved in individual slice management. In MonB5G, it has been decided to keep the orchestration part agnostic to slices; therefore, the runtime management of slices is performed by different components than the orchestrator. DMO keeps the repository of slice templates that can be deployed in its domain; this repository can be updated by IDMO if specific slice template is not available in a specific domain. The IDMO-DMO-IDM interaction, as it has been already mentioned, is used by IDMO for resource brokering decisions to split the slice and deploy it on the infrastructure belonging to multiple owners.

In the "legacy" implementation of network slice orchestration and management, DMO may play a role of NSSMF (described in detail in [8]); for MonB5G compliant slices, it is the orchestration part of NSSMF only, not involved in slice runtime management.

3.3.4 INFRASTRUCTURE DOMAIN MANAGER

The proposed framework assumes that the Infrastructure also needs management, and such management will benefit from its programmability. To that end, we have proposed a separate management entity called **Infrastructure Domain Manager – IDM** presented in Figure 6. IDM provides the overall management of the Infrastructure of specific orchestration domains. Its interface to DMO allows for the allocation of resources via Network Functions Virtualization Infrastructure (NFVI) agent, exchange of the information related to the energy consumption of resources, and cost of resources that IDMO can use for resource brokering. DMO can dynamically deploy management functions that cooperate with IDM to achieve programmable infrastructure management. IDM has an interface to the Infrastructure Provider, who can use the MonB5G portal asking for the deployment of additional infrastructure management

functions, called IOMFs (see further). The functions are orchestrated in a similar way to slices, and LCM requests are sent by the Infrastructure Provider to the MonB5G Portal.

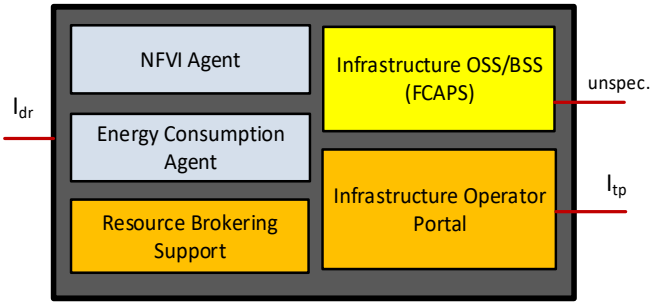

Figure 6. Internal structure of IDM (an example).

The details of IDM are out of the scope of the MonB5G framework; however, in the case of MANO, the minimal implementation of IDM should include the NFVI Agent, the Energy Consumption Agent (energy data collection and exposure to Infrastructure Operator), Resource Brokering Support (functions facilitating resource provision from multiple Infrastructure Providers), Infrastructure Operator Portal (the exposure of Infrastructure Management mechanisms to the Infrastructure operator) and Infrastructure-oriented OSS/BSS (used for the purpose of Infrastructure FCAPS).

3.4 Dynamic components of the architecture

The dynamic components of the architecture are slices that are defined in a different way than NGMN Alliance has defined them. In the MonB5G approach, a slice is a set of functions that implement a specific goal; however, they do not have to implement a network as such. It can be a set of functions that implement a specific aim, for example, network management, implementation of services or accelerators that support certain operations of multiple slices. Interactions between such slices and classical slices can be implemented using the PaaS approach. The approach is sometimes called vertical stitching of slices in opposition to the horizontal stitches of single-domain slices in the case of a multi-domain slice. How the MonB5G framework is using PaaS and the benefits of the approach will be described later.

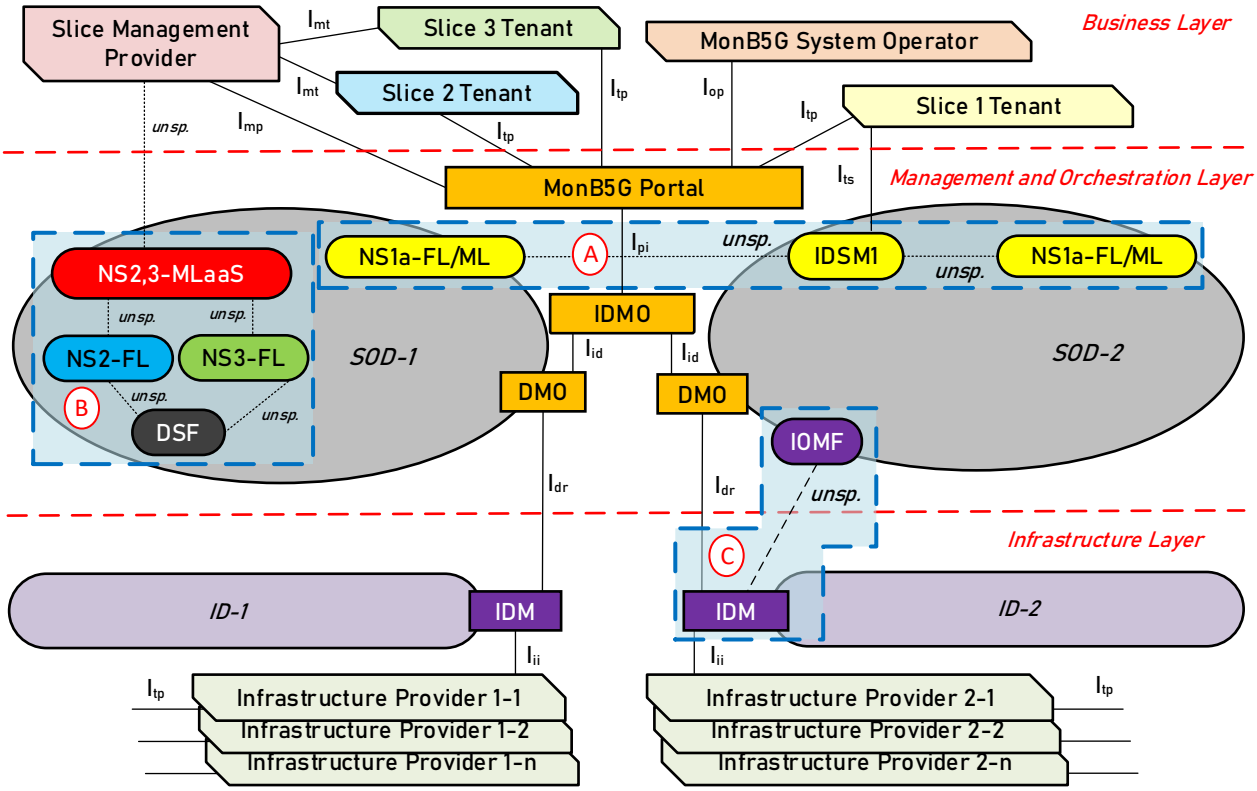

Figure 7. Overall MonB5G management and orchestration framework

The MonB5G follows the ISM concept in which the slice management plane is a part of the slice template. Therefore, the slice management is implemented (with one exception that will be described later) as other components of the slices, for example, a set of VNFs. Such an approach leads to the creation of self-managed slices and the reduced information exchange between the slices and the external management components of the architecture (i.e., DMO and IDMO). The usage of PaaS and the need for the creation of slices that span multiple domains leads to several slice deployment options, as presented in Figure 7.

The Option A of Figure 7 concerns the deployment of a self-managed multi-domain slice. Such type of slice requires a special component that is responsible for E2E slice management – it is worth noting that this component is implemented as a part of the E2E-slice template, not as a part of DMO. Option B shows the deployment of slices that use the PaaS approach, i.e., shared functions that implement the Management as a Service (MaaS) paradigm. Another set of shared functions, DSF (see further), is in this option exploited by the functional part of the slice. Option C shows infrastructure management-oriented and orchestrated by DMO functions that are created in a similar way to slices, on the request of the Infrastructure Provider. The components that are deployed within each option are described in detail in the forthcoming sections.

3.4.1 MONB5G SLICE STRUCTURE AND FUNCTIONS

The generic structure of the MonB5G slice is presented in Figure 8. In the MonB5G slice structure, two separate layers can be distinguished – the slice management part called Slice MonB5G Layer (SML) and the slice main part called the Slice Functional Layer (SFL).

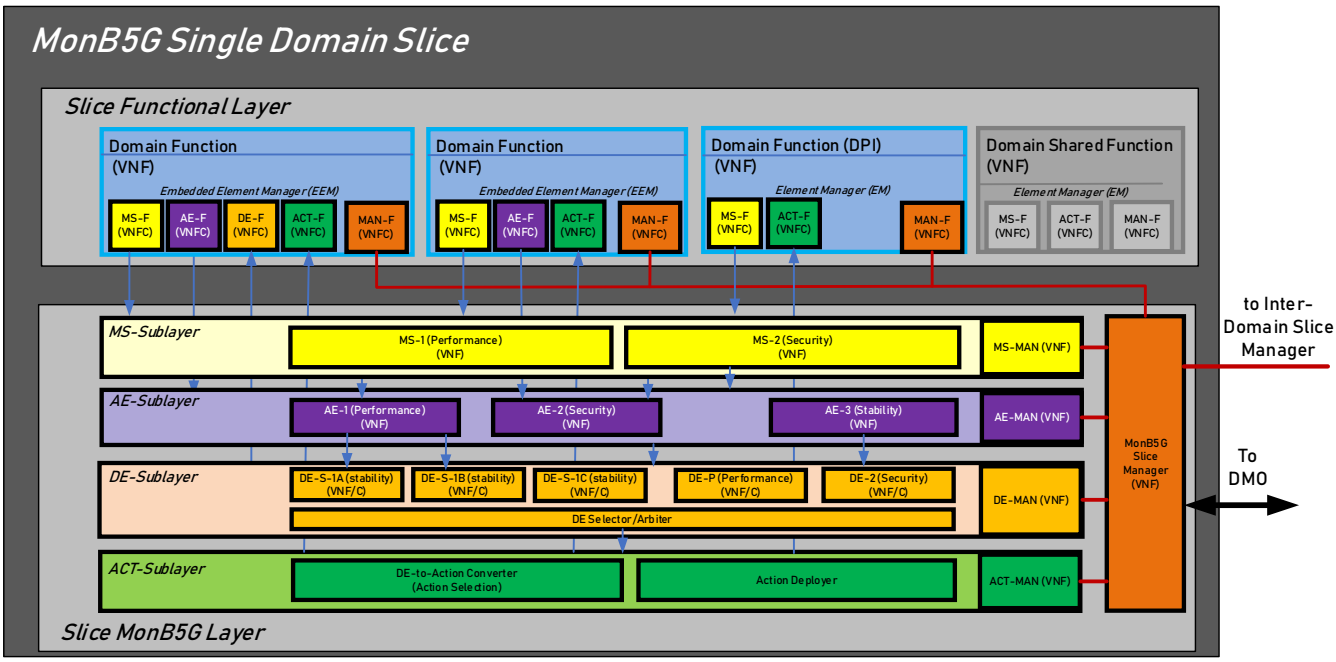

Figure 8. Generic structure of MonB5G slice template (example).

SML is composed of the four layers responsible for monitoring (MS-Sublayer), performing slice-related analytics (AE-Sublayer), making reconfiguration decisions (DE-Sublayer) and slice components actuation (ACT-Sublayer). Manual slice management can be performed via MonB5G Slice Management entities. The detailed description of the Slice MonB5G Layer internals will be presented in section 3.4.2. SML performs FCAPS at the slice level and can be considered as an embedded slice-level OSS/BSS, with interfaces to the Element Managers (EMs) of the slice's VNFs/PNFs or Cloud-native Network Functions (CNFs), and to DMO. SML is a part of the slice template or is deployed independently using PaaS/MaaS paradigm. In such a case, SML is implemented as an independent slice that can manage multiple instances of SFLs (see further) of the same template.

3.4.1.1 SLICE FUNCTIONAL LAYER

The Slice Functional Layer (SFL) contains a set of virtual functions that form the network slice to be deployed. The SFL part is composed of virtual functions that are dedicated solely to a slice (they are included in the slice template). However, SFL can also use functions that are shared functions available in a SOD. Such functions may be used by all or some slices. These functions are called Domain Shared Functions (DSFs) and can be implemented as PNFs/VNFs or CNFs. In fact, this is a PaaS approach used for SFL. The use of DSFs provides a reduced footprint of the deployed slices improving that way also the slices deployment time. DSFs are grouped (i.e., form a slice) for the purpose of their management. They are managed by DMO.

According to the ITU-T and ETSI MANO model, each of the functions (i.e., VNF) should have an Element Manager (EM) that typically interacts with OSS/BSS. In the MonB5G case, EM is replaced by MAPE-based Embedded Element Manager (EEM) that is implemented as a component of a functional entity (e.g., VNF-C).

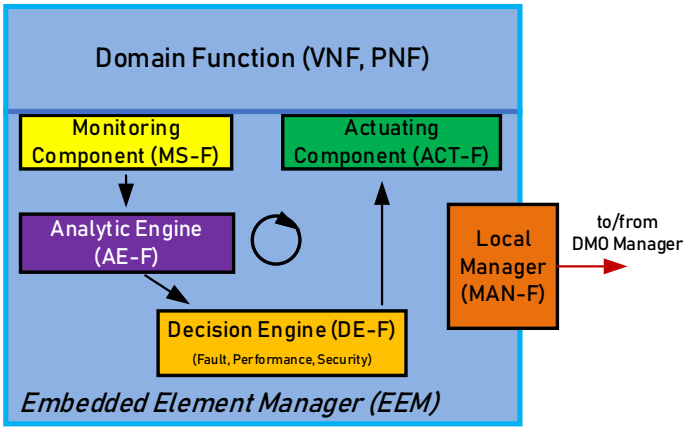

Figure 9. Typical interactions between EEM components.

EEM is internally split into components responsible for its VNF monitoring (MS-F), anomaly detection (AE-F), decision engine (DE-F) and actuating component (ACT-F), as presented in Figure 9. MS-F is used for the monitoring of the function/node behaviour, AE-F is looking for potential anomalies in the function/node behaviour, DE-F is taking decisions concerning function/node reconfiguration, and finally, the ACT-F component converts the DE-F decision into a set of low-level commands. All the entities cooperate to achieve MAPE behaviour at the node level. EEM also includes the management component (MAN-F) that is used for the external management of the component, in order to validate function/node MAPE decisions or to override them. For the backward compatibility, it is assumed that some network functions/nodes may not have EEM; therefore, EM in such a case must be used.

The usage of EEM reduces the management-related traffic significantly and introduces self-managed functions/nodes in the management architecture hierarchy. The usage of AI for AE-F or DE-F is dependent on implementation; the EEM footprint has, however, to be kept small. The monitoring information pre-processed by EEM is fed to the SML part of the slice. Depending on the specifics of SOD, the SOD functions/nodes allocated to slices can use different orchestration approaches. In a non-virtual environment, EEMs (i.e., RAN slicing in 3GPP Release 15) must be implemented as a part of SML in a virtual environment. In general, EEMs are the links between SFL and SML parts of a slice. It is expected that the interaction can use a message bus or APIs.

3.4.2 SLICE MONB5G LAYER

It is an implementation of the ISM concept that uses AI-based MAPE management. Therefore, the SML components are: the Monitoring System Sublayer (MS-S), the Analytic Engines Sublayer (AE-S), the Decision Engines Sublayer (DE-S) and the Actuating Functions Sublayer (ACT-S). Each component of MS-S, AE-S, DE-S and ACT-S sublayers can be managed from SML by individual SM entities, namely MS-MAN,

AE-MAN, DE-MAN, and ACT-MAN, respectively. The implementation details regarding MS-S and AE-S will be described in WP3 deliverables, while DE-S is in the scope of WP4.

SML can adapt its functionality according to the specific slice type and technological domains involved in the corresponding deployment. As an example, in the case of single domain slice deployments, SML may play the role of CSMF, NSMF and NSSMF, or, in the case of multi-domain slices, it can inherit the runtime functionalities of NSSMF. Please note that SML is a part of the slice template and therefore, it is not generic but tightly coupled with SFL. The internal functions of SML and their interactions are therefore not standardised ones. Nonetheless, some examples of the metrics that can be used for slice operation are included in the descriptions of AE-S and DE-S (cf. sections 3.4.2.2 and 3.4.2.3), as well as some generic AI algorithm evaluation metrics have been gathered (cf. Appendix I: Metrics for AI algorithms evaluation). In the following subsections, generic functionalities of SML are therefore presented.

3.4.2.1 MS SUBLAYER OF SML

The **MS Sublayer (MS-S)** depicted in Figure 10 is in charge of providing generic and reusable monitoring information to AEs, DEs, and other SML entities, as presented in section 3.2.

The monitoring information may be consumed by each SML entity. MS of SML is responsible for the collection, aggregation, filtering, and interpolation of the monitoring data related to a slice (that includes resources monitoring). It is also responsible for the calculation of slice KPIs and collecting information about faults, topology changes and so on. To meet the heterogeneous requirements of each SML entity, time granularity and degrees of data aggregation are tuneable parameters in the definition of the specific data collection and pre-processing strategy. MS of SML collects the monitoring information from EM and EEM and, in a generic case, is composed of the following blocks:

- **Monitoring Information Collector/Aggregator** –, an entity which interacts with the EM/EEMs of SFL;
- **Monitoring Information Database** – a database in which collected monitoring data are stored in raw and/or pre-processed format;
- **Monitoring Information Processor** – an entity that is responsible for filtering, interpolation and prediction of the monitoring data;
- **Monitoring Sublayer Manager** – an entity that allows remote configuration of MS sublayer operations.

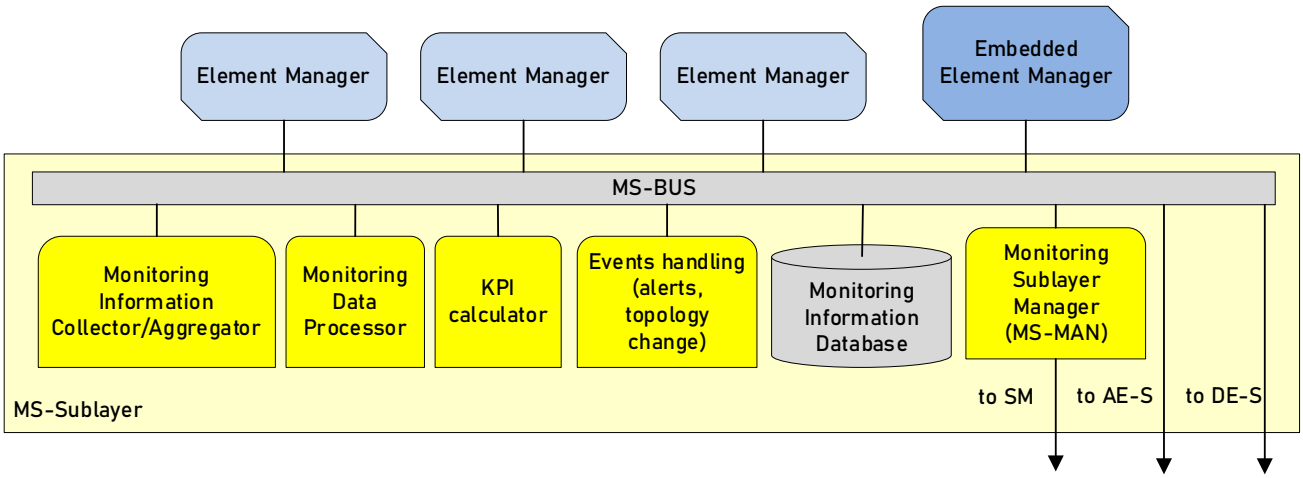

Figure 10. Monitoring System Sublayer internal components.

The output of MS-S is accessible to other components of SML via a dedicated message bus. In this way, we enable the publish/subscribe paradigm of specific monitoring metrics. In general, MS has to interact with EEMs of different technological domains that are VNF specific. However, several MS operations and queries, e.g., those related to computing resource availability and consumption, are generic. Therefore,

many of the internal components of MS can be reused for multiple slice templates. The definition of ad-hoc protocols for efficient communication between the EEMs and MS and the adaptability of the monitoring (adaptive sample rate or resolution, gossiping protocols, etc.) is out of the scope of this deliverable. The MS efficiency can be evaluated. Table 1 consists of two example metrics that can be used for that purpose.

Table 1. Exemplary metrics of the MS Sublayer.

MS metric	Metric calculation	Target values
Overhead for monitoring activities and level of data aggregation	- Overhead data monitoring: number, size of messages Frequency of sending data ≤ Frequency of received data	Frequency of sending data ≤ Frequency of received data
Data ingestion rate	- Data ingestion should be: fast, not complex. - Check if the data has been ingested correctly and validate the result data file.	- High data ingestion speed. - Errors in data integration must be zero or very low towards zero.

More about implementations of MS-S on different hierarchy levels can be found in WP3 deliverables of MonB5G.

3.4.2.2 AE SUBLAYER OF SML

The **AE Sublayer** includes a set of AEs and the corresponding AE Sublayer Manager that is used to interact, configure and deploy the individual AEs, as depicted in Figure 11.

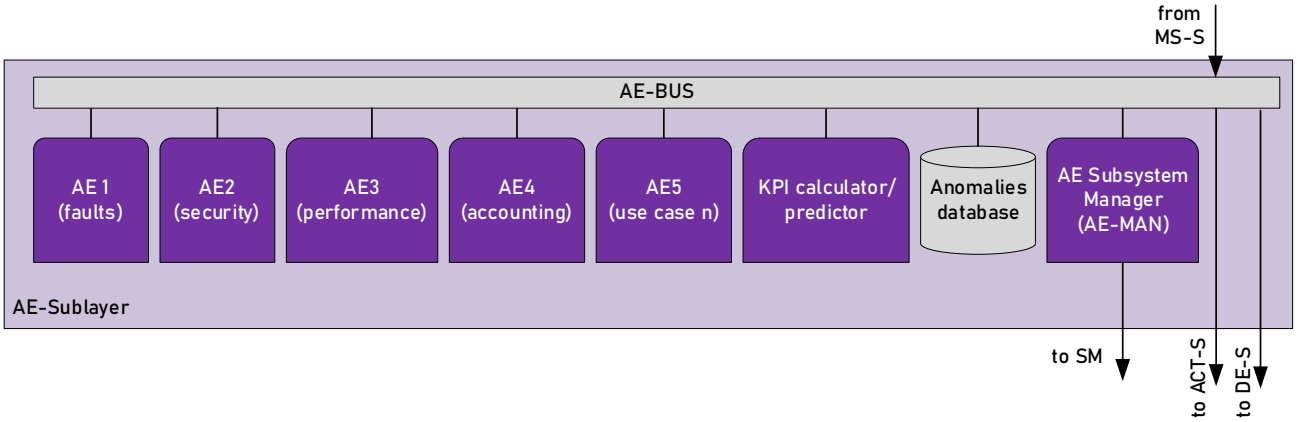

Figure 11. Analytic Engine Sublayer internal components.

Adhering to scalability and flexibility practices, we envision the deployment of multiple distributed AEs, each one performing a singular task (or a limited subset of them) according to the target scenario. For example, AEs may analyse monitoring information generated from slice-specific traffic traces, implement a specific mechanism for security threat detection, and investigate in real-time fault and/or performance degradation. Given the wide set of possible functionalities needed to monitor a multi-domain slice deployment, the internal specification of AE algorithms is use-case dependent and cannot be provided *a priori*; there is, however, the possibility to create a library of AEs algorithms that can be taken as reference solutions for multiple different slice templates or adapted to novel analytic requirements with a relatively small effort. One of the components of this sublayer is Slice KPI calculator – an entity that is responsible for the calculation and prediction of slice specific KPIs;

In spite of the fact that a one-to-one mapping of AE and DE is preferable, to avoid conflicting scenarios, the proposed architecture allows a single DE to seamlessly interact with multiple AEs at the same time. It is worth mentioning that MS-S provides the abstraction of the monitored data that positively contributes to the reusability of AEs and to the reduction of the communication overhead. Table 2 consists of a (non-exhaustive) list of metrics that can be used for the AE evaluation.

Table 2. Exemplary metrics of the AE Sublayer.

AE metric	Metric calculation	Target values
Event processing capacity of AE	Calculate the capacity needed by AE to analyse data received from MS.	The capacity of processing data or event ≤ capacity of AE.
Time series prediction error	Since AE uses ML to analyse data, the evaluation of ML algorithm used for this purpose is based on ML metrics: for example, using times series prediction accuracy.	Target values depend on ML used metric.
Anomaly/faults detection frequency	- The sum of Anomaly/faults detected - Use ML metrics, e.g., AUC, F1, precision, recall	The sum of Anomaly/faults detected ≤ a specific threshold defined according to the considered model. Target values depend on the used-ML metric.
Load profiling and forecasting speed & precision	Speed: high Precision: high	High
Computing resource consumption speed	Resource consumption: CPU, memory, etc.	High speed, set a minimum speed threshold according to the considered model.
Reconstruction error due to lossy Encoding-Decoding processes	A measure of the reconstruction error introduced by data compression models. The metric can be computed as an MSE or RMSE of decoded data with respect to the input data.	As low as possible (around zero).

More details on AE implementation at different hierarchy levels are presented in the WP3 deliverables of MonB5G.

3.4.2.3 DE SUBLAYER OF SML

In order to meet the different characteristics of multiple technological domains and enhance the scalability of our proposed architecture, we envision the overall DE Sublayer as a composition of **multiple DE entities**, as presented in Figure 12.

DEs are the entities that are responsible for the reconfiguration of SFL or SML. Each of them may pursue a local goal (e.g., domain-specific), for example, resource allocation optimisation according to slice KPIs, local fault handling, security decisions, or even energy-aware operations of specific slice settings. In order to perform the most accurate decisions according to the real-time context, the DE Sublayer interacts with the MS-S to obtain up-to-date monitoring metrics, as well as with the AE sublayer to acquire accurate forecasting and/or statistics information from the ongoing slice deployments and the corresponding underlying infrastructure resource utilisation.

The distributed nature of the proposed architecture allows performing different FCAPS functionalities by multiple DEs. However, the co-existence of multiple "selfish" DEs pursuing local optimisation may lead to conflicting scenarios. This demands for an arbitrage entity able to solve such conflicts. To address this problem, we introduce the **DE Selector/Arbiter** component implemented in the same DE sublayer. This component can be both model-based or AI-driven. In fact, our software-based deployment allows instantiating several DEs that implement different algorithms for the same goal. In such a case, the Selector/Arbiter may define a ranking of DEs ranking to discard conflicting outcomes.

As the stability of the feedback-loop-based management can introduce chaotic behaviour in the system, e.g., the ping-pong effect, a special entity dubbed as **DE Observer** is introduced in the DE Sublayer. Its database stores recent reconfiguration decisions together with the corresponding input values that triggered the reconfiguration decision and uses this historical information to restore the settings to a stable configuration after the detection of an issue. In general, a DE can be deployed to make decisions based on slices, virtual resources or physical resources (or anything that it wants to manage within the 5G architecture). As the number of entities to be managed by DE grows (as its scope increases,), then the latencies associated with response time (computational cycle length), the time it takes for the decision of

DE to reach actuators and the time it takes for the system to respond to these changes are expected to also increase. At some point, the compounding of all these times (latencies) will make the system inoperable, thus it will be necessary to make a smaller DE, or to reduce its scope (to reduce the number of entities it can control), in order for it to manage. This will mean that DE will reduce in complexity and/or will manage fewer entities. The DE Observer will be involved in such a process.

The DE sublayer, as other sublayers of SML, have a **DE Sublayer Manager** that can be used for the change of the configuration of its components or their policies.

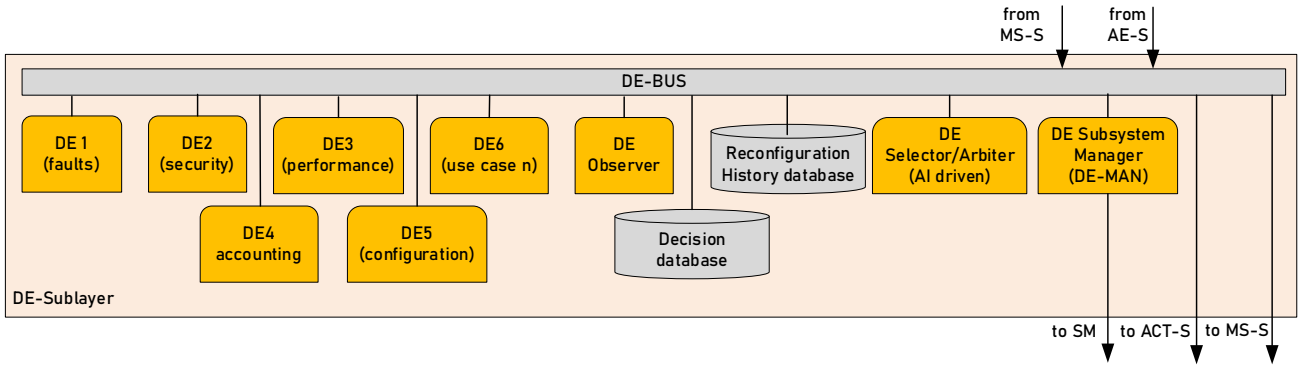

Figure 12. Decision Engine Sublayer Internal components.

The DE decision can be used for the reconfiguration of SFL or SML. In both cases, there are three possible reconfiguration operation types:

- reconfiguration of functions/nodes of SFL/SML;
- change of resource allocation to SFL/SML components (including transport). Dependent on the implementation, it can be done directly or by the interaction with DMO;
- modification of SFL/SML by upgrading the slice template (i.e., adding or removing VNFs). In this case, SML will interact with DMO requesting deployment or removal of a specific function or a node.

For the DE performance evaluation, the metrics proposed in Table 3 are applicable. The metrics are calculated for each DE by the DE Observer entity.

It is noteworthy that the modification of resource allocation in SFL or SML by means of SML can be triggered proactively, instead of the reactive allocation that is provided by the ETSI MANO orchestrator. Moreover, in the case of SFL, resource allocation can be driven by slice QoE, not by slice QoS. Respective calculations can be done by SML. The modification of the SFL template that is driven by SML may be used for cloning some slice functions in order to optimize slice traffic or to add additional components like Deep Packet Inspection entities or firewalls. The same mechanism can be applied in SML, where new components like AEs or DEs can be dynamically deployed (even at runtime) in response to triggering scenarios, therefore enabling full programmability of the slice management plane. This feature is important for the evaluation of the quality of different AE and DE algorithms, as well as for verification of their correct operation.

Table 3. The exemplary DE metrics calculated DE Sublayer components.

DE-related metrics	Metric calculation	Target values
Response time of DE: time consumed from sampling its inputs until generating a decision	Response time of DE: low or ≤ Time threshold. It can be estimated using Littles law	Response time of DE ≤ time threshold
Scalability: the number of entities it can simultaneously control. Important to quantify due to the granularity of the resources an instantiation that DE will handle	Evaluate scalability include evaluate the following metrics: Response time, Requests per seconds, Network usage, Memory usage and the time it takes to execute tasks	These target values have to be defined according to the concept and model considered based on the requirements.

DE-related metrics	Metric calculation	Target values
Power/cost management efficiency: Quantifies the cost of deployment depending on memory footprint, computation load, and actual energy cost.	Cost/power: low or ≤ Threshold Apply the solution that has a lower cost/power.	Cost/power ≤ Threshold to be set according to the model considered and desired power/cost management
Decision-To-System State Compliance: the error of how close is the state desired by DE with respect to the state achieved by the system DE is controlling.	The time it takes the system to reach the desired state and how exactly it can reach it might cause DE to issue multiple decisions to reach that desired state until the previous decision is fully enforced. This can be quantified by sampling the current state and the previous output from DEs. It is also necessary to consider convergence time separately.	Decision errors ≤ Threshold to be set according to the model considered and desired accuracy
Domain-specific decision impact on other domains: In the case of multi-agent scenarios (overall and per-slice metric).	Accuracy of multi-agent ML algorithm	Maximize accuracy by carefully adjusting the parameters of the multi-agent algorithms.
VNF scaling, placement, and grouping policies frequency.	VNF Performance: Network workload vs. throughput VNF horizontal scaling: instance capacity VNF vertical scaling: vCPUs/memory resources	The target values have to be defined according to the concept and requirements.
Decision quality: Evaluation of the decisions through a measurement of the gap between an approximative decision made by DE and the baseline DE algorithm.	When a baseline algorithm (BASE) exists, a performance of a new algorithm (DE) with regard to some metric (e.g., latency, cost, resource consumption, execution time, etc.) can be done and expressed as a scalar.	Very low
Memory footprint: memory consumed by DE. Metric has three components: memory it needs while running, memory consumed while training, and the training frequency.	Memory footprint can be measured using "top" directly and see how much memory it consumes when different stages are executed. It is necessary to run multiple instances to measure the time it takes, extract minimum and maximum values of memory consumption, generate a mean, a variance, and a histogram.	
Computation-cycle length: time consumed from sampling its inputs until generating a decision.	The time that it takes for DE to generate output once it samples a state instance is dependent on many factors, among which we can mention (but there can be more): the size of the state space, the values that it has and the platform (the testbed). The measurement should collect statistics (min and max times, the mean, and the standard deviation).	Set a time threshold of the computation cycle, according to the considered model and factors.

More details on DEs on different hierarchy levels is provided in MonB5G WP4 deliverables.

3.4.2.4 ACT SUBLAYER OF SML

The role of the ACT Sublayer is to convert high-level (intent) reconfiguration commands obtained from the DE Sublayer into a set of atomic reconfiguration commands, as shown in Figure 13.

The ACT Sublayer abstracts the domain-specific details to DEs and enables intent-based control. Therefore, they can be designed in a more generic way. The ACT Sublayer can be seen as a set of technology-specific (i.e., node/functions) drivers. ACT typically interacts with EMs/EEMs of SFL, but they may also interact with DMO requesting orchestration-related action (adding or removing a VNF).

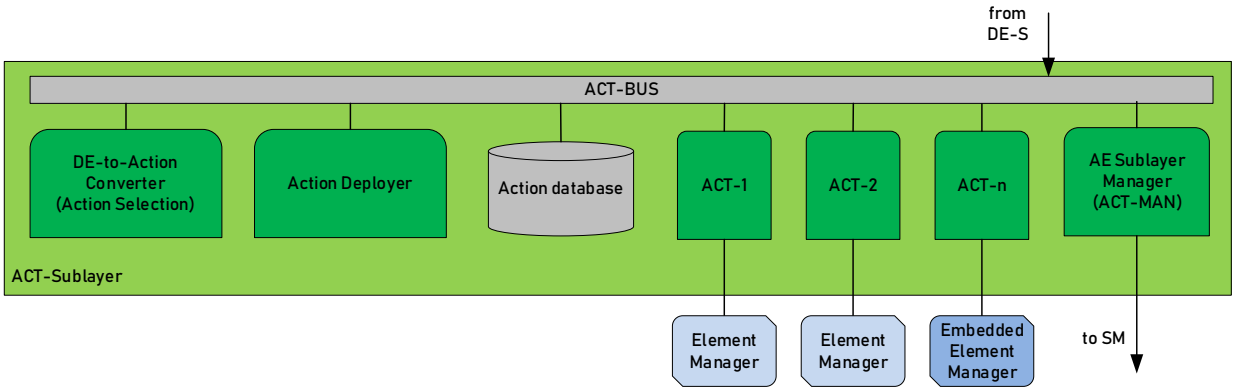

Figure 13. Actuator Sublayer Internal components.

3.4.2.5 SLICE MANAGER

Slice Manager (SM), presented in Figure 14, is an entity of SML that provides interactions with DMO and Inter-Domain Slice Manager (IDSM), described in this section. It can also be used for the manual management of SFL or to implement Policy-Based Management (PBM). It interacts with EEMs, MS, AEs and DEs. The component is responsible for sending to DMO and, if applicable, to IDSM, slice-related synthetic information (KPIs). SML provides slices management plane isolation due to direct, intent-based management to Slice Tenant. For that purpose, SM has a tenant portal that gives access to a set of management tools to enable simple and direct access to the slice tenant at the slice reconfiguration options. A *conditio sine qua non* for such management is the embedded intelligence of the management that is in our case provided by the AI algorithms.

The management interface is created after slice deployment, and the slice tenant can use it for the entire lifetime of a slice. For accounting and historical reasons, the tenant details combined with slice resource consumption data and KPIs are transferred to IDMO Accounting database before termination of the slice.

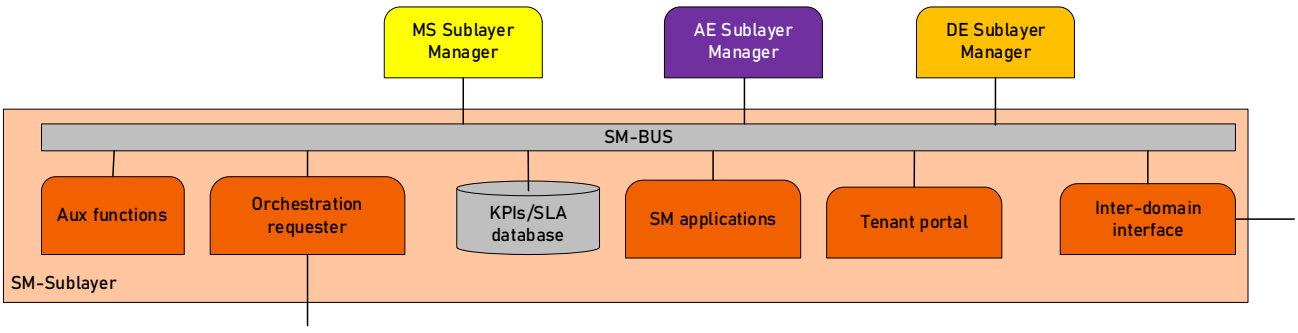

Figure 14. Slice Manager internal components.

3.4.3 INTER-DOMAIN SLICE MANAGER

The proposed SML-based slice management approach can also be used for E2E slice management when slices are implemented across multiple SODs, as shown in Figure 15. In such a case, the entity called Inter-Domain Slice Manager (IDSM) is responsible for the E2E slice management. It interacts with SMLs of all domain slices that compose the E2E slice. IDSM is a part of a slice template (a set of VNFs), and, in some

cases, it can be generated automatically by IDMO (if IDMO is responsible for slice template split between multiple SODs). When IDSM is in use, it provides to the Slice Tenant the management interface. IDSM is also responsible for the calculation of slice-related KPIs. Figure 15 shows a multi-domain slice with IDSM deployed in one of SODs.

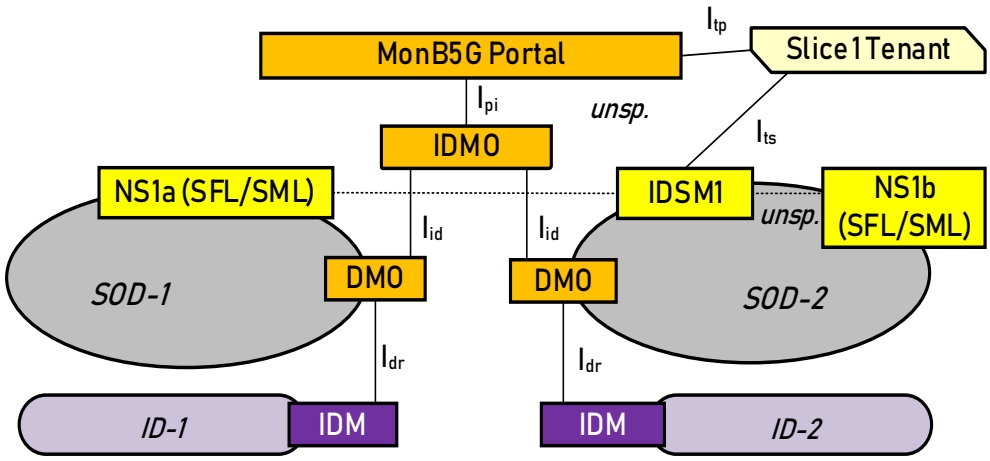

Figure 15. Multi-domain slice – IDSM is deployed in one of SODs.

The Inter-Domain Slice Manager (IDSM) can be seen as an instance of SML. The generic structure of IDSM and its usage has been presented in Figure 16.

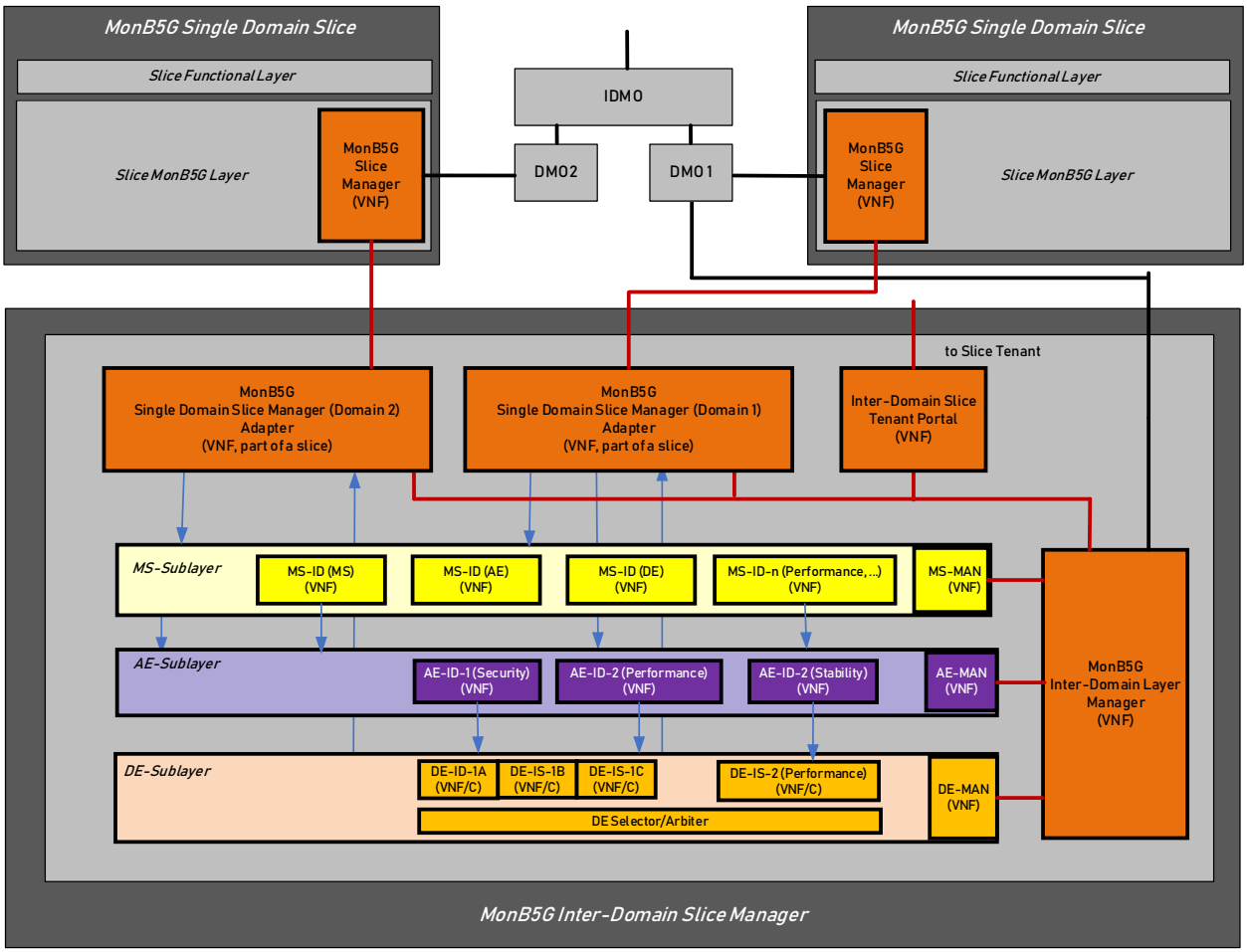

Figure 16. Example of usage of IDSM.

IDSM implements the runtime part of CSMF and NSMF of 3GPP (see [8] for details on NSMF and CSMF).

3.4.4 MONB5G LAYER AS A SERVICE

The addition of SML to SFL undoubtedly increases the slice footprint, implying also longer slice deployment times. Moreover, in some cases, the Slice Tenant is not interested in slice management. To solve the mentioned issues, MonB5G proposes the use of the Management as a Service (MaaS/PaaS) paradigm, as described in [58]. In this case, SML is an independent slice capable of managing multiple SFL instances of the same template, as SML cannot be generic. Such split requires the implementation of additional functions in SML related to the creation of secure partitions for the managed SFLs, as well as dynamic adaptation in case of deployment of a new SFL, or termination of the existing one. The MaaS platform (called MonB5G Layer as a Service – MLaaS) can be operated by a business entity called Slice Management Provider. LCM of MLaaS is done via I_{mp} interface. The case for a single SOD is marked as Option B in Figure 7, where SFLs of the MLaaS-managed slices also use services provided by a DSF for reduction of SFL footprint.

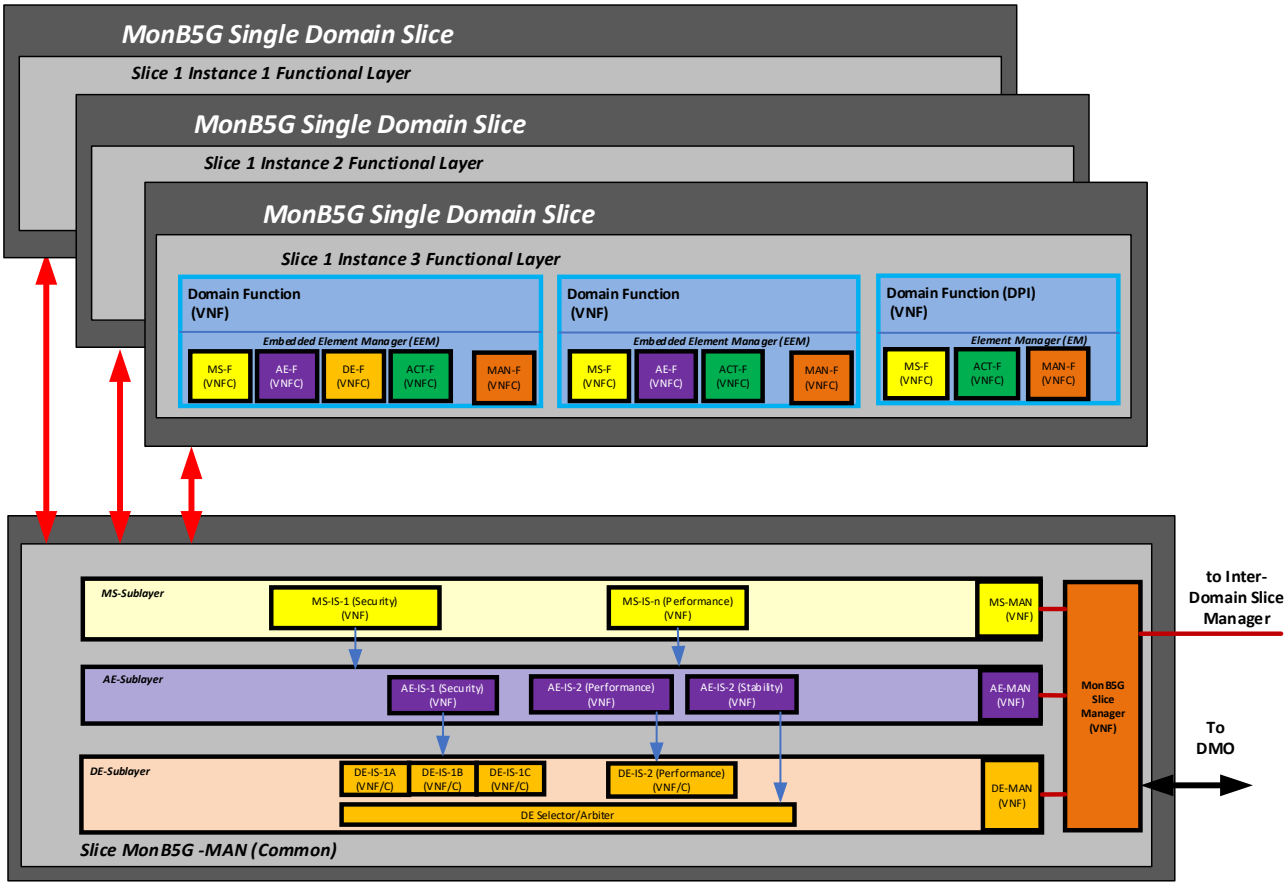

Figure 17. Example of usage of MLaaS.

To implement this approach, MonB5G architecture will leverage the concepts proposed in [58] to provide PaaS capabilities to management services. These PaaS may also be requested by consumers via formal APIs, which will define the type of PaaS required (e.g., there exist many Container Infrastructure Services, such as Kubernetes[13] and OpenStack Zun[14]), and other design considerations. MonB5G components (i.e., MS, AE, DE, and actuation components) are deployed inside PaaS alongside each managed function (e.g., components within a slice orchestration domain for a particular technological domain). LCMs of MonB5G components are performed by the infrastructure owners. On the contrary, PaaS requested by consumers is managed as a single entity by the operator (e.g., as a tenant slice), leaving LCM tasks associated with the services residing therein to the consumer.

[13] For further information also see: https://kubernetes.io/.

[14] For further information also see: https://docs.openstack.org/zun/latest/.

The MLaaS model allows the reuse of infrastructure resources employed in the management of network slices that share a common descriptor. That is, if two or more network slice instances from the same descriptor are deployed, the MonB5G management system may deploy an SML in the form of a shared MLaaS (i.e., a shared Network Slice instance or VNF Common Service) and configure its internal management components to trigger the target SFLs. This approach favours the scalability of the management system as the additional load (e.g., resource allocation, OPEX, etc.) will not grow linearly with the number of managed network slice instances, i.e., a single set of resources will support PaaS, while only configuring additional MonB5G components targeting each SFL. A MLaaS usage example is depicted in Figure 17.

3.4.5 *DOMAIN SHARED FUNCTIONS*

The Domain Shared Functions (DSFs) are a set of shared functions implemented as PNF/VNF or CNF and they can be reused by a single or multiple SFLs of the same or different template. The approach provides a reduced footprint of the deployed slice, as enjoyed by VNF Common Services specified in [58], or may be dedicated to a specific SFL – as a complementary part of it – in the manner of VNF Dedicated Services. Its operation can be seen as vertical stitching of slices, in which DSF can be seen as a PaaS servicing the slice's SFL.

In the PaaS case (VNF Dedicated Services), both vertically stitched slices are overlooked by a specific SML providing customizable management options to achieve any DSF optimization. Similarly, DSF as VNF Common Service (DSF-C) and their subscribed slices (DSF-C Set) are managed by a dedicated IDSM, which considers the particularities of shared functionality, e.g., ensuring DSF-C availability across domains.

DSF-C may be used in combination with MLaaS for further reducing a slice resources footprint. In such a case a tenant slice may only contain a common SFL part (as to make it suitable for MLaaS management), while the operator provides extra functionality via DSF-C, including a shared service. Management functions associated with each DSF-C Set (e.g., specific MS/AE/DE components) are then deployed at MLaaS of corresponding technological domains.

3.4.6 *INFRASTRUCTURE ORCHESTRATED MANAGEMENT FUNCTIONS*

The **Infrastructure Orchestrated Management Functions** (IOMF) are specific functions that support infrastructure management. They can be orchestrated by IDMO upon request of an Infrastructure Provider via the MonB5G Portal as described in section 3.3.1 and presented in Figure 18.

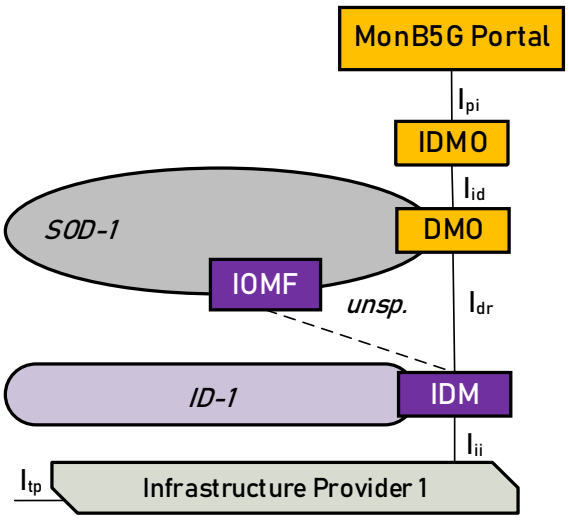

Figure 18. Deployment of IOMF function.

IOMF cooperate with IDM to achieve their specific goals. The IOMF functions can cover a variety of functions that can improve the effectiveness of infrastructure utilisation and contribute to the overall quality of infrastructure management. The most prospective use cases include the deployment of IOMF for predictions of resource consumption and resource groupings to optimise utilisation and, as a result,

achieve energy-saving goals. The IOMF functions are orchestrated in a similar way to slices, and LCM requests are sent by the Infrastructure Provider to the MonB5G Portal. The IOMF functions are specific for the virtualisation technology used in the infrastructure and tools; therefore, they must be customised for each IDM type separately.

MonB5G considers two types of IOMF, namely, **On-Prem** and **Cloud** IOMF. The former refers to IOMF that interacts with an IDM which itself manages on-premises or metal resources. An On-Prem IDM (e.g., Metal as a Service[15], TinkerBell[16]) then exposes a variety of telemetry and actuation Northbound Interfaces (NBI) in the form of APIs. Cloud IOMF, on the other hand, interacts with Cloud IDM. Such IDM (e.g., Terraform[17], Google Cloud Platform[18]) manages infrastructure deployed on external cloud providers, which offer billing as well as resource consumption and reconfiguration HTTP APIs NBI for a determined ID.

Regardless of their type, IOMF are deployed as part of a Slice Management Layer (SML), which allows reconfigurations (e.g., updating optimization goals), but relevantly, subscribe to new IDM to widen their management domain (e.g., adding a new Cloud IDM for achieving energy efficiency goals).

3.5 Security components of the MonB5G architecture

The security orchestration in MonB5G is based on the methodology recommended by well-known security frameworks to implement and reinforce network cybersecurity. The National Institute of Standards and Technology (NIST) cybersecurity framework [59], for example, is divided into five main functions: identify, protect, detect, respond, and recover, which comprise the security lifecycle. From the network model designed by the manager (NSMF or NSSMF), the security orchestrator makes an inventory of assets, captures the security needs of the intent object and then analyses the vulnerabilities, threats and risks to determine the set of security objectives that need to be achieved. It uses AE to understand the structure of the network to help DE to manage security controls. Then the protective measures can be identified and deployed together with the network instance to protect it against cyber-attacks. Identity and access control, the security of data and networks are examples of measures. However, security measures may have breaches as threats are constantly evolving; it is necessary to monitor services and resources in order to detect new security problems by anomaly or artefact. The events emitted by the detection system are then handled by the response system, which will determine, based on its model and rules, the actions to be taken, for example, hypothesising a known attack and then verifying it based on the observed data or activating existing mechanisms or deploying additional ones to limit the effects of the incident and eradicate it. This should be implemented by closed-loop automation involving MS, AE, and DE components. The security framework also recommends a post-incident analysis to learn from the incidents handled. The speed of remediation, the inadequacies of the data collected, and the overall cost of an incident are all pieces of information that help to identify new vulnerabilities and risks, strengthen defensive measures, broaden the scope of analysis and improve response plans.

MonB5G security framework concerns multiple levels of the management and orchestration hierarchy. The global security is managed using the E2E Autonomic Security Orchestrator located in IDMO, besides the security-related management loops (cf. Figure 19). However, at the domain level, the security management is hosted in DMO, where we have a Domain Autonomic Security Orchestrator managing the security closed loops and the security enablers (Virtual Security Functions). In case of multi-domain network slices, NSMF (located in IDMO), and in case of single-domain slices, NSSMF (located in DMO), delegates part of the management of security goals or intents to the Local Autonomous Security Orchestrator in slice management level, i.e., to SML.

Security control functions and the core functions of the cybersecurity framework are provided by a Security Service Platform (SECaaS). As a set of VNFs, SECaaS can be dedicated to the safeguard of a network slice, or it can be shared. The application of the security framework at this layer requires the

---

15 See: https://maas.io/.
16 See: https://tinkerbell.org/.
17 See: https://www.terraform.io/.
18 See: https://cloud.google.com/apis/docs/cloud-client-libraries.

presence of a security platform in Local Autonomic Security Orchestrator (L-ASO), Domain Autonomic Security Orchestrator (D-ASO) and E2E-ASO. Indeed, SECaaS of IDMO manages the security lifecycle globally from the framework core function and identifies the necessary framework core function respond/recover operations. SECaaS of D-ASO can monitor multiple slice instances and amplify the signal to detect a threat that is transparent to every individual network slice. SECaaS of L-ASO provides security management functionalities within a given slice. Another valuable point is the sharing of information between the network slice to reinforce the protection. In addition, a Security Platform as a Service (SecPaaS, and implementation of PaaS) can be dedicated to a group of network slices according to some criteria: per tenant, vertical, slice type, security level. That isolation between SECaaS with the security orchestrator allows the possibility of employing multiple strategies and policies to manage the security lifecycle.

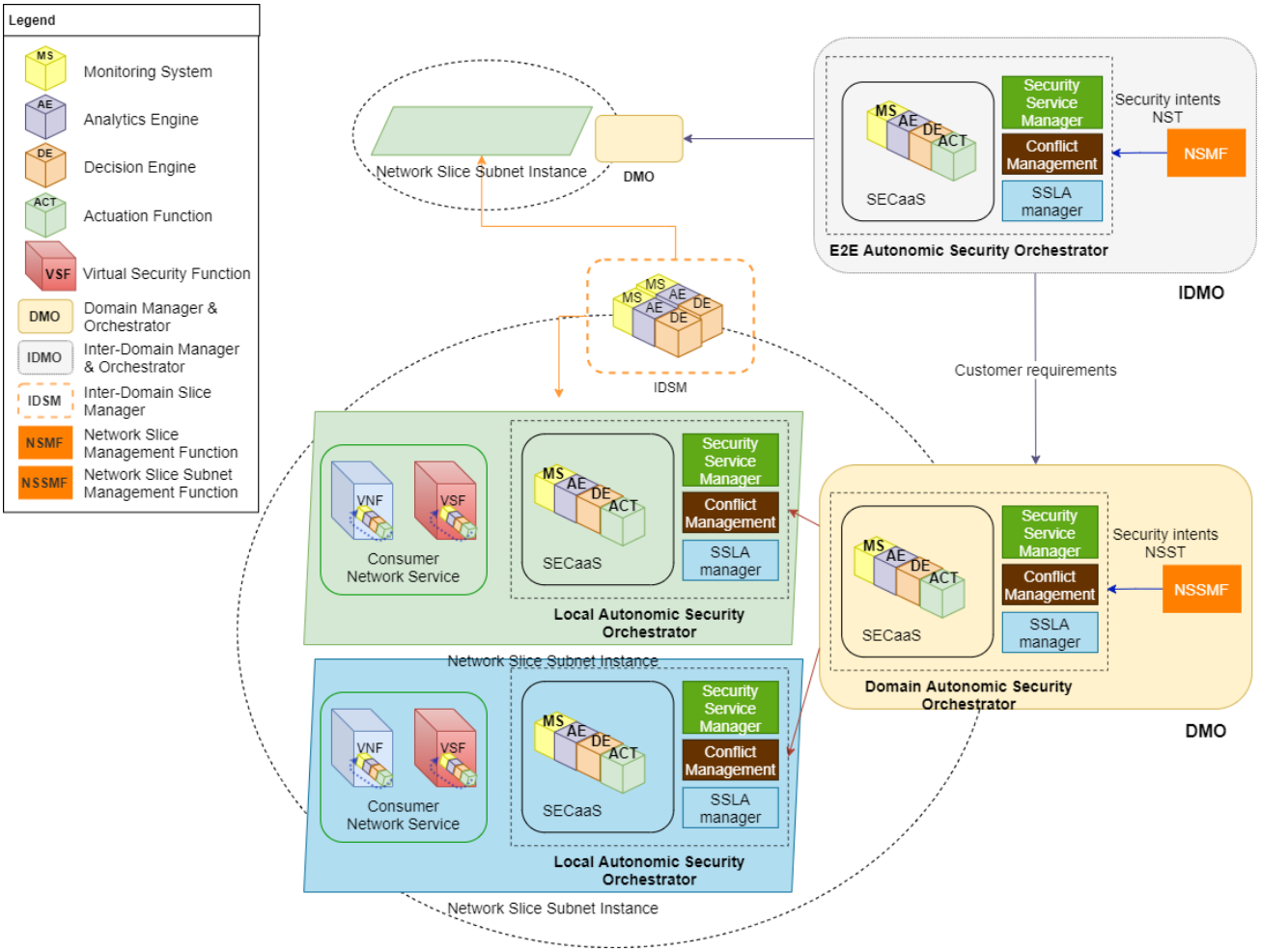

Figure 19. Security components of the architecture.

At the inter-domain level, that of network slice, which aggregates several slice subnets to offer E2E services, a SECaaS is also in place to manage the security lifecycle of network slice instances. From the customer security intent, SECaaS of NSMF can extract and distribute goals to each NSSMF. As for the detection, it can also collect security reports and detect a threat thanks to its view over the network segments. As a response, for instance, a change of security requirement is sent to NSMF of a network slice subnet instance.

3.5.1 SECURITY ORCHESTRATION

Security is distributed with the slice instances by implementing the required security management closed loops following SECaaS paradigm. The closed loops and the security enablers used to enforce the security policies are managed through the "Security Service Manager" component of the security orchestrators (SOs) deployed at E2E, domain, and slice levels, namely the E2E Autonomic Security Orchestrator (E2E-

ASO), Domain Autonomic Security Orchestrator (D-ASO), and Local Autonomic Security Orchestrator (L-ASO). In what follows, we will elaborate on the security orchestration capabilities of the three types of SOs.

3.5.1.1 E2E AUTONOMIC SECURITY ORCHESTRATOR

E2E-ASO (part of IDMO) has a global view and is responsible of the security of all slices from the E2E perspective. At the slice creation phase, E2E-ASO checks (via communicating with domain SOs) whether the slice can be deployed at the agreed Security SLA (SSLA) requirements depending on the security capabilities provided by the respective technological domains where the slice will be deployed. If feasible, the E2E-ASO identifies the security policy to enforce and the corresponding enablers to deploy based on the slice blueprint, the received SSLA and the performed risk assessment. During the slice runtime, E2E-ASO is responsible of enforcing cross-domain security decision policies coming from IDMO closed loops, such as migrating a sub-slice to a new domain to avoid a security threat in the original domain [60].

D-ASO manages those IDMO closed loops and might coordinate and assist the local security orchestrator with selecting SECaaS to be deployed with each domain slice (sub-slice). E2E-ASO stores the policies enforced from his level into the conflict manager that permits to check any conflicts.

3.5.1.2 DOMAIN AUTONOMIC SECURITY ORCHESTRATOR

Local to every domain, D-ASO (part of DMO) ensures the local security management, by instantiating and managing the appropriate closed loops using a SECaaS approach when deploying a sub-slice instance. The adoption of SECaaS model allows the reusability of the deployed closed loops or some of their components (i.e., MS, AE, DE and ACT) between sub-slices. D-ASO ensures that the reusability is performed according to the isolation level of slices. The security policies issued by DEs and their enforcement status are stored and managed by the “Conflict Management” component, providing D-ASO an overall view of all security policies enforced within the domain’s sub-slices and allowing to avoid any conflict between policies. Once issued, the security policy is saved with “Enforcing” status. Its status is changed to “Success” if the ACT component can correctly execute the policy’s actions, otherwise it is set to “Failure”. D-ASO ensures the successful enforcement by adjusting the security policy and executing the necessary corrective operations. If the mitigation of the security issue is not possible at the domain level, D-ASO escalates the problem to E2E-ASO, which assesses the problem and generates the mitigation actions in E2E scope [60].

3.5.1.3 LOCAL AUTONOMIC SECURITY ORCHESTRATOR

L-ASO located on the slice level which is responsible for managing the internal security of a sub-slice. L-ASO is in charge of communicating with the sub-slice VNFs to monitor, analyse local data and take mitigation actions directly within the sub-slice scope. In case the mitigation is not possible, L-ASO informs D-ASO in order to take the adequate actions to address the security issue.

Similarly to E2E-SO, D-ASO has a Security SLA (SSLA) manager that interprets the SSLA requirements into security actions and KPIs while deploying a slice to monitor and ensure that the tenant security requirements are met. The Security Service Manager is responsible for instantiating and deploying SECaaS. In addition, to the Conflict Manager that stores the domain enforced policies and prevents potential conflicts.

3.5.1.4 SECURITY AS A SERVICE COMPONENTS

Our architecture is strongly based on the triplet of components consisting of the Monitoring System (MS), Analytic Engine (AE), and Decision Engine (DE). At certain levels, a component called actuator (ACT) is added (see details in section 3.4.2.4). These components provide security as a service (SECaaS) and can be shared between multiple slices. As we expect to have different security and management components, we use the layering concept for reusability and simple communication. To clarify, all MSes are connected in a layer (Monitoring System Sublayer), and similarly, we have an Analytic Engines Sublayer and Decision Engines Sublayer. A vertical channel connects the three sublayers; for example, an AE can request

additional data from other MSes. Also, this channel opens an interface between the other hierarchical management layers, so that any management element can have access to any services.

- The Security AE subscribes to MS the real-time security-relevant data generated by MS The data sources are identified depending on the hierarchical level and the final objective of DE. MS pre-processes the collected data.
- The Security AE processes the security-related monitoring data in order to provide high-level security information and events. Analysing the collected KPIs, network flows and resources status will help in diagnosing the node and the network to detect or predict attacks and security issues.
- The Security DE is the mastermind that has the ability to tell the system what to do as a reaction or prevention to protect the network against security threats. The Security DEs' role is crucial in detecting attacks and deciding on dynamic security policy per slice and per attack episode. The decision can configure an existing security enabler in the slice or deploy a new one; however, these decisions are described in an abstract model rather than vendor-specific. Among DEs distributed horizontally and vertically, each can be related to an application/VNF, domain slice, or an E2E slice scope. DEs are atomic elements that implement a specific autonomic security function. For instance, at the VNF level, the embedded DE is responsible only for a unique security threat that may target its hosting VNF. In this case, the security vulnerability or the potential attack should be known earlier (before deploying DE) based on the application running or the protocols which may differ from a VNF to another. At the higher level, the slice-attached DEs depends on the slice threats and characteristics. Similarly, for slice DEs deployed in IDSM. This distribution will greatly simplify the autonomic threat detection and fast, local response.
- The Actuation component (ACT) is in charge of executing security policies. First, it performs a translation from the high level into vendor-specific configuration according to the targeted enablers. ACT can trigger deploying a specific Virtual Security Function (VSF) through the NFV MANO or update the configuration on an existing VNF/VSF.

3.5.2 SECURITY ORCHESTRATOR INTERFACES

The following reference points are defined for SO and the security platform SECaaS at the inter-domain, domain and functional layers (cf. Figure 20):

- **So-Os**: The reference point between SO and OSS is used by OSS to delegate the management of the security goals of the network slice to SO. This is IDMO/DMO internal interface.
- **Sc-MB5G**: The reference point between the security services and the MonB5G sublayer components is used to interact with MonB5G components to create closed loops for the security process.
- **Sc-Sc**: The reference point is used for the control communications between two security services. The E2E security service has a global view over the activities of slice subnet security platforms and manages the E2E security requirements. In turn, the slice subnet security service controls the functional security platforms to ensure that each slice subnet instance is well protected.
- **Sc-Or**: The reference point between the security platform and NFVO. It is used to monitor the health of NFV objects and perform management operations on them. It is related to the reference point Sc-Or as defined in ETSI GS NFV-IFA 033 [61].
- **Sc-Vnf**: The reference point between the security platform and the consumer NFV object. The functional security platform offers VSFs such as firewall, Intrusion Detection System (IDS), access management as protection measures to improve the security of the slice subnet instance. This is a slice internal interface.

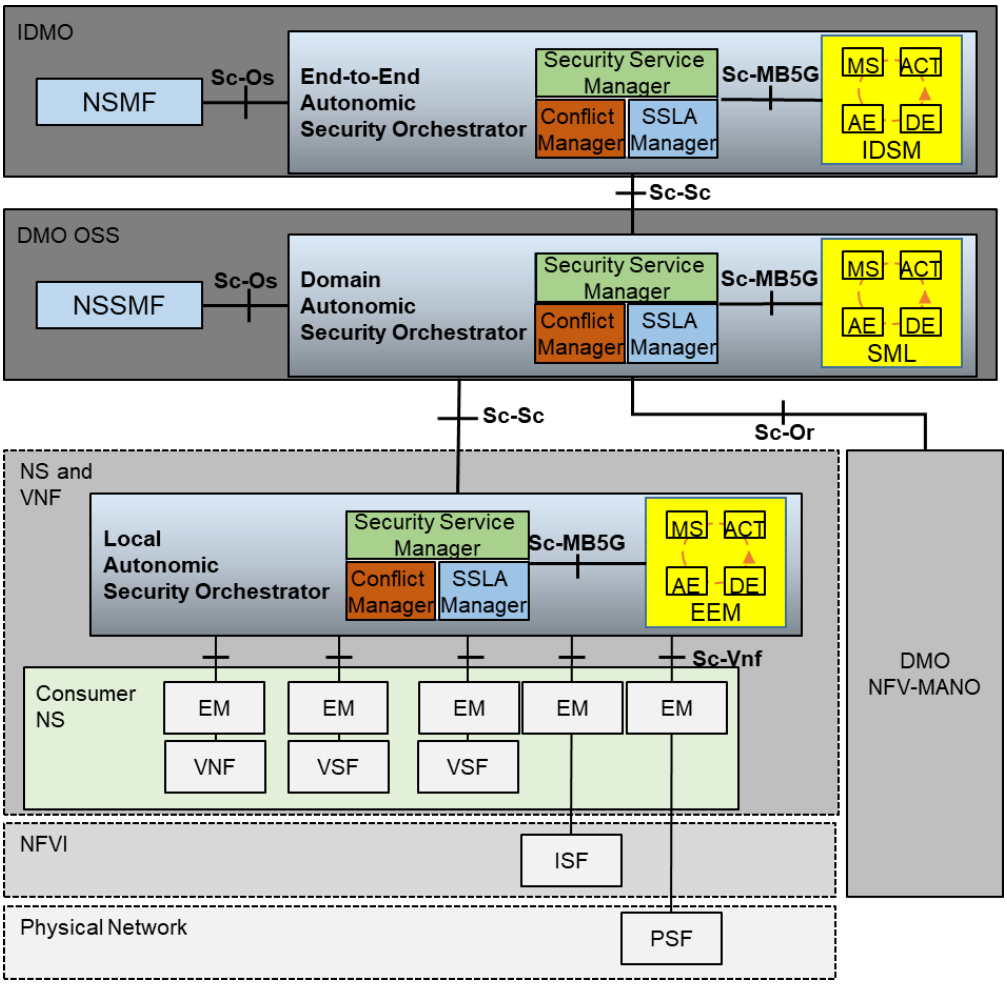

Figure 20. Reference points of the security orchestrator.

3.5.3 POLICY MANAGEMENT

To outline how the MonB5G security architecture approaches the management of security policies, it is useful to map the components above to the roles defined by Zero Trust Architecture [62] and similar frameworks as illustrated in Figure 21:

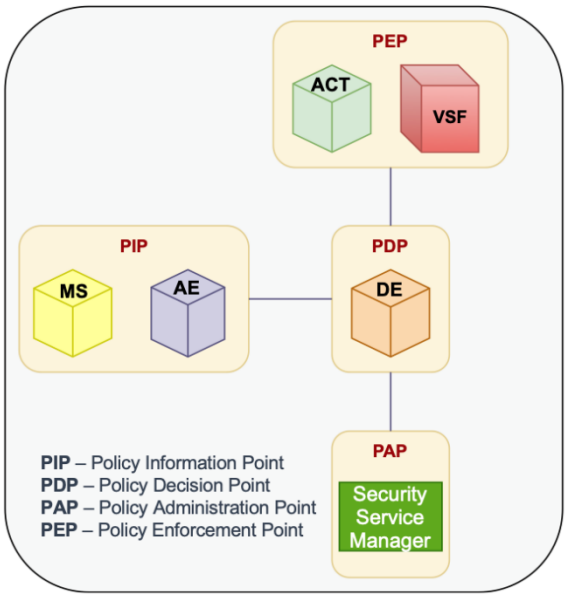

Figure 21. Zero-Trust model of security policy management in MonB5G security architecture.

- MS and AE collectively map to the Policy Information Point (PIP). MS collects security-relevant events from various data sources, crucial for implementing threat and attack detection techniques in AE, and for maintaining the security context of all actors or entities involved.
- DE, as well as the Security Service Manager of SO are a distributed and hierarchical implementation of a Policy Decision Point (PDP). DE performs policy evaluation in a decentralised fashion, based on events and context from AE and localised policy configurations. However, the Security Service Manager orchestrates policy decisions within and across slice and domain boundaries to implement the security intents configured by NSMF/NSSMF, which thus maps to the Policy Administration Point (PAP).
- ACT, typically in conjunction with VSFs that run continuously, or are instantiated on demand, maps to the Policy Enforcement Point (PEP). ACT translates policy decisions to specific actions that establish a perimeter of network security defences, which continuously protect the services that reside in the slice, or execute remediation, which addresses or mitigates attacks to security.

Based on evaluating technology choices so far, we are looking at adopting the ONAP's APEX PDP[19] Engine to implement the Policy Engine of the MonB5G's Security Service Manager (see section 7.1). The engine is able to handle adaptive policies, i.e., policies that can modify their behaviour based on the current system and network conditions, allowing to support automated decision making.

3.5.4 TRUST MANAGEMENT IN MONB5G

The effectiveness and efficiency of SECaaS closed loops, deployed at domain or slice level, is dependent on the accuracy of the included ML models, which in its turn heavily relies on the availability of a large amount of high-quality training data. Such data may not be available in one domain (or for a given slice) and collaboration between domains/slices is essential to improve accuracy and enable cross-domain/cross-slice security self-managing operations. Nevertheless, the exchange of raw data among domains/slices may not be possible due to strict security and privacy regulations established to prevent leakage of sensitive information. To handle these restrictions while fostering fully distributed collaboration between domains, we adopt a decentralised Federated Learning (FL) approach. This results in increased training efficiency and allows the sharing of learning knowledge without compromising data security and privacy. Furthermore, the adoption of a fully decentralised FL enables greater communication efficiency compared to the vanilla FL[20], as a central entity is not required to exchange the model updates.

Despite its benefits in preserving data privacy, FL is proven vulnerable to adversarial attacks [63]. In fact, malicious participants may perform poisoning attacks by uploading false or low-quality local model updates to undermine the accuracy of the global model. To overcome this challenge, we employ blockchain to enable secure and trustable decentralised FL. The blockchain ensures the integrity of the local model updates to prevent their manipulation. Furthermore, the quality of the local model updates is assessed using a smart contract, allowing to detect poisoning attacks by considering only local models with high performance in the update of the global model.

Figure 22 illustrates the proposed solution for enabling trustworthy collaborative training of an ML-based security model at the domain level in MonB5G's security orchestration framework. Let us consider a training task that is carried out by a set of N domains $\{D_1, D_2, \dots, D_N\}$. Each domain D_i uses its own dataset S_i to train its local model. The local model updates ω_i of the domain D_i are uploaded to the blockchain for integrity assurance and a notification is broadcasted to inform the involved domains about the availability of new updates. Blockchain is also leveraged to thwart poisoning attacks against FL models. To this end, we use smart contracts to evaluate the quality of model updates and automatically identify

[19] See: https://docs.onap.org/projects/onap-policy-parent/en/latest/apex/apex.html.
[20] For further reading also see: https://arxiv.org/abs/2102.12920.

malicious participants in the learning process. Upon receiving a model update notification from D_j $, a domain D_i proceeds as follows:

- If D_i is training its local model, it broadcasts a *pause* message to inform the other domains that an update is ongoing in order to wait for its new model before performing the aggregation. A timer T_{pause} is then triggered at the destination.
- If this update notification is the first since the last aggregation operation, D_i triggers a timer T_{notif} to wait for more updates before starting the aggregation process.
- Once the waiting timers expire, D_i retrieves the model updates that are qualified trustable by the smart contract and computes the global model locally using the Federated Averaging (FedAvg) technique. The trustworthiness of local model updates is determined by evaluating the performance of the local model against a test dataset.
- It is worth mentioning that the trustworthy collaborative training process described above also applies at the slice level.

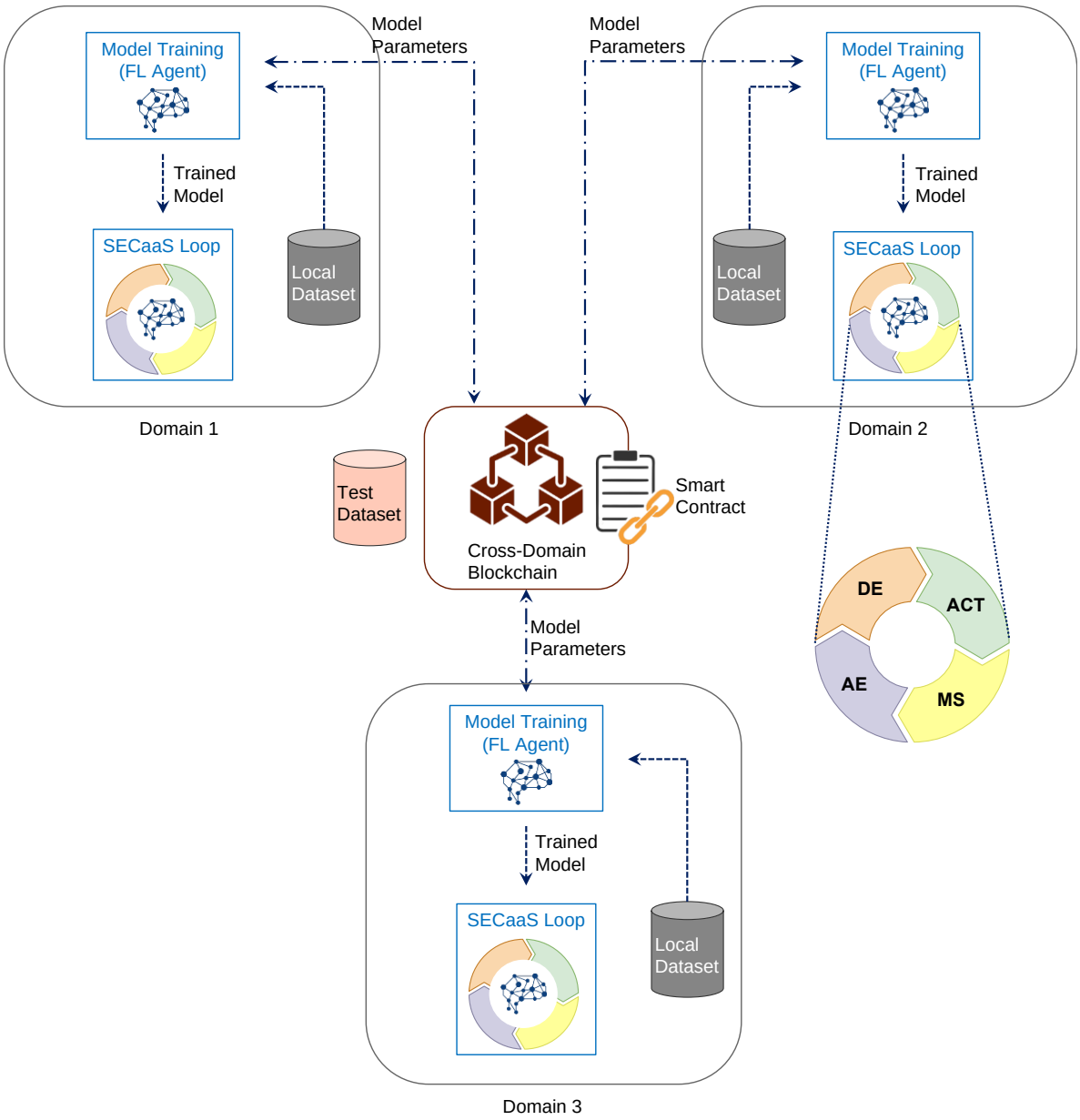

Figure 22. Blockchain-based decentralised federated learning for trustworthy SECaaS.

The proposed solution will be implemented using a permissioned blockchain (e.g., Hyperledger Fabric[21]). In deliverable D2.3 [64], we proposed a new approach to protect the monitoring data from manipulation. The proposed approach introduces a learning pipeline with blockchain-based trustworthy data. As illustrated in Figure 23, the pipeline comprises four components, namely: (i) a Data Collector which collects data from various sources; (ii) a Feature Extractor which extracts features relevant to learning tasks from the raw data; (iii) ML Algorithm and Model which uses the extracted data for training and inference, respectively; and (iv) a Data Integrity Module which maintains and assesses the integrity of data used by the three aforementioned components using blockchain's smart contracts.

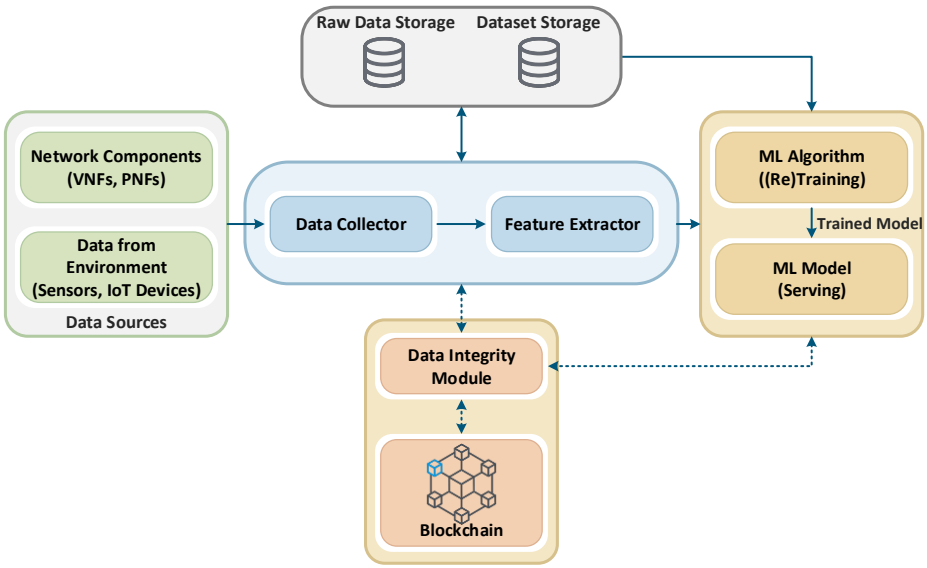

Figure 23. Learning pipeline with blockchain-based trustworthy data.

The data integrity assurance and the audit trails provided by the blockchain comes at the price of the time overhead induced by the blockchain, which may limit the scalability of the proposed solution. Thus, it is of paramount importance to ensure the scalability of the blockchain; that is, the ability of the system to support an increasing load of transactions and nodes. The use of blockchain in the proposed approach may lead to an increased latency in accessing data, which will certainly impact the training/inference time of the ML algorithm/model incorporated in MonB5G's distributed components. Based on these limitations, it is clear that blockchain's main features, including decentralisation, security, and scalability, cannot perfectly coexist. This problem is known by the research community as a "Blockchain Trilemma" [65]. Accordingly, researchers started resolving this problem. We can categorise the existing scalability solutions into two layers:

- Layer 1 (on-chain): Focuses on the blockchain's consensus algorithm such as Proof-of-work, Proof-of-Stake, etc. These algorithms differ on how the consensus can be reached, cost, and confirmation time. Other on-chain solutions target the blockchain's network/data structure. Several techniques took the route of expanding the block size of the blockchain. For example, Bitcoin cash [65] increased the block size up to 32 MB and showed some promise on improving the throughput and thus scalability. Another solution to improve the throughput is based on applying compression to the blocks, specifically redundant data (e.g., Compact block relay [65] and Txilm [66]). Other solutions focus on lowering the stored data and thus reduce storage pressure on the users (e.g., Jidar [67] system for Bitcoin). This approach forces the nodes to keep only the data with the most interest. Therefore, only a small part

[21] Hyperledger Fabric is intended as a foundation for developing applications or solutions with a modular architecture. Hyperledger Fabric allows components, such as consensus and membership services, to be plug-and-play. Its modular and versatile design satisfies a broad range of industry use cases. It offers a unique approach to consensus that enables performance at scale while preserving privacy. For more details see: https://www.hyperledger.org/use/fabric.

of the block data is stored by the node, and if the full data is needed, the node will demand the missing fragments from the other nodes.

- Layer 2 (off-chain): Operates on top of the underlying blockchain protocol. Some transactions are transferred to a temporary off-chain to reduce the overhead on the main chain and improve the throughput. For instance, Bitcoin deploys the Lightning network [68]. This concept uses a temporary channel to make multiple time-sensitive transactions. As a result, only two transactions (tokens) are sent to the main chain. Another solution is based on off-chain computation to enhance the scalability of smart contracts. This solution outsources any complex computation tasks to an off-chain market. As an example, Ethereum uses Truebit [69].

Both layer 1 and layer 2 scaling solutions can be used in MonB5G's trust and security system. This will make the blockchain network faster and more adaptable to a rapidly growing number of users and slices. Moreover, a solution based on combining both on-chain and off-chain approaches can help to achieve maximum scalability without scarifying decentralisation or security.

3.6 Interfaces of the MonB5G framework

The MonB5G architecture has introduced many different types of interfaces. In this section, the most important ones are described. However, the fully detailed description of these interfaces requires a tremendous effort (cf. the series IFA/SOL/TST/SEC documents developed by ETSI NFV group[22], for instance) and goes beyond the scope of the deliverable. In order to enforce the adoption of the MonB5G architecture, it has been proposed to use the existing orchestration and management interfaces as proposed by ETSI MANO or 3GPP as far as possible. The distributed approach of MonB5G makes such adoption in many cases rather limited.

Regarding the MonB5G interfaces, the following observations can be made:

- The MonB5G architecture is composed of static and dynamic components. The interfaces between the dynamic components are left to the programmer's discretion as a part of the slice template and, therefore, they do not need to be defined a priori. There is a need to define only those interfaces that are used for the interactions with the static components of the architecture.
- It is worth to mention, however, that ETSI MANO has not defined many interfaces (EM-OSS, NFVI-OSS, etc.), and in the case of others, the detailed standardisation has been intentionally desisted (e.g., Vi-Vnfm and Or-Vi – no SOL-level specifications exist for these reference points), as they are implementation-specific. The MonB5G architecture is characterized by a similar approach.
- It must also be noted that due to the ISM approach, the number of external slice interfaces has been significantly reduced.
- All the interfaces between the business entities and the MonB5G System as well as the interfaces between Slice Manager/IDSM and Slice Tenant or MLaaS and Management Provider, are assumed as Web-based interfaces.
- The most important and complex interfaces are the following ones: Management Portal – IDMO, IDMO – DMO and DMO – IDM.
- The orchestrators of different levels of the hierarchy, i.e., DMO and IDMO, do not interact directly. For the interaction between them, an OSS integrated with the orchestrator is used. Such an approach makes the possibility of more abstracted, i.e., orchestrator type independent, orchestration interface definition.
- In comparison to MANO interfaces, the MonB5G interfaces are enhanced to provide an exchange of information related to energy-aware operations and resource brokering.

[22] Also see: https://www.etsi.org/technologies/nfv?highlight=YToxOntpOjA7czozOiJpc2ciO30=.

As it has been already mentioned, the MonB5G architecture is a reference one, the interfaces are not defined in a detailed way – their definition is implementation-dependent. In fact, in most cases, proper protocols have been defined and such protocols, in general, are not seen as a part of the architecture. Therefore, an outline only of the specification of the interfaces of MonB5G is provided. The interfaces (or reference points) of the architecture of the static components of the architecture are listed in Table 4, whereas their more detailed description is presented in the forthcoming subsections (orchestrator interfaces are described in section 3.5.2, while MonB5G interfaces used in the context of energy related operations are presented in section 3.7).

Table 4. Interfaces of the MonB5G framework.

Interface	Type	Main functions	Relation
Iop	Web interface	• MonB5G system management.; • System-level KPI monitoring.; • Overall framework reconfiguration in case of failures.	MonB5G System Operator – MonB5G Portal
Itp	Web interface	• Slice selection (or infrastructure-oriented management functions) and its LCM.	Slice Tenant/Infrastructure Provider – MonB5G Portal
Imt	Web interface	• Slice Tenant interface for runtime slice management.	Slice Management Provider – Slice Tenant
Its	Web interface	• Slice runtime management (interactions with IDSM or SML).	Slice Tenant – IDSM
Imp	Web interface	• MLaaS LCM.	Slice Management Provider
Ipi	Web interface	• Actions involving the negotiation of the slice deployment policies that are used for LCM of multi-domain slices (MonB5G Portal interaction with IDMO).	MonB5G Portal – IDMO
Iid	Os-Ma-nfvo-like interface	• LCM management of slices implemented by DMO.; • Extended NFV Os-Ma-nfvo interface. • May provide LCM abstractions and provides to IDMO data and management capabilities of DMO.	IDMO – DMO
Idr	DMO-IDM interface	• Allocation, update or deallocation of resources. • Exchange additional information about the infrastructure, for example, about energy consumption and cost (between DMO and IDM);. • Extended Virtual Infrastructure Manager (VIM) – NFVI interface.	DMO – IDM
Iii	Web interface	• Infrastructure domain management (by Infrastructure Provider).	IDM – Infrastructure Provider

3.6.1 MONB5G SYSTEM OPERATOR – MONB5G PORTAL (I_{OP})

MonB5G Portal exposes I_{op} web-based interface (e.g., REST-based) to the MonB5G System Operator for the purpose of the MonB5G System management. The interface provides the means for:

- System-level KPI monitoring that reflects the overall health of the system, status of dynamic (slices) and static components (e.g., IDMO, DMOs) of the architecture, system stability, etc. This information is acquired from the lower-level entities, i.e., IDMs, DMOs and IDMO and propagated to the MonB5G Portal via the respective entities' interfaces (already described in this section).
- Reconfiguration of the overall framework in case of instabilities, failures, SLA violations, etc. The available functions should enable the operator to dynamically connect new DMOs to IDMO (register DMO handlers), as well as provide means for enforcement of slices migration to other domain, e.g., in case of DMO malfunctions or security violations.

3.6.2 SLICE TENANT/INFRASTRUCTURE PROVIDER – MONB5G PORTAL (I_{TP})

The MonB5G Portal exposes an interface to be used by Slice Tenants and Infrastructure Providers (which in that case act as Slice Tenants) for the purpose of slice LCM (creation, update, termination). This interface provides the Slice Tenant with information about specific slices deployment options, including instantiated VNFs, provided management functionalities, management automation algorithms, accounting estimation, etc. Moreover, I_{tp} provides functionalities for authentication and authorisation of slice tenants. It should be noted that during slice runtime, this interface is not used, and another interface is provided to the Slice Tenant/Slice Management provider.

3.6.3 SLICE MANAGEMENT PROVIDER – SLICE TENANT (I_{MT})

This interface is used by Slice Management Provider for the communication with Slice Tenant for the purpose of runtime slice management. This interface is in use when the Slice Tenant is not involved in slice management and the slice management is done by Slice Management Provider. The interface provides to Slice Tenant essential information about slice status, including violation of slice KPIs, etc. The information is properly visualised. Moreover, some reconfiguration slice requests can be send using this interface. In general, the amount of exchanged management information is minimal.

3.6.4 SLICE TENANT – IDSM/SM (I_{TS})

The I_{ts} interface is used by Slice Tenant for the purpose of runtime slice management. It can be used for interaction with both, IDSM and SML (if a slice is a single-domain slice). This interface does not have to be defined formally, but it is assumed that it provides a comfortable, high-level management interface. It provides visualised information about slice status, including violation of slice KPIs, etc. Moreover, slice reconfiguration requests in the form of intents can be send using this interface. The way in which the interface is handled is dependent on the slice template provider.

3.6.5 SLICE MANAGEMENT PROVIDER – MONB5G PORTAL (I_{MP})

This interface is used for the purpose of MLaaS LCM. It has to be noted that MLaaS is not triggered by the Slice Management Provider but indirectly by a Slice Tenant who requests deployment of the slice, which management is implemented as MLaaS. Using this interface, Slice Management Provider may interact with MonB5G System Operator for passing the information about the new or terminated slices (composed of SFL part only) that are managed by this Slice Management Provider.

3.6.6 MONB5G PORTAL – IDMO (I_{PI})

The I_{pi} interface is used by the MonB5G Portal to interact with IDMO. The interactions involve the negotiation of the slice deployment policies and are used for LCM of multi-domain slices. The interface is used for transferring to the portal essential information for MonB5G System Operator, Slice Tenants and Slice Management Providers.

3.6.7 IDMO – DMO (I_{ID})

The I_{id} interface is used by IDMO and DMO for multiple purposes:

1. For the preparation phase of slice deployment. For that purpose, IDMO interacts which DMO checking the possibility of slice deployment. If it is supported, the resource brokering process is triggered, which main goal is to find the optimal selection of resources or infrastructure provider, selection of a part of slice template that should be deployed in the specific domain, etc.
2. For LCM management of slices implemented by DMO. It can be seen as an extended NFV Os-Ma-nfvo interface. It may provide LCM abstractions and provides to IDMO data and management capabilities of DMO. It can be assumed that in some implementations, for the sake of implementation simplicity, the Os-Ma-nfvo interface will be exposed to IDMO. These functions include (according to NFV MAN-001 [54]):

a. Network Service Descriptor and VNF package management.

b. Network Service instance lifecycle management:

   i. Network Service instantiation;

   ii. Network Service instance update (e.g., update a VNF instance that is comprised in the Network Service instance);

   iii. Network Service instance query (e.g., retrieving summarised information about NFVI resources associated with the Network Service instance or to a VNF instance within the Network Service instance);

   iv. Network Service instance scaling;

   v. Network Service instance termination.

c. VNF lifecycle management:

   i. For VNF lifecycle management, NFVO identifies VNFM and forwards such requests (see Or-Vnfm description).

d. Policy management and/or enforcement for Network Service instances, VNF instances and NFVI resources (for control of authorisation/access, reservation/placement/allocation of resources, etc.).

e. Querying relevant Network Service instance and VNF instance information from OSS/BSS.

f. Forwarding of events, accounting and usage records and performance measurement results regarding Network Service instances, VNF instances, and NFVI resources to OSS/BSS, as well as and information about the associations between those instances and NFVI resources, e.g., number of Virtual/Containerised Machines (VMs/CMs) used by a certain VNF instance.

3. It can be used for DMO registration (obtaining DMO handlers) if such a possibility is supported.
4. It is used as information relay from IDM to IDMO regarding the cost, performance, energy efficiency, KPIs, SLA violations alerts and other important factors that can impact slice deployment (template split, polling of accounting information (stored in the Accounting Database) regarding a slice from respective DMOs (IDMs).
5. Interactions between security components of IDMO and DMO, i.e., E2E-ASO and D-ASO.

3.6.8 DMO – IDM (I_{DR})

This interface is used by DMO to allocate, update or deallocate resources. This interface is also used to exchange between DMO and IDM additional information about the infrastructure, for example, about energy consumption, cost (time-of-day dependent). It may be involved in resource brokering negotiations. In general, the interface can be seen as extended VIM – NFVI interface, which is used for exchanges to support:

- Allocation of VM/CM with the indication of compute/storage resource;
- Update of VM/CM resources allocation;
- Migration and termination of VM/CM;
- Creation, configuring, and removing of connections between VMs/CMs;
- Exchange of configuration information, failure events, measurement jobs, and their results, and usage records regarding NFVI (physical, software, and virtualised resources) to DMO.

3.6.9 *IDM – INFRASTRUCTURE PROVIDER (I_{II})*

The interface is used by the Infrastructure Provider to manage its infrastructure domain. This interface is implementation-specific. Moreover, the definition of IDM is not in the scope of MonB5G.

3.7 Architecture components related to energy efficiency

Within the MonB5G architectural paradigm, Management Layer as a Service (MLaaS) reduces SML resources footprint. This is because a single MLaaS may provide service to many Slice Functional Layers (SFLs). It follows directly that this is the approach that needs to be followed when targeting Energy Efficiency (EE) goals.

Focusing first on the components enabling Energy Efficiency (EE) strategies, it is worth to mention those related to the Infrastructure Domain, i.e., IOMF, IDM. IOMF may indeed be placed within MLaaS and leverage IDM reference points to e.g., retrieve infrastructure telemetry and perform actions tailored to EE, like turning off/on compute nodes. IOMF may also provide reference points to MS, as to make infrastructure-specific telemetry available to AI/ML-based components (i.e., AE, DE). Specific EE procedures, such as turning off compute nodes, necessitates the involvement of DMO as SFL components must be migrated before the turning off operation. Moreover, the MS/AE/DE triplet of SML forms a closed control loop, where feedback interfaces have been defined between DE and AE, DE and MS and AE and MS as shown in Figure 24. The role of these interfaces is to reconfigure MS and AE to achieve EE and scalability goals. Specifically, DE may implement a stochastic policy to select which subset of AEs participate in the e.g., federated learning (FL) training task during a certain number of rounds based on their local performance. For instance, AEs having accurate models that yield low SLA violation prediction can be assigned a higher probability to take part in the training. In this regard, DE needs to communicate an activation bit to all the local AEs via the feedback interfaces.

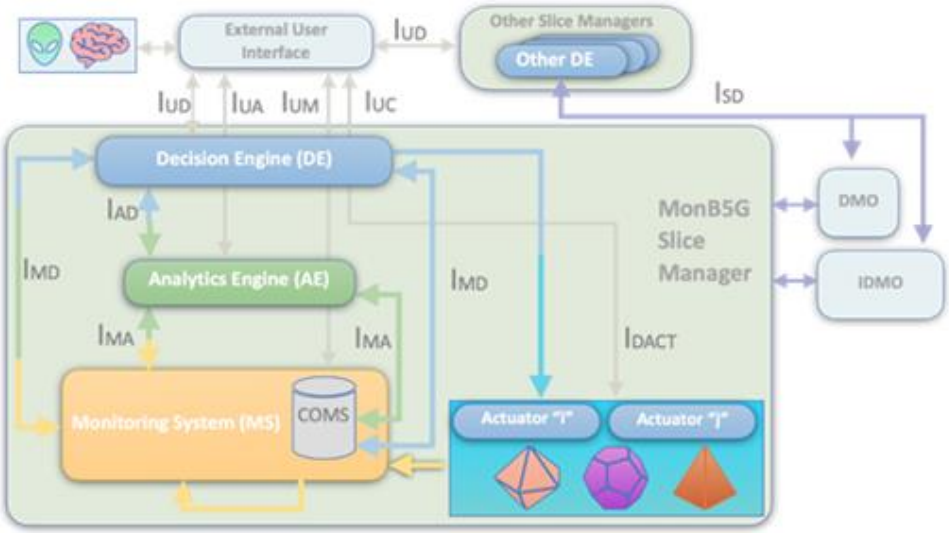

Figure 24. SML internal interfaces.

Table 5 provides a description of the different interfaces that link MonB5G DE with the other control blocks.

Table 5. Description of the type of DE interfaces and the associated role.

Interface	Type	Role
I_{AD}	Tensors/Database query	DE reads the predicted KPI from AE (either online or from COMS)
I_{MD}	Tensors/Database query	DE reads raw MS measurements (either online or from COMS)/Store AI metrics and DE decisions in COMS
I_{MA}	Tensors/Database query	AE reads raw MS measurements/Store AI metrics, predictions in COMS
I_{UD}	Database Query	EUI reads/changes DE configuration (e.g., discount factor of a DRL algorithm[23])
I_{UA}	Database Query	EUI reads/changes AE configuration (e.g., prediction interval, learning rate)
I_{UM}	Database Query	EUI reads/changes MS configuration (e.g., granularity)
I_{UC}	Database Query	EUI reads/changes actuation configuration (e.g., API primitives' parameters)
I_{DACT}	REST API Call	DE sends decisions to Actuators

3.7.1 ENERGY-AWARE SERVICE DYNAMICS

Because of the advent of virtualisation into the telco world, a big part of 5G and beyond 5G communication infrastructure will be composed of data centres spanning the Cloud and Edge technological domains. Either for hosting VNFs, enabling URLLC slices, or hosting virtual RAN functions, the capabilities brought by virtualisation define the road forward.

Energy-efficiency is a major key performance indicator for the sustainability of beyond 5G networks. In this regard, network resource management algorithms should achieve the best quality of service (QoS) with minimum energy consumption. This involves network-level and slice-level strategies that require architectures with more flexibility/programmability in resource placement and allocation.

In this context, both SDN and NFV technologies are envisioned as the appropriate platform to deploy optimisation models (based either on AI or heuristic approaches) and management functions enabling energy-aware network slicing operations. Based on energy consumption estimations or network parameters information (e.g., traffic load, radio coverage, equipment activation intervals, or active users), the SDN/NFV architectural framework can carry out actions such as optimised routing of traffic flows or allocation of physical (networking, computing, and storage) and/or dynamically scale-in/out Virtual Network Functions (VNFs) to meet the desired energy savings for each slice [70]. Having said that, a set of conceptual requirements should be met:

- Priority order for services within a network slice in the case the available energy at a certain time is insufficient to meet all demand. Based on this level of priority, the infrastructure provider must be able to differentiate the access to the energy resource and the application of management strategies for each service [7][9][71],
- Services belonging to network slices must be able to work in energy-saving states (e.g., low energy consumption) or suspend their execution in case of low activity (i.e., enter into sleep or idle mode) according to the availability [71][72][73],
- Measurement of metrics or indicators to assess the energy efficiency achieved with the proposed energy management model [74]. This includes the available power (managed by the infrastructure provider) to be fed to the AI models for making decisions.
- Energy-aware decisions (e.g., VNF placement, scaling, idle modes) taken by AI algorithms (generally Deep Reinforcement Learning (DRL) schemes) should be enforced via MANO API calls.

In this framework, several solutions have emerged recently. Specifically, in the context of a cooperative multi-operator, 5G network based on virtualised radio access and core, a sleep-mode and spectrum-

[23] For further information about Deep Reinforcement Learning algorithms also see, for example: https://arxiv.org/abs/1906.10025.

sharing strategy to minimise the base station (BS) power consumption has been presented [75]. The proposed dynamic inter-operator spectrum–sharing formulation is cognizant of inter-RAN traffic volume to motivate mobile network operators (MNOs) to cooperate to achieve energy efficiency in their RANs. In this intent, the inter-operator joint optimisation problem is formulated to obtain power-efficient intra- and inter-RAN beamforming vectors for supplementary energy gains and improved UE signal reception. Meanwhile, a distributed Q-learning algorithm has been introduced [76], which chooses how deep a BS can sleep according to the best switch-off sleep mode (SM) level policy that maximises the trade-off between energy savings and system delay and may reach an energy saving of 90% when users are delay tolerant. Moreover, as cell load impacts its energy-efficiency, a multi-agent online reinforcement learning-based traffic offloading algorithm has been introduced [77], which benefits from the awareness about other macro-cells offloading strategies to improve the quality of the selected traffic offloading action without explicit information exchange. This yields 14% improvement in network energy-efficiency. In the same direction, a joint energy-aware deep Q-network (DQN) traffic offloading and demand forecasting strategy has been presented [78], which leverages an open dataset from a major telecom operator to train BSes' control model leading to 5% energy-efficiency gain compared to native Q-learning.

Leveraging Network Function Virtualization (NFV), energy-efficient Integer Linear Programming (ILP)-based dynamic network functions placement has been proposed [79], which can adapt the joint locations of Centralised Unit/Distributed Unit (CU/DU) and MEC to the actual distribution of network processing and transport resources so that to aggregate CUs/DUs into fewer cloud servers, resulting thereby in 20% energy saving. Further research into the Edge domain tends to favour small form-factors due to their lower energy consumption (mostly due to the physical restriction imposed by the deployment locations). These are mainly the reasons why Multi-access Edge Computing (MEC) platforms are coming from the OS container (e.g., Docker, Container Orchestration Engines for the Edge like KubeEdge[24], etc.) direction, instead of that from the traditional Virtual Machines (VMs), whose LCM is delegated to big software, such as OpenStack[25].

It is then in the Cloud Technological Domain where the potential for energy-savings strategies could thrive, mainly due to the wide availably of: (i) tools inherited from decades of data centre management (e.g., VIM plugins for energy quotas); (ii) protocols (e.g., Preboot Execution Environment[26] – PXE), and; (iii) the possibility to redistribute allocated resources conditioned by service-level KPIs via NFV MANO reference points and Objects (e.g., PaaS).

MonB5G Architecture essentially generalises the elements contained within its Functional Layer. That is, MonB5G management layer can be tuned for new models of infrastructure management. Leveraging policies dictated at the Inter-Domain level, the entire collection of Network Services (NSs) on a determined Infrastructure Domain may be subject to MonB5G Administrative Elements; the latter being configured towards, e.g., efficient resource placement and compatible data centre energy management strategies.

[24] KubeEdge is an open source system for extending native containerized application orchestration capabilities to hosts at Edge. It is built upon Kubernetes and provides fundamental infrastructure support for network, app. deployment and metadata synchronization between cloud and edge. For further details see: https://kubeedge.io/en/.

[25] OpenStack is an open source platform that uses pooled virtual resources to build and manage private and public clouds. For further information see: https://www.openstack.org/.

[26] See: https://en.wikipedia.org/wiki/Preboot_Execution_Environment.

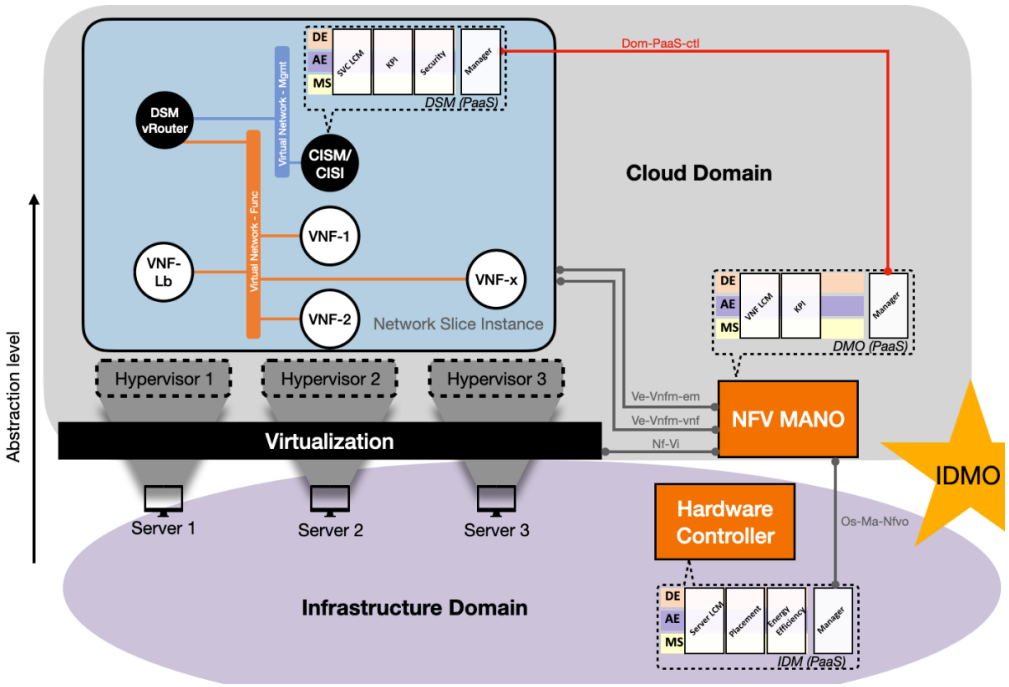

Figure 25. DE interactions.

Figure 25 serves as a first step towards the implementation of a data-driven energy management system leveraging MonB5G Architecture, i.e., the orchestration capability of Infrastructure Domain Manager (IDM). This solution will leverage APIs and operations provided by data centre management tools, e.g., MaaS, FOG Project[27] or OpenStack Ironic[28] to control as well as extract relevant metrics from a pool of servers (i.e., Hardware Domain in the figure), i.e., Hardware Controller. Furthermore, MonB5G Administrative Elements from IDM can then extract relevant metrics from Hardware Controller, while enabling very valuable inter-domain optimizations such as dynamic Network Functions Virtualisation Infrastructures (NFVIs).

Figure 25 exemplifies this by showing Cloud and Infrastructure Domains compatible with MonB5G Architecture, while the corresponding SML deals with service-level optimizations and lifecycle management. In turn, DMO leverages SML for resource-oriented optimizations or policy enforcement operations (e.g., VNF migration, VNF placement, NS admission control, etc.). Relevantly, IDM may predict valuable overall energy savings if alternative placement directives or resources migration is executed and therefore allows for the powering off of Servers at the Infrastructure Domain. The converse would also be true: failing to admit a Network Slice due to resource insufficiency, IDM may be instructed by other Technological Domain's DEs (e.g., DMO) to increase the pools of Servers in the Infrastructure Domain, and consequently the virtualization capacity of NFVI.

[27] See: https://fogproject.org/.
[28] See: https://wiki.openstack.org/wiki/Ironic.

4 Domain-specific MonB5G architecture instantiation

As introduced earlier, the MonB5G architecture considers several technological domains when deploying an E2E network slice. An E2E network slice is composed of several sub-slice instances that will be run on different technological domains. The life-cycle management procedures of each sub-slice are executed and handled by DMO. This section gives possible mappings of DMO and its closed-control loops components to two main technological domains, cloud and RAN. The objective here is to provide a step forward vision to the implementation of the MonB5G architecture components when considering the cloud and RAN domains, particularly in the context of two well-adopted architectures of ETSI NFV and O-RAN, respectively.

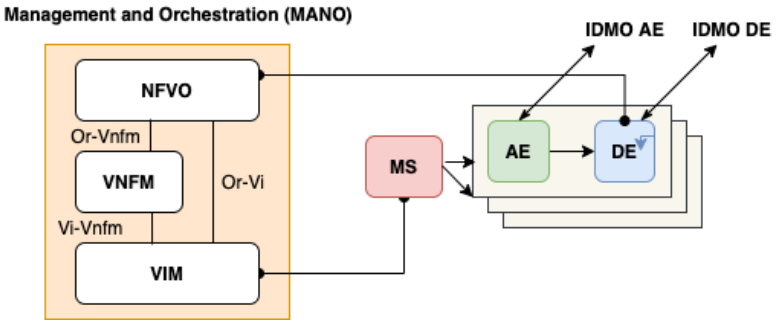

Figure 26. DMO of cloud domain.

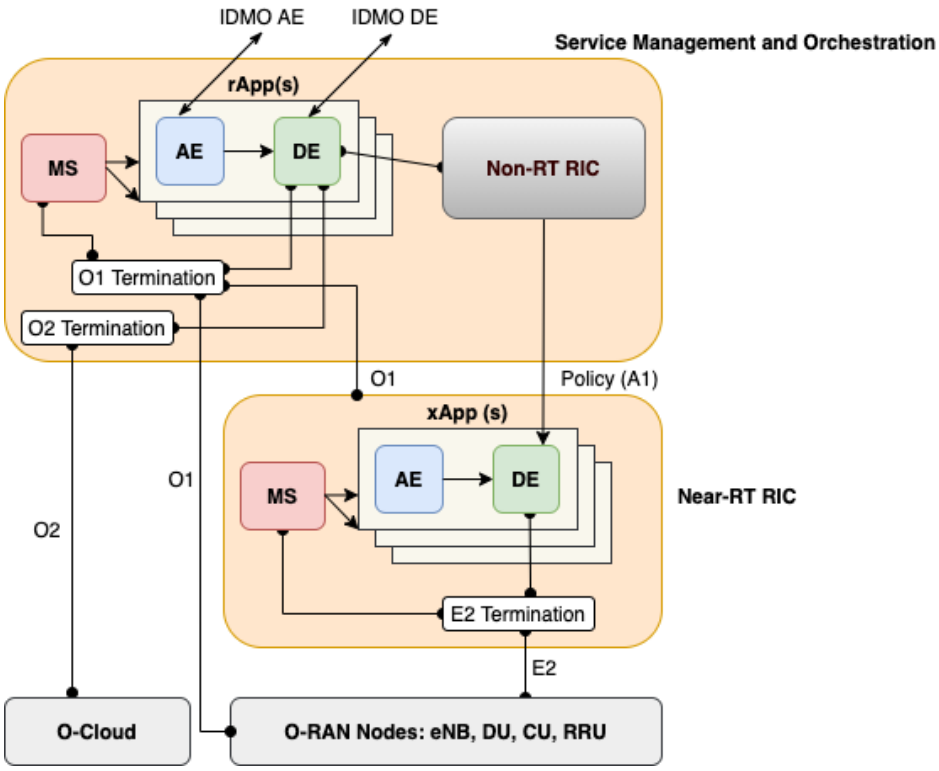

Figure 27. DMO of RAN domain.

In Figure 26 and Figure 27, we illustrate the mapping of DMO with its closed-control loops for two different technological domains, cloud and RAN, respectively. For the cloud domain we use the well-known NFV architecture, while for the RAN domain we consider the emerging O-RAN architecture[29].

In the case of the cloud and edge (cf. Figure 26), DMO corresponds to the NFV MANO entity, consisting of the NFV Orchestrator (NFVO), VNF Manager (VNFM), and Virtual Infrastructure Manager (VIM).

[29] For further information also see: https://www.o-ran.org/.

Monitoring information is collected by MS from VIM covering mainly infrastructure-level metrics: such as CPU and memory usage, consumed throughput, and packets per second; organised per VNF or aggregated per sub-slice. Besides, the NFVO may expose available management actions that can be performed on a cloud or edge sub-slice, which DE can use. We can mention VNF scale-in or out, VNF migration, block incoming or outgoing VNF, instantiate a new VNF, etc. Typically, all the actions are related to VNFs' LCM. It should be noted that DE assisting MANO may refer to the IDMO's DE, if a local decision is not enough to resolve an issue. For example, if no more CPU is available at the infrastructure level or need to migrate one VNF between two VIMs, the local DE may delegate the decision to the IDMO's DE that may request more details from IDMO's AE or use a local policy to select a new VIM to serve the sub-slice, using a multi-domain placement algorithm [80]. Finally, MS is common for all running AE/DE; it collects monitoring data, formats, and aggregates data to be consumed by subscribed AEs.

Regarding the RAN domain (cf. Figure 27), DMO encloses Service Management and Orchestration (SMO) and the Near Real-Time RAN Intelligent Controller (Near-RT RIC). The closed-control loops are run as rApps (i.e., applications that run Non-RT RIC) in SMO for everything related to non-RT management loops, such as Fault-management, the configuration of CU/DU, and LCM of xApps (i.e., applications that runs Near RT-RIC); and as xApps for everything related to Near-Real time management functions, such as MAC scheduling, Mobility management, Radio resources management, etc. It should be noted that according to O-RAN the O-Nodes correspond to RAN functions, either run as a monolithic block (gNB[30]), as PNFs, or disaggregated functions: CUs, DUs, and Radio Units (RU)s). CU and DU may run as VNFs on top of a cloud or edge infrastructure (noted as O-Cloud), while RU is a physical component that runs as a PNF.

Closed-control loops running in SMO collect monitoring data on O-RAN nodes in addition to Near-RT RIC components and O-Cloud, using O1 and O2 interfaces, respectively. O1 allows collecting data on xApps performances, logging, and monitoring on the status of Near-RT RIC internals components, radio information extracted from CU/DU/RU. On the other hand, O2 provides cloud-oriented information similar to what VIM provides, focusing only on VNFs running CU and DU, such as consumed CPU and memory, exchanged traffic, etc. MS is in charge of collecting the monitoring data using these two interfaces, while AEs will subscribe to the data of interest. DEs may enforce the derived decisions using either O1 or O2. The actions are related to the configuration of CU/DU – (for instance, turn-off DU to save energy or change the configuration of the Time Duplex Division Duplex (TDD) pattern –) or on the scale-up/down of VNFs running CU/DU. Moreover, DE may use the A1 interface, via the Non-RT RIC to push a new policy toward the running xApps, for instance, to increase radio resources dedicated to a slice, or change the 5G New Radio numerology or update the MAC scheduler of a network slice.

[30] Node B is the radio base station for 3G UMTS (Universal Mobile Telecommunications System), while eNodeB is the radio base station for 4G LTE (Long Term Evolution). The gNB is the logical 5G radio node, the equivalent of what was called NodeB in 3G-UMTS and eNodeB or eNB (i.e., evolved Node B) in 4G-LTE, is now called as the "next generation NodeB".

5 MonB5G architecture usage scenarios

In this section, two scenarios of usage of the MonB5G architecture will be presented. The first scenario concerns the basic operations of the framework related to E2E slice deployment and real-time management, whereas the second scenario deals with the usage of the framework for security purposes.

These two usage scenarios respectively illustrate some procedures that can be triggered in the MonB5G planned proof-of-concepts PoC1 and PoC2, namely,

- PoC1: Zero-Touch Network and service management with end-to-end SLAs,
- PoC2: AI-assisted policy-driven security monitoring & enforcement,

wherein the necessary elements of the architecture will be implemented, integrated and tested as discussed in Section 7.

5.1 E2E slice lifecycle and runtime management scenario

The MonB5G System operations can be split into the following phases:

- **Phase 1: Slice Negotiation Phase** – establishment of the contract between the Slice Tenant and MonB5G framework regarding the slice deployment details and pricing.
- **Phase 2: Slice Preparation Phase** – preparation of the selected slice template based on the Slice Tenant preferred configuration to enable E2E inter-domain operation.
- **Phase 3: Slice Deployment Phase** – slice deployment in multiple orchestration domains by IDMO and DMOs.
- **Phase 4: Slice Runtime Phase** – operation of the slice and its management by the Slice Tenant.
- **Phase 5: Slice Termination Phase** – slice removal and resources deallocation.

Each of the above-mentioned phases is described in detail in the forthcoming subsections.

5.1.1 SLICE NEGOTIATION PHASE

In this phase, shown in Figure 28, the Slice Tenant interacts with the MonB5G Portal in order to create its slice. In the first step of the phase, the tenant performs authentication procedures. In the next step, the tenant fills the Slice Request Form (SRF) online. SRF is composed of a selected slice template, its parameters, the way in which the slice will be managed (by Slice Tenant of Slice Management Provider). In this step, GSMA GST/NEST with some modifications can be used[31]. The slice deployment time and expected duration and priority are also filled in the form. When SRF is completed, it is passed by the MonB5G Portal to IDMO. IDMO analyses the request and interacts with DMOs to check the possibility of deployment of subnetwork slice (single domain slice) in each of the domains that are needed for the deployment of the slice. DMOs check if the request can be satisfied. To that end, DMO checks the availability of the resources of the Infrastructure. If supported in this phase, the resource brokering procedure is triggered. In this procedure that can be auction-based, DMO interacts with Infrastructure Providers to select the resources in an optimal way in terms of cost and energy consumption. DMO reports the resource availability and their price to IDMO. IDMO compares the information about resources obtained from multiple DMOs and checks whether a single domain slice can be deployed using resources of a single Infrastructure Provider or the resources of multiple infrastructure providers (for example due to lack of sufficient resources in each of the domains or pricing aspects) have to be used. In the latter case, it makes the split (partition) of the slice template into two domains. The process is repeated for each orchestration domain that has to be involved in the E2E slice deployment. At the end of this phase, IDMO sends to the MonB5G Portal the information about possible options of slice deployment with related costs. On that basis, the Slice Tenants select the preferred variant of slice deployment. The main difference between the variants will be the estimated price that can be related to the time when the slice

[31] For further details see the discussion in: GSM Association (2020, November): Generic Network Slice Template. Version 4.0. Available at: https://www.gsma.com/newsroom/wp-content/uploads//NG.116-v4.0-2.pdf.

can be deployed. After the selection of slice deployment, the phase is ending. If there are no options to deploy the slice, the procedure is aborted and the Tenant is informed.

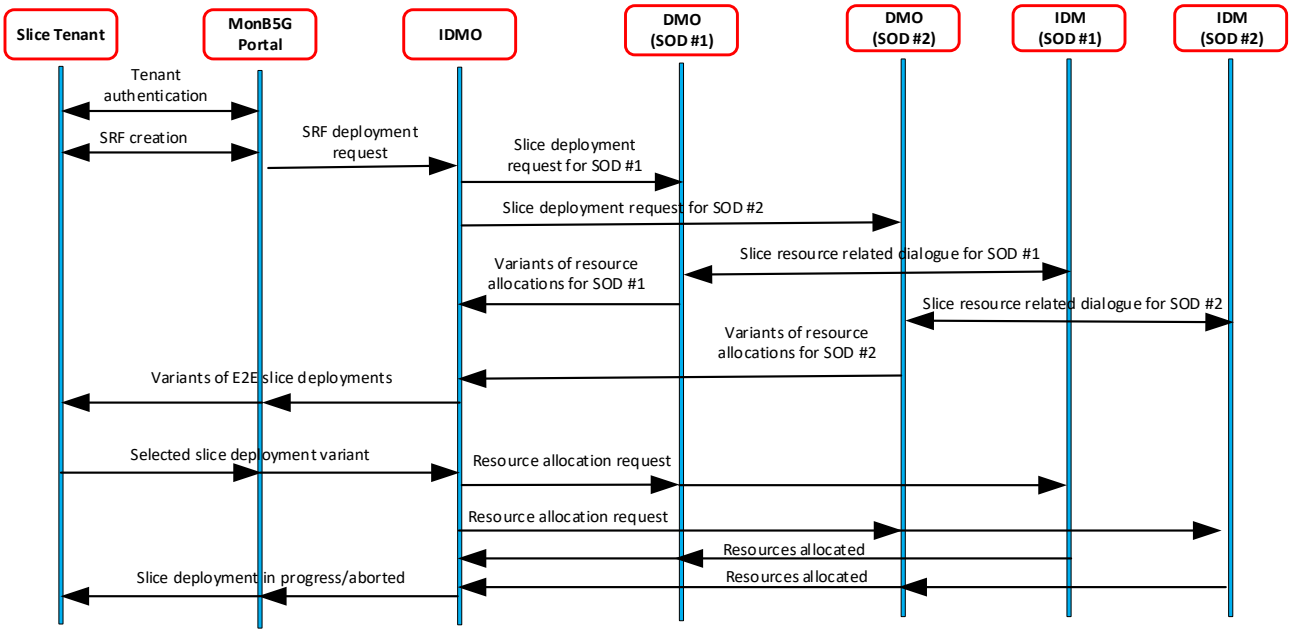

Figure 28. Simplified message sequence chart of slice negotiation phase.

In case when the slice has to be deployed immediately (a case of an emergency slice) the procedure can be significantly simplified as in such case deployment is not price-sensitive and Slice Request Form can be prepared a priori. Therefore, the time devoted to this phase will be much shorter.

5.1.2 *SLICE PREPARATION PHASE*

In this phase IDMO knows the slice deployment variant accepted by Slice Tenant, that comes with accepted SRF and a set of parameters that were set up during slice negotiation phase. It also knows that the slice can be admitted in this configuration. In the first step the slice template is partitioned into multiple SODs if this operation is needed. This operation involves the addition of the components to the slice template that are needed for the E2E slice operations such as IDSM and entities needed for inter-domain data transfer (marked as Slice Template Update[32], cf. Figure 29). In the next step, the configuration data of SM of each domain and IDSM are updated according to the split. After splitting and initial configuration of slice, single domain templates the templates (SFL and SML parts) and their parameters are sent to respective DMOs. In case the template is already known to DMO and is stored in the DMO database of templates, only the slice configuration is transferred. This is the last step of this phase.

[32] GSM Association (2021, June: Generic Network Slice Template. Version 5.0. Available at: https://www.gsma.com/newsroom/wp-content/uploads//NG.116-v5.0-7.pdf.

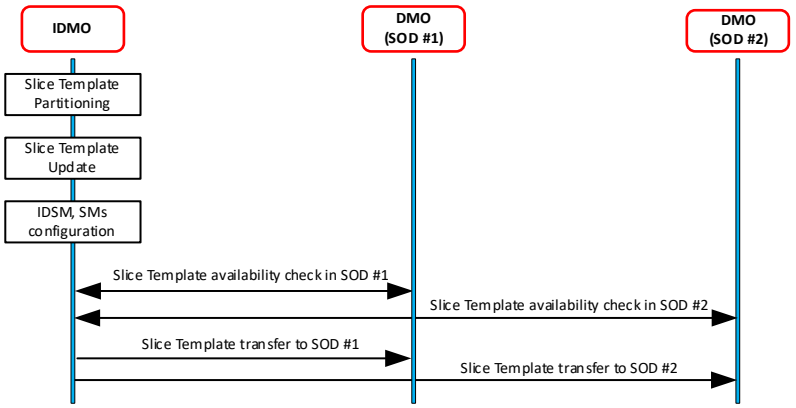

Figure 29. Simplified message sequence chart of slice preparation phase.

5.1.3 SLICE DEPLOYMENT PHASE

In the first step of this phase DMO of each domain deploys its part of the E2E slice template (cf. Figure 30). If it is needed, IDSM in a form of a slice (PaaS) is also deployed (cf. Figure 31 for a single domain deployment case). The way of the deployment is dependent on the technological domain. In the cloud domain, it is assumed that the ETSI MANO model will be used. Therefore OSS/BSS of DMO interacts with the NFV MANO orchestrator (that is also a part of DMO) in order to deploy the slice. The procedures described in [81] can be used for that purpose. In other domains (RAN) specific orchestrators can be used. After the deployment of slice VNFs, the slice configuration takes place. The process is governed by SM of each domain. For the purpose of node/functions configuration, SM interacts which respective EMs or EEMs. It is worth noting that EEM can provide autoconfiguration of its node/function and in such case the involvement of SM is marginal. When SM finishes the configuration of its slice it sends the information to IDSM that the part of the slice is ready for operations. When IDSM will collects such information from all SMs of the E2E slice it checks proper interslice information exchange via active testing and at the end it informs IDMO that the slice is ready to be used. This information also includes a link to IDSM that will be used by the Slice Tenant or Slice Management Provider for slice runtime management and orchestration.

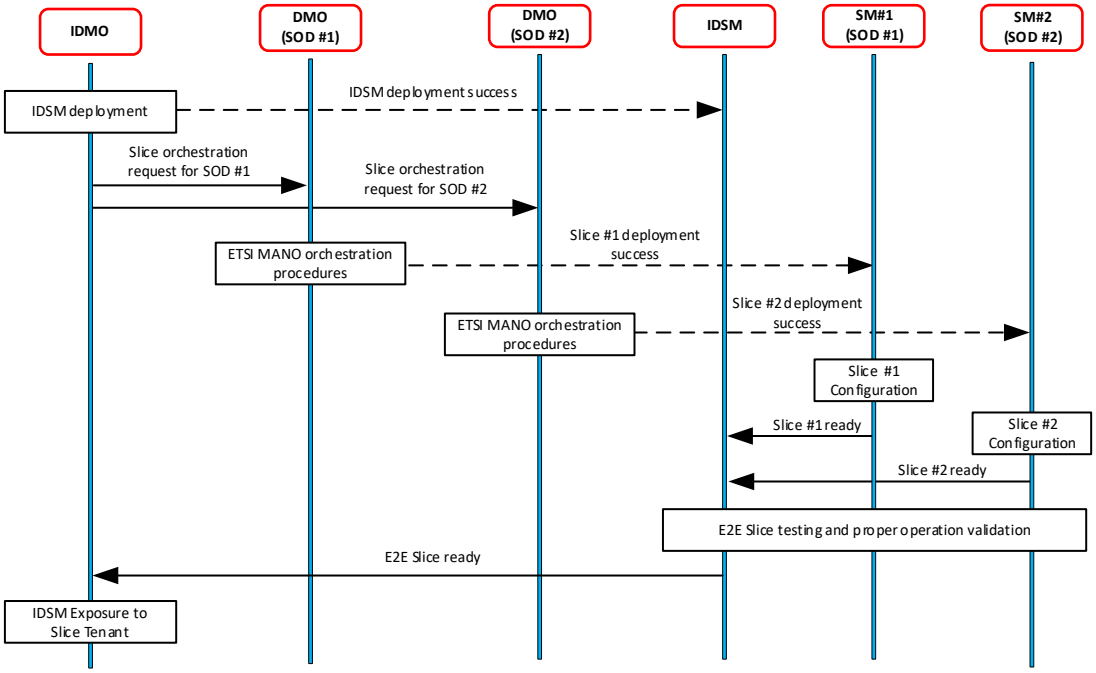

Figure 30. Slice deployment for the inter-domain cloud case (driven by ETSI MANO).

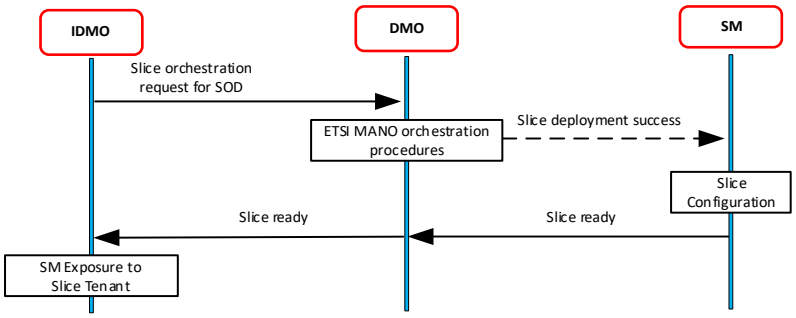

Figure 31. Slice deployment for the single domain cloud case (driven by ETSI MANO).

5.1.4 SLICE RUNTIME PHASE

During the runtime phase the slice is already operational and Slice Tenant – IDSM interface is active. During this phase the following information is periodically exchanged between IDSM, Slice Tenant and IDMO:

- SMs of each domain level slice reports this slice's KPIs to IDSM;
- IDSM, on the basis of reports of SMs, sends to IDMO and provides direct access to Slice Tenant E2E slice KPIs;
- DMOs report to IDMO slice-related resource consumption and billing-related information.

During slice runtime the following events can be triggered:

- Resource scaling triggered by SM. This resource scaling is based on DEs of slice SML. In such case the request is sent to DMO to change the allocation of resources to specific node/function or link. This type of resource scaling is proactive.
- Resource scaling triggered by VNFM. Such operation in general is seen as a backup as SM is responsible for resource scaling. It can be triggered if SM fails with proper resource allocation and it has been noticed by VNFM.
- Slice modification by adding/removing or moving to another location its function (VNF). In such a case SM sends request to DMO requesting slice template modification. The request is based on DE decision and may concern as well VNF of SFL or SML. SM uses Os-Ma-nfvo reference point interfaces and NS Update procedures in the NFV MANO case are applicable.
- Alerts in a direct form, i.e., signalled directly by some nodes/functions or triggered by AEs of SML of a slice. Such alerts can be handled by SM, DMO or IDSM, but in any case, information about the alert has to be passed to IDMO.
- Slice reconfigurations triggered indirectly by Slice Tenant or Slice Management Provider. In such case the IDSM sends to SM(s) intents that were generated by the mentioned business actors. In response the appropriate DEs should react to the changes according to their goals.

Figure 32 shows interactions between MONB5G components during slice runtime.

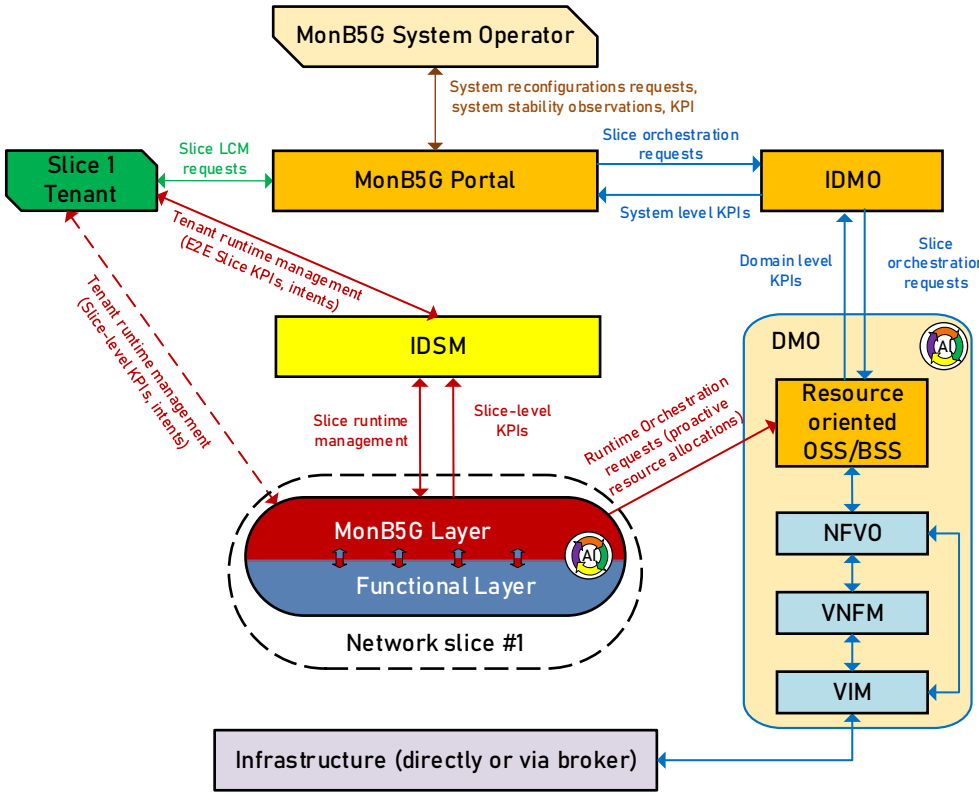

Figure 32. Interactions between MonB5G entities during slice runtime (the dashed line presents the case in which the Slice Tenant manages a single domain slice).

The exemplary workflows for some of the operations mentioned above are presented in Figure 33 and Figure 34.

Figure 33 describes the E2E multi-domain slice reconfigurations driven by intra-slice AI entities (a) and by SM. During the slice operation, the EEMs of domain slices transfer the collected VNF level KPIs to the SMLs of their domains. Afterwards, the data is processed and the local reconfiguration decisions are taken and enforced in order to optimise the operation of a slice. During the slice operation, the high-level slice KPIs that reflect the slice operation are accessible by the Slice Tenant. Reconfigurations can also be triggered by the Slice Tenant directly. After issuing the reconfiguration request, it has to be validated at the IDSM (or SM in case of a single domain slice) level to prevent possible slice malfunctions. Afterwards, the reconfiguration requests are sent to domain slices' SMLs and to appropriate EEMs. In case of reconfiguration failures, all of the changes are reverted to bring the slice back to its last state.

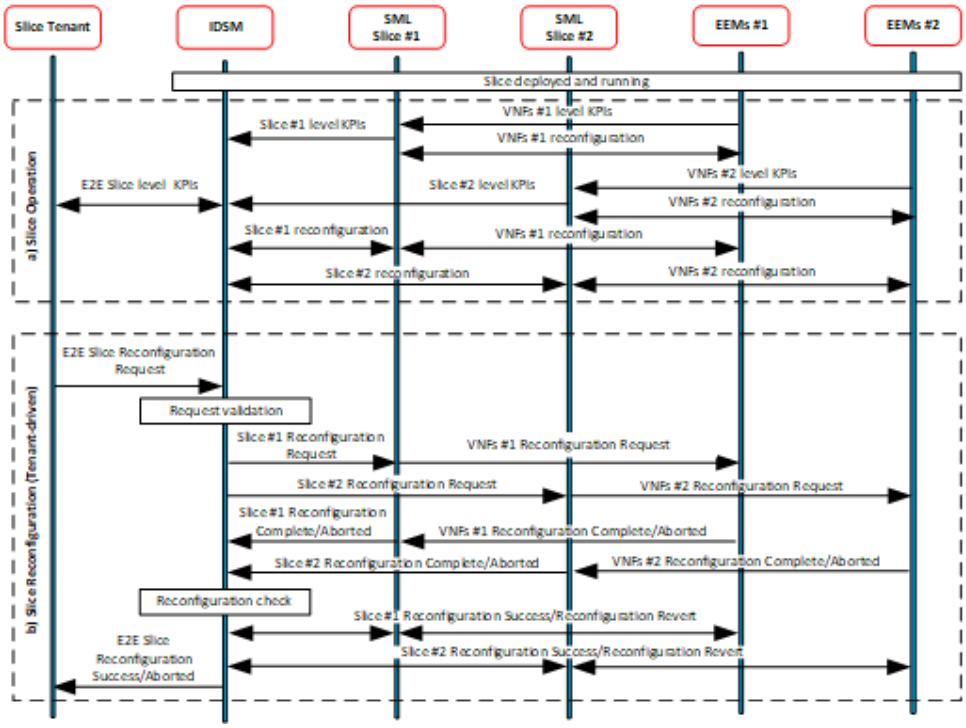

Figure 33 Workflow of E2E multi-domain slice (a) runtime and AI-driven reconfigurations by SMLs (b) reconfigurations driven by a Slice Tenant.

Figure 34 presents the example of the framework's actions to the risk of slice SLA violation related to: a) the shortage of resources required for slice VNFs proper functioning, and; b) the deployment of a supporting VNF (e.g., additional servers for the purpose of load balancing). For simplification, both examples have been presented in the context of a single domain slice. When SML of a slice detects the possibility of SLA degradation (e.g., due to analysis of alerts issued by EEMs), it can trigger orchestration operations to mitigate it, based on the root cause of the problem. In case of lack of per-VNF allocated resources, SM sends to DMO of its SOD the request to perform resources upscaling with the detailed information regarding its needs. DMO performs a check of available resources and, if viable, upscales slice VNFs according to its possibilities. Afterwards, both IDMO and Slice Tenant are notified about the performed upscale and its impact on charging. If the upscale is not possible, the slice migration to another SOD has to be performed, however, the details regarding this operation are implementation-dependent. The request of additional VNF orchestration is very similar to resource upscaling with one additional step i.e., EEM conducts the VNFs registration in SM. Both procedures have been presented in the context of SLA violation, however, they can be driven by any events defined in the slice template (e.g., downscaling can be performed to minimize slice costs).

In the case of multi-domain slices, the procedure is much more complex as it requires more interactions between IDMO, DMOs and IDMs to select the appropriate SODs, Infrastructure Providers, resource chunks etc. as well as communication between SMs and IDSM of a slice. Nonetheless, in this case the same approach is followed.

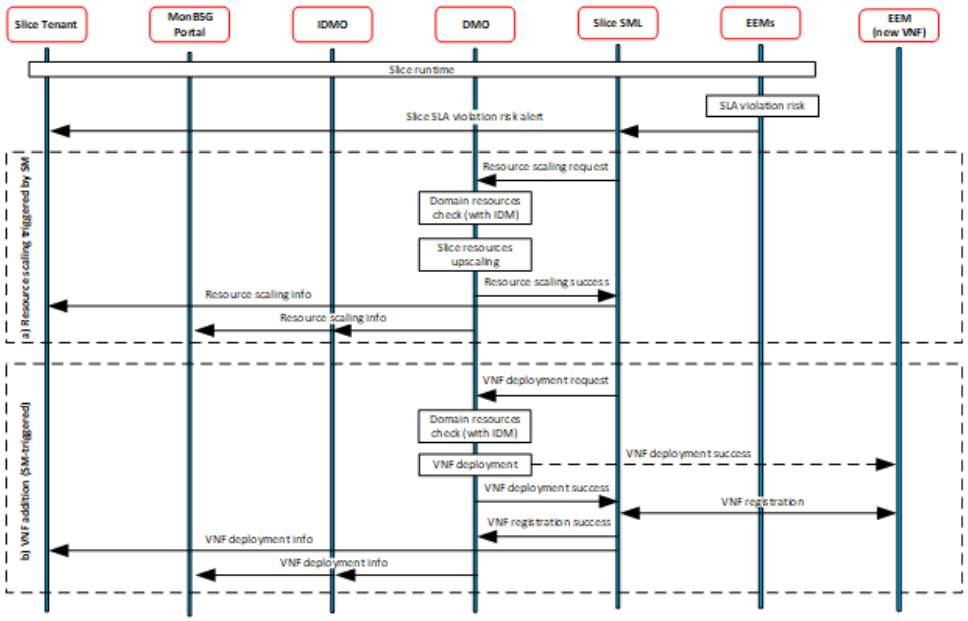

Figure 34 Examples of simplified SM-triggered procedures in case of SLA violation risk caused by a) the possible lack of resources assigned to slice VNFs, b) need for the deployment of an additional VNF(s).

5.1.5 SLICE TERMINATION PHASE

The slice Termination Phase (cf. Figure 35) starts on Slice Tenant request, or it can be triggered by IDSM in the case when abnormal behaviour of the slice has been noticed. In the first case Slice Tenant sends to MonB5G Portal the Slice Termination Request. The portal forwards the request to IDMO and IDMO starts the procedure of slice termination. To that end, it triggers a dialogue with IDSM and DMOs in order to finish the collection of information about the consumed resources by slice and their price (accounting data). In the next step slices of each domain are terminated and resources are released by respective DMOs. This also concerns IDSM. Before termination of IDSM, the information about that fact is sent to Slice Tenant/Slice Management Provider. After obtaining information from all DMOs that slices are terminated, IDMO informs MonB5G System Portal that slice is terminated, resources are released and the accounting data are collected.

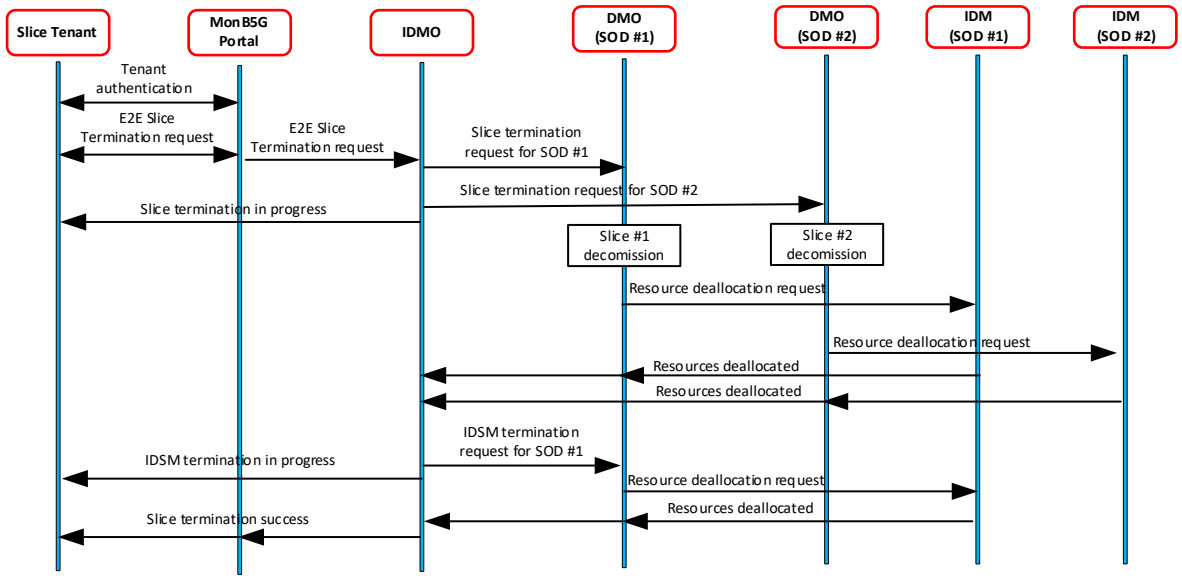

Figure 35. Slice termination phase.

5.2 E2E security management scenario

5.2.1 SECURITY INCIDENT MANAGEMENT

A cybersecurity framework (CSF) is mainly a set of standards, guidelines and best practices for managing the risks exposed to a system. CSF generally addresses security objectives, such as preserving the confidentiality and integrity of operations and data, as well as ensuring the continuity of services. Moreover, the framework is essential for the system provider which wants to comply with vertical, national, and international cybersecurity regulations.

The NIST Cyber Security Framework [82], the UK NCSC Cyber Assessment Framework guidance [82], Managing Cybersecurity of Industrial Control Systems published by the French Network and Information Security Agency (ANSSI) [83] as few instances of well-known cyber security frameworks, follow a substantially similar approach. The framework's core is a list of cybersecurity objectives that follow four main stages of cyber defence as exemplified in Figure 36:

(1) Identify and prepare;

(2) Prevent and protect against cyber-attack;

(3) Detect and triage cyber security events;

(4) Respond to cyber security incident and recover.

First, the framework outlines the procedures for the involved organisation to identify risks and key assets that require protection (1); then it lists the ways the organisation must protect these assets with counter-measures to mitigate risks (2); monitoring assets and detecting signs of compromise, analysing the impact, threat, resulting damage (3), and finally;, responding to incident by carrying out actions to mitigate or eradicate the incident, recovering assets and learning from the incident to prevent it from happening again (4).

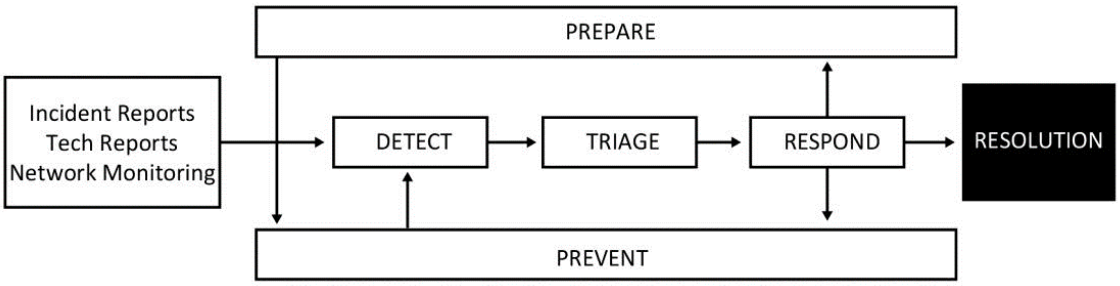

Figure 36. High-Level Incident Management Process Workflow proposed in New Zealand Security Incident Management Guide [85].

As applied to 5G systems, the network provider could play the role of the organization while the assets to be protected would be the delivered network slice including the underlying and building components such as network slice subnets, NFV objects and virtualised resources.

5.2.2 NETWORK SLICE LIFECYCLE ALONG CYBER SECURITY MANAGEMENT

Network slice security management is closely tied to the security incident management as presented previously in 5.2.1 and to the network slice lifecycle management.

The basic pattern of network slice security management when applying cyber security framework consists of sub-cases related to the network slice lifecycle. The network slice lifecycle management (LCM) as specified in [9] consists basically of the below pattern of four main phases:

(1) The preparation phase;

(2) The commissioning phase;

(3) The operation phase;

(4) The decommissioning phase.

During preparation of the network slice based on a Network Slice Template (NST) and the customer requirements received, the assets, the associated threats and risks will be identified, understood and assessed (1). According to the mitigation strategies supported by the experts, the security enhanced NST (called se-NST) including new requirement parameters is generated and it contains the appropriate protection solutions (2) and associated security policies. The solutions contain the proactive defence measures as well as the reactive functions of Detect (3) and Respond (4) of the cyber security framework as shown in Figure 37.

In the next three stages of the network slice LCM (commissioning, operation, decommissioning), the security enhanced network slice (se-NSI) is instantiated from the se-NST model. In this process, security functions are managed (creation, configuration, update, scaling, termination) jointly with the lifecycle of the network service instance associated to the se-NSI.

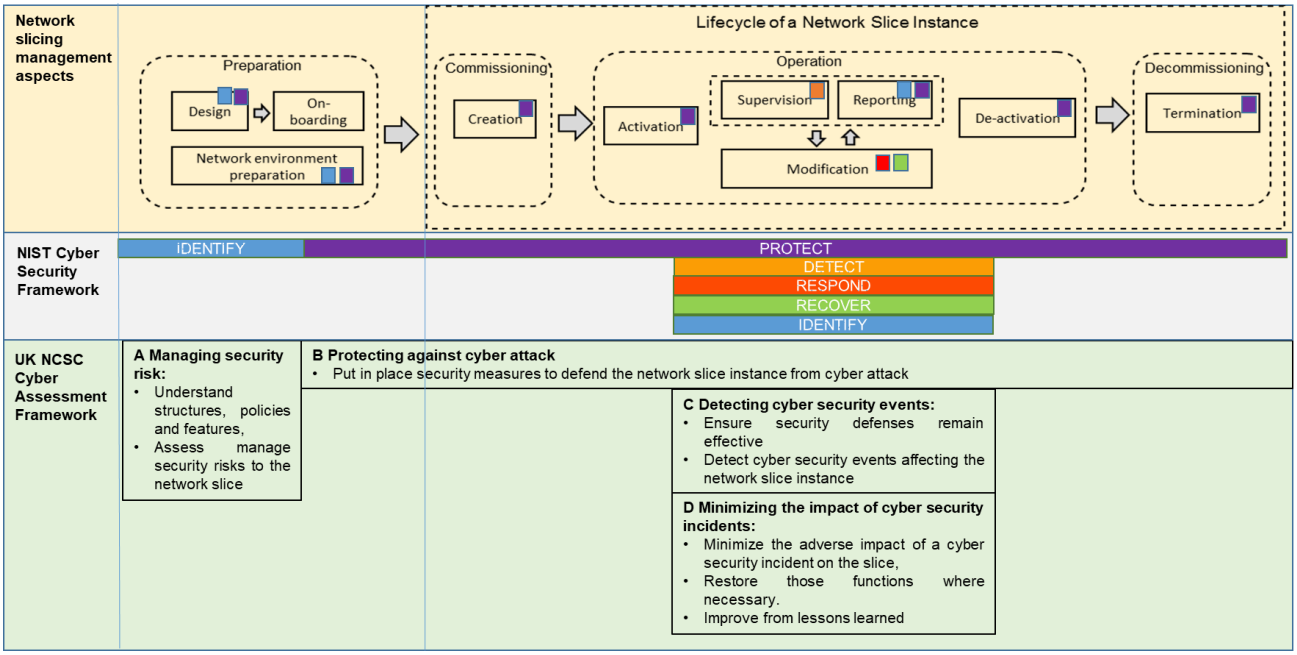

Figure 37. High-level mapping of the Cyber Security Frameworks [82][82] to the Network Slice Lifecycle Management.

Among the security measures in place to protect the network service instance, there is a specific security function focusing on performing security orchestration with a local impact by its control actions. The local behaviour of the security orchestration gives many advantages, in particular a faster reactivity in presence of an incident event, more flexibility to scale to cope with workload variation, better separation of domains that are managed by different security policies.

5.2.3 *LOCAL AUTONOMIC SECURITY MANAGEMENT*

5.2.3.1 DESCRIPTION

The role of the local autonomic security orchestrator (L-ASO) (cf. Figure 38) is to manage the security functions (virtualised or physical function or provided as a NFVI resource) that secure the NFV objects (VNF, PNF, VSF, Physical Security Function (PSF), virtual links, nested NS...) constituting the network service associated with the network slice, while being agnostic of the network slice construction.

These NFV objects are the assets of the 5G system needing to be protected from cyber-attacks, by leveraging the directives and guidelines of cyber security framework. Some security procedures derived from the framework can be automated by means of closed loops to improve response time and accommodate large workloads while being limited in resources and gain in cyber-resilience. To allow the

creation of automation loops, the MonB5G functions MS, AE, DE and its related ACT are used to assist the local security orchestrator in monitoring, data analysis and decision making, and consequently they transform the security orchestrator into an autonomic system called local autonomic security orchestrator.

L-ASO is created as a dedicated or common PaaS offering as defined in ETSI GS NFV-IFA 029 [58] along with the network service instance. We can consider that a PaaS offering is instantiated as VNF instances integrated with the network service or as shared VNF instances. The NFV MANO system ignores the purpose (security, management and network) of VNFs it deploys. Consequently, LCM of L-ASO (VNF placement, automatic scaling) benefits from the same MANO services as any ordinary network function. L-ASO is able to, reserve or free virtualised resources to adapt to the workload of the network service it defends within the scaling limit defined in the VNF deployment flavour).

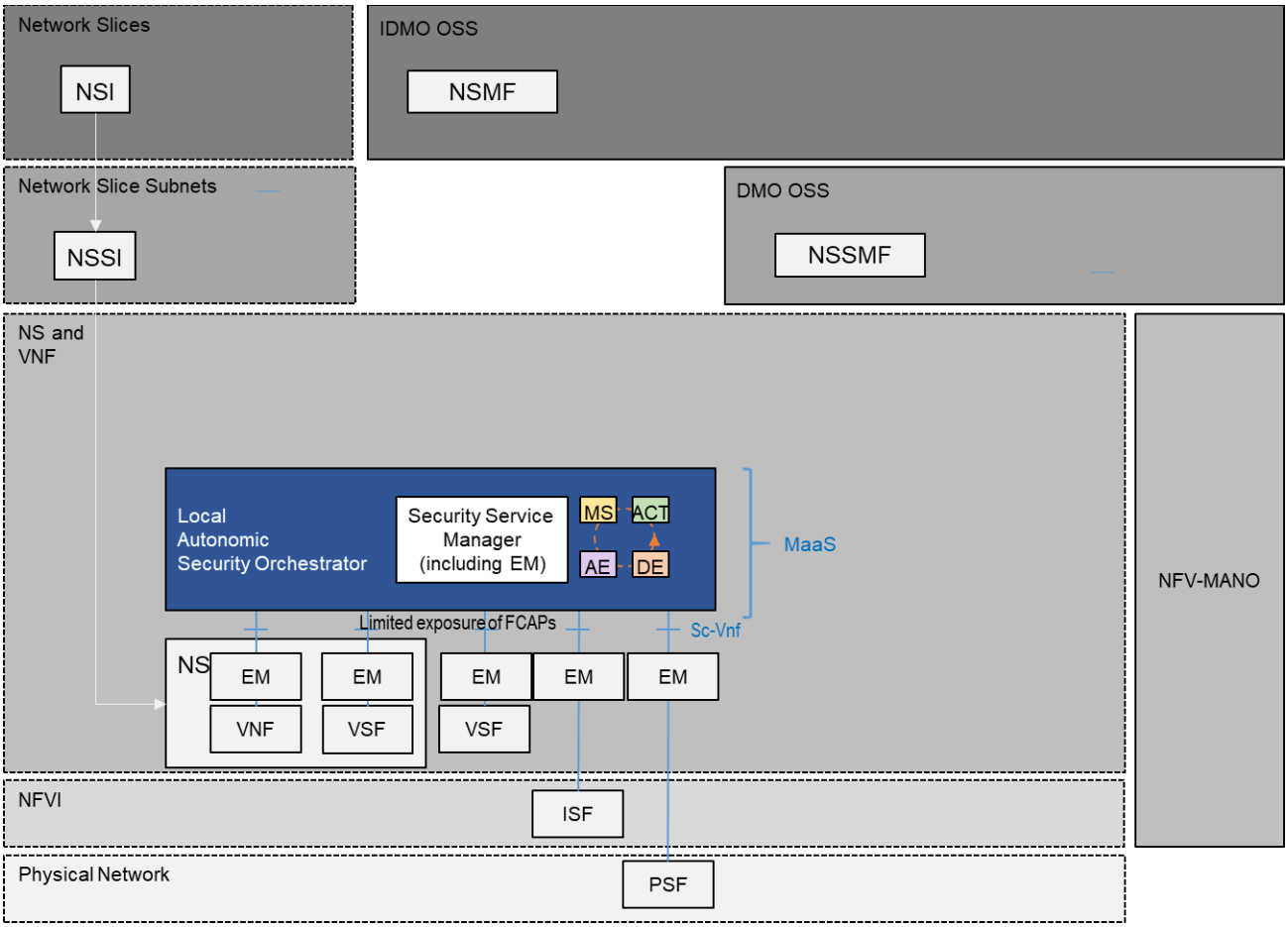

Figure 38. High level architecture of L-ASO managing the intra-slice subnet security.

5.2.3.2 PREPARATION SUBCASE

When preparing the network slice subnet, L-ASO will be associated with the implemented network service as a virtualised management function, being either an integrated or as an auxiliary component attached via a service access point (SAP). As a VNF, the deployment flavour of L-ASO will be derived from the performance-related conditions of the network service promised to the customer with the objective of maintaining L-ASO transparent and efficient at any workload level during the operation stage. Leveraging the MonB5G components MS, AE, DE, ACT, L-ASO is designed to implement security framework recommendations to ensure the protection solution remains effective and to detect potential signs of threat, which are likely to be precursors or indicators of a breach. Figure 39 shows the preparation operation which enhances the security of NSD by including L-ASO among the required security functions.

L-ASO can be either:

- Embedded into the network service of the network slice subnet, in that case, L-ASO is dedicated to managing the security of the Network Slice Subnet Instance (NSSI);
- Referenced as a shared VNF, where it is likely a security manager common to multiple NSSIs.

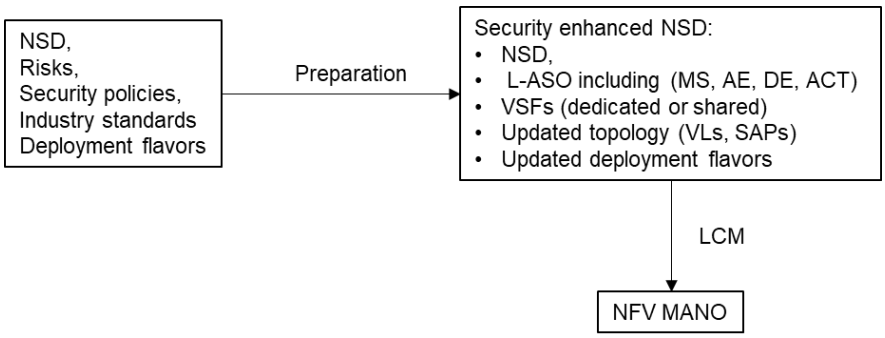

Figure 39. Preparation stage: L-ASO is associated with the network slice subnet NSD as an embedded or shared VNF to manage its internal security.

5.2.3.3 OPERATION SUBCASE

In the previous section 3.5.3, we described how the MonB5G components are involved in managing security policies. Among the configuration parameters provided by OSS, L-ASO receives the security policies to set up the policy engine (PE), the component responsible for generating an action plan in the presence of an incident event. The generated action plan is then executed by the policy administrator (PA), the component in charge of sending commands to relevant policy enforcement points (PEPs) which realise the actions. Mapping to the MonB5G constructs, DE can perform the PE role, ACT can serve the role of PA, and PEPs are Ems (cf. Figure 40).

At the commissioning phase of the network slice subnet, DE (as a PE) is updated with the configuration parameters including the security policies of incident response. As for MS, it is instructed to perform continuous security monitoring of the connection points (network) and functions considered as essential for the network service, while AE receives ML models and algorithms to detect anomalies and known attacks. These configurations may be delivered by OSS via the EM of these components either in direct mode, or indirectly via DE. As a policy administrator, ACT is granted permission to use the services exposed by PEPs via the authentication and authorization procedures, to execute the actions planned by DE.

At the operation phase of the network slice subnet, security monitoring by MS collects asset activity logs, network traffic, and other indicators that provide an overall picture of the cyber security measures in place. MS performs some preliminary processing to reduce the storage volume and facilitate analysis such as extracting high level protocol from raw network data.

Analysing the pre-processed data exposed by MS, AE detects inventoried threats or unusual behaviours leveraging cyber threat intelligence and AI-based techniques. Moreover, AE verifies the effectiveness of the protection, and it tries to discover, in case of a security breach, the most plausible attack hypothesis from the correlation of the received events. When an incident is confirmed from the alerts, DE receives the trigger from AE to apply security policies and plan containment and remediation actions.

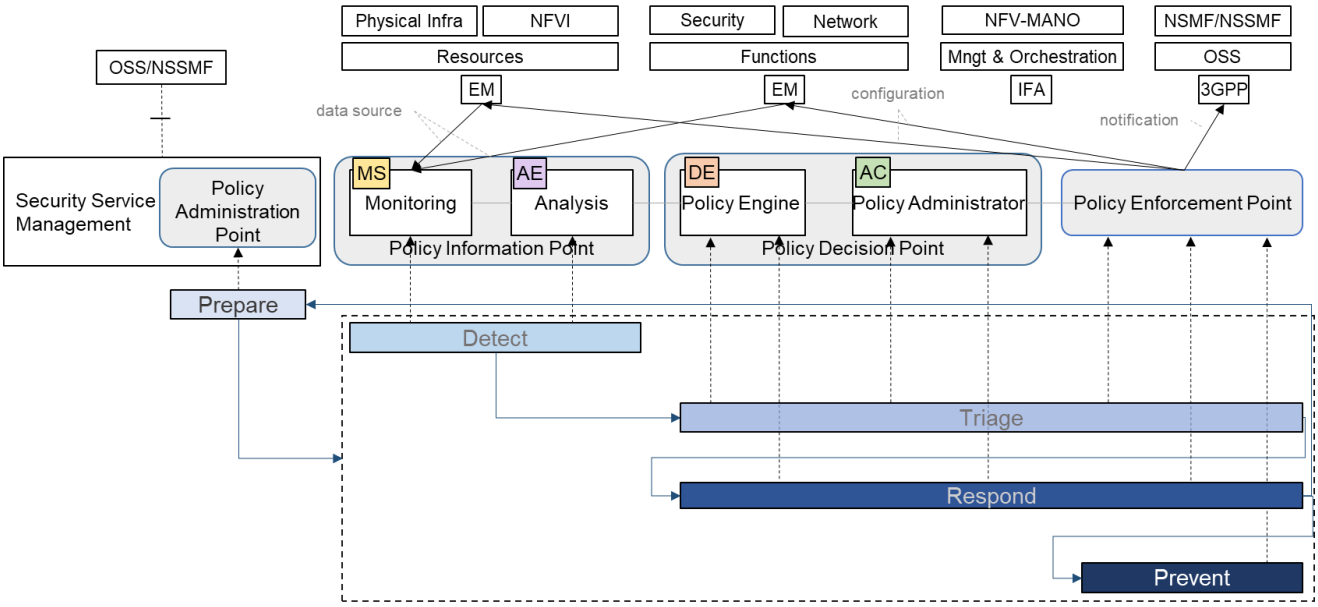

Figure 40. Integration of MonB5G components MS, AE, DE, and ACT in the Incident Management Process of L-ASO and Interaction with the 5G system

5.2.3.4 ALTER ATTACK USE CASE

5.2.3.4.1 USE CASE DESCRIPTION

The aLTEr attack is a Man-in-the-Middle (MitM) attack[33] and is performed between the user equipment (UE) and gNB. Figure 41 gives an overview of the attack scheme. The attacker, as MITM, can relay high-level messages from the user and exploits the fact that integrity protection may be absent for the user data as a vulnerability to perform the attack. Indeed, enabling PDU session[34] user plane protection (integrity protection and/or encryption) is optional in 5G networks, and it depends on the network operator's security policies to find a trade-off between greater protection and higher latency and power consumption.

The attack consists of intercepting the user's Domain Name Server[35] (DNS) query message in the user plane using its MITM position on the radio bearer and replacing the original DNS server address with its own. The message is then redirected to the rogue DNS and the attacker has control over the translation of hostnames into IP addresses.

[33] For further details also see, for example: https://en.wikipedia.org/wiki/Man-in-the-middle_attack.

[34] See, for example: https://www.5gworldpro.com/5g-knowledge/what-is-pdu-session-in-5g.html.

[35] DNS translates domain names to IP addresses so browsers can load Internet resources. Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as, e.g., 192.168.136.237.

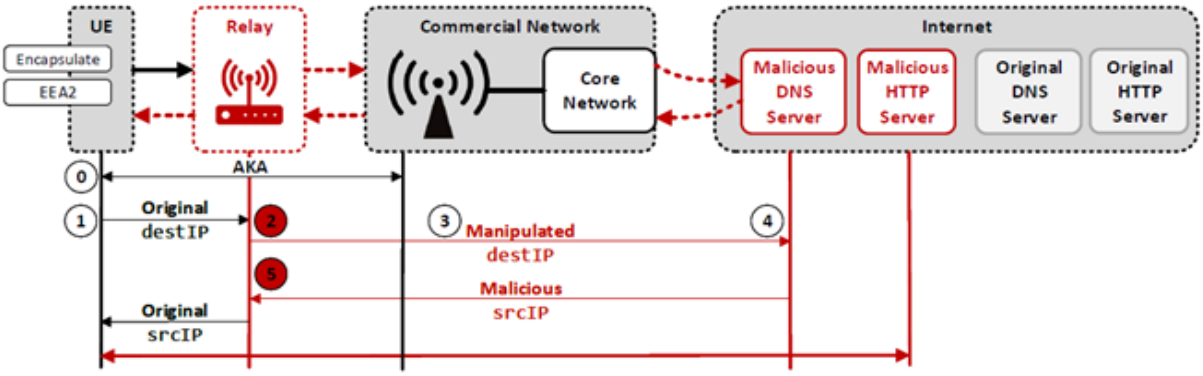

Figure 41. aLTEr: Overview of the DNS redirection attack. We deploy a malicious relay as a MitM between UE and the commercial network and alter the destination IP address of a DNS request to redirect messages to our malicious DNS server [86].

5.2.3.4.2 THE PREPARATION AND PREVENTION PROCESSES

At the initial phase, the monitoring system (MS) collects raw IP data sourced from the router of switch placed at the N6 interface[36], these data are sent or received by UEs. MS translates on the fly the raw captured data into transaction data and extracted content data, in the form of logs summarising the protocols and files seen traversing the N6 interface. The generated logs are streamed to the database for a batch analysis and to an application for a near-real time analysis.

An instance of the Analytical Engine (AE) is configured to perform behavioural analysis of DNS traffic. Many ML techniques can be used, such as the unsupervised learning method based on the Isolation Forest algorithm or the One-Class Support Vector Machine (SVM) semi-supervised learning algorithm[37] to better catch anomalies.

The Decision Engine (DE) as a Policy Engine (PE) receives polices to perform the actions of:

- Triage process to verify the alert event;
- Response process to contain the ongoing malicious activities and prevent the attack from occurring again.

The Actuator (ACT) as a Policy Administrator (PA) is setup to execute the possible actions, which DE may plan. It also receives all necessary credentials (authentication and authorization parameters) to trigger the management API exposed by network functions and security functions (cf. Figure 42).

[36] N6 is the interface between the Data Network (DN) and the User Plane Function (UPF).

[37] Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. Also see, inter alia: https://en.wikipedia.org/wiki/Support-vector_machine.

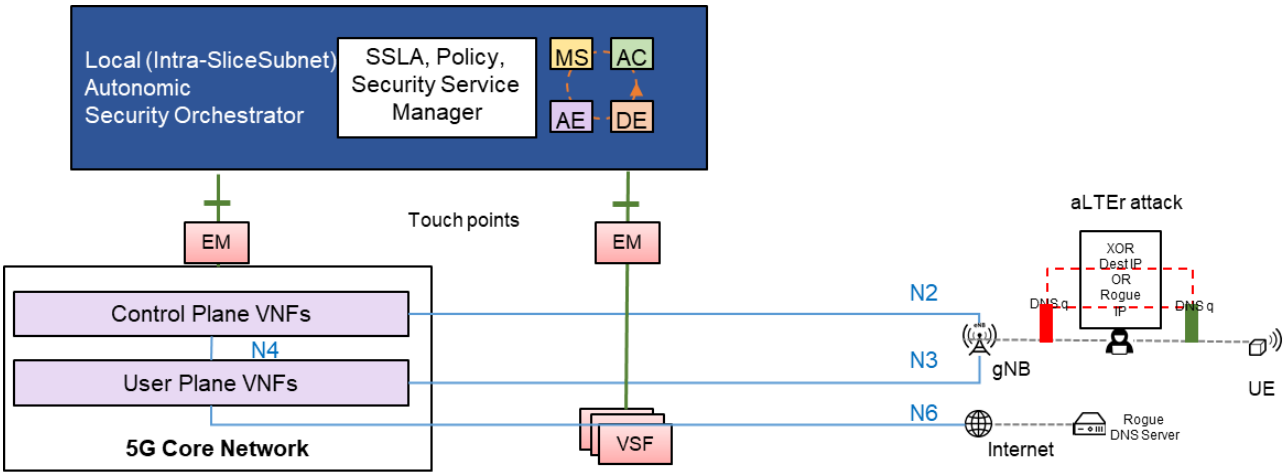

Figure 42. Overview of the safeguard measures deployed to protect the 5G network against the aLTEr attack.

5.2.3.4.3 THE DETECTION PROCESS

The detection process aims at finding anomalies in the DNS transaction data sent by MS. Using the semi-supervised method[38], AE has been trained with data of the normal DNS logs and it analyses the DNS transaction logs streamed by MS to detect anomalies in DNS server address. Figure 43 shows the detect call flows.

5.2.3.4.4 THE TRIAGE PROCESS

The detection process raises an event indicating an unusual DNS server address in the traffic log, where UE is supposed to use the network operator DNS server to translate domain names to IP addresses. This event can be correlated with the aLTEr attack scenario[39] recorded in the Cyber Threat Intelligence (CTI) database[40].

In this scenario, the attacker exploits the lack of user data integrity protection in the radio channel to do an XOR with the well-known DNS address of the network operator in order to replace it with the address of his own DNS server. However, this correlation hypothesis must be verified against the use of a private DNS by UE. The hypothesis can be resolved by updating the default DNS configuration of UE with a new undisclosed and computed DNS address: if the anomaly persists with the same previous address, it means no one is trying to redirect the DNS requests to the network operator's server and UE is just using a private DNS address.

DE receives these rules to verify and change the state of the hypothesis. As the outcome, it triggers an event of a private DNS or an aLTEr attack incident. Figure 43 below shows the interaction between components during the processes of detect and triage.

---

[38] Also see, for example: https://en.wikipedia.org/wiki/Semi-supervised_learning.
[39] For further information also see: https://alter-attack.net/.
[40] Further reading is suggested at: https://www.eccouncil.org/cyber-threat-intelligence/.

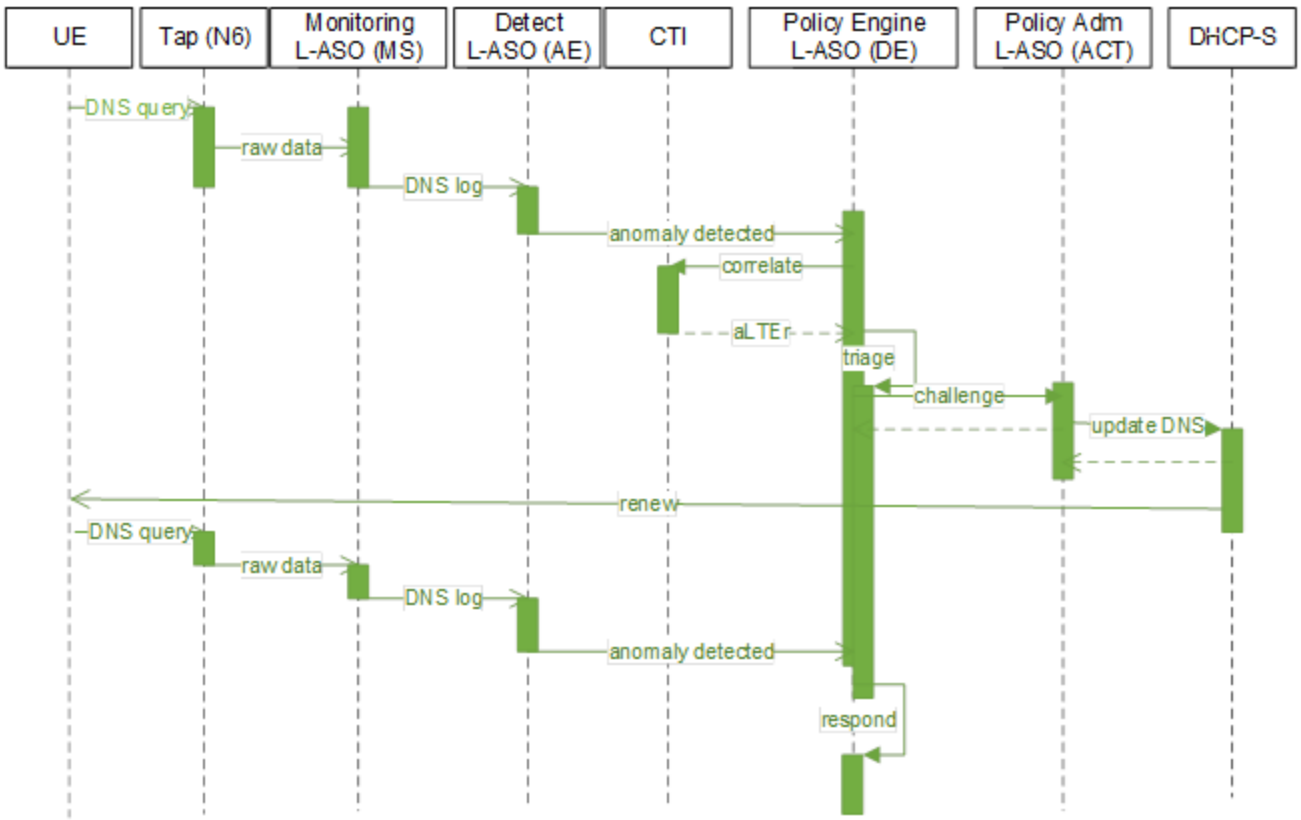

Figure 43. Sequence diagram of detect and triage processes.

5.2.3.4.5 THE RESPONSE PROCESS

In the case of a private DNS utilisation, the network operator may decide to steer the traffic per couple of (UE, DNS server) away from the detection process to avoid any unnecessary processing. Otherwise, the aLTEr attack is confirmed for the connection, and some remediation action plans are possible, such as:

- Change the security policies for that UE in the database enabling the integrity protection for user data bearer and reconnect UE, (Figure 44 shows the respond process call flow);
- Enable DNS over TLS or DNS over HTTPS[41].

To realise these actions, PE (DE) relies on PA (ACT) to trigger the relevant PEP (VNF, VSF) and related services. For the decision plans listed above, the relevant PEPs and related services might be:

- DHCP server[42]:
    - Assign a temporary DNS server address for UE;
    - Send DHCPACK[43] to the client to rebind or renew its lease to receive the updated IP configuration;
    - Enable DNS over TLS/HTTPS;

[41] Also see: https://en.wikipedia.org/wiki/DNS_over_TLS.

[42] A DHCP Server is a network server that automatically provides and assigns IP addresses, default gateways and other network parameters to client devices. It relies on the standard protocol known as Dynamic Host Configuration Protocol or DHCP to respond to broadcast queries by clients.

[43] Also see: http://www.on-time.com/rtos-32-docs/rtip-32/programming-manual/dhcp-server/dhcp-messages.htm.

- SMF[44]:
    - Disconnect and reconnect IMSI[45];
    - Convert UE's IP address to its IMSI;
- SFC:
    - Steer away the traffic (UE's IP, DNS server) from the continuous monitoring;
- UDM/AUSF:
    - Update the PDU session user plane security police to require integrity protection;
- OSS:
    - Via the notification interface, security incidents and response activities are reported to NSSMF.

[44] Also see: ETSI (2018). ETSI White Paper No. 28: "MEC in 5G Networks", June 2018. Available at: https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf.

[45] The International Mobile Subscriber Identity, abbreviated as IMSI, is the internationally standardised unique number to identify a mobile subscriber, as defined in ITU-T Recommendation E.212.

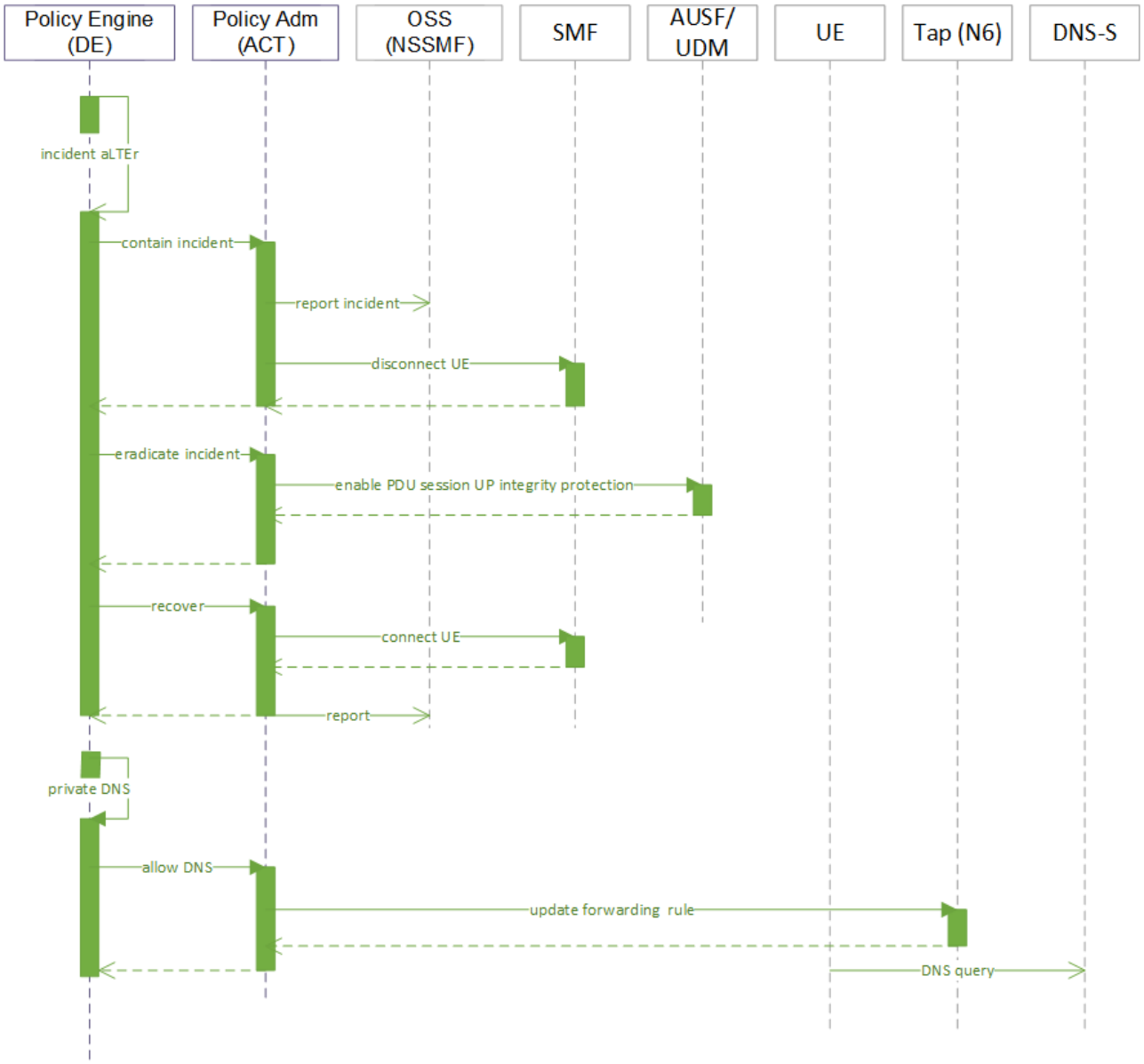

Figure 44. Sequence diagram of the response process in the event of an aLTEr incident and use of private DNS.

In the sequence diagram shown in Figure 44, the response to the impacts of the aLTEr incident consists of disconnecting UE to limit the effect of the attack, then activating the security protection for the PDU session User Plane to prevent the attack from taking place, and finally connecting UE to resume the normal operation. As for the private DNS case, the response process simply stops analysing the DNS traffic from UE to avoid unnecessary processing.

5.2.4 *DOMAIN (INTER-SLICE) AUTONOMIC SECURITY ORCHESTRATION*

5.2.4.1 DESCRIPTION

At the initial phase, the security expert should prepare and onboard the NSD of Domain Autonomic Security Orchestrator (D-ASO) in the NS catalogue of the NFV-MANO system. D-ASO NSD embeds MS, AE, DE, and ACT as VNFs to respectively perform security activity logging, security analysis, policy engine (PE) function, and policy administrator (PA).

The security expert as a security service provider may create one or many instances of D-ASO to manage separately security policies of network slice subnets in accordance with certain criteria, such as for

instance the type of network, industry standards, organisation and technologies to guarantee a better slice security isolation.

D-ASO manages the security including protection and incident response of one or more NSST and their derived NSSIs.

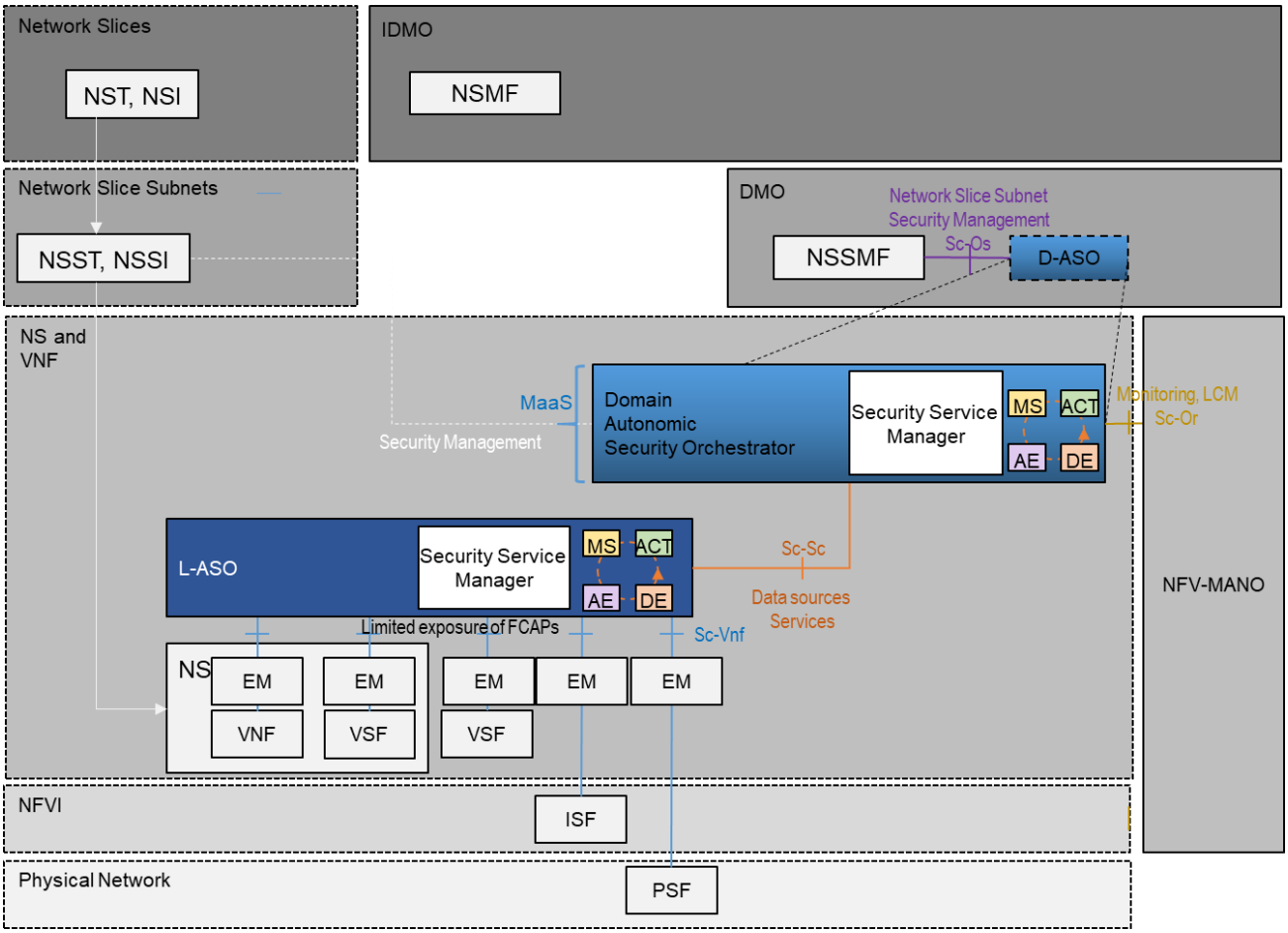

Figure 45. High level architecture of D-ASO.

5.2.4.2 NETWORK SLICE SUBNET PREPARATION SUBCASE

At the domain management layer, the assets the domain autonomic security orchestrator (D-ASO) (cf. Figure 45) shall consider are the network slice subnets and its constituent network services for their lifetime. In the preparation stage of the network slice subnet, D-ASO identifies the security needs that are derived from:

- The customer's input expressions in terms of security SLA (SSLA) or security intents;
- The industry-specific standards of the vertical;
- The structure of the baseline network service descriptor (NSD) referenced by the network slice subnet template (NSST);
- The threat, vulnerability and risk assessments[46] (TVRA);
- The NSD instantiation parameters to size the service according to the promised performance.

Based on the security needs, D-ASO selects the appropriate protection solution according to the risk mitigation strategies defined by the security experts of the network slice provider, updates the baseline

[46] A Threat, Risk and Vulnerability Assessment (TRVA) considers the client's need to protect people and assets, minimise exposure to crime and terrorism, breaches of security and overall business risk.

NSD with the available and relevant security functions as well as the security virtual links, including new sizing parameters. Among the security functions included, D-ASO can insert according to its security policies a security management function L-ASO to which it delegates the management of the internal security of the network slice subnet (cf. Figure 46). In the feedback direction, D-ASO subscribes to L-ASO's notification service to receive its activity reports. D-ASO analyses all received data to assess whether the safeguard measures implemented in the network slice subnet require any escalation of their security policies and enforcement points. Any security policy update might be generalized and applied to other network slice subnets sharing the same security policies.

As a result, D-ASO produces:

- The security enhanced NSST, called se-NSST;
- NSD is extended to accommodate security elements including a L-ASO and new network topology;
- The new instantiation parameters may contain specific requirements for NFVO to deploy the security functions in a trusted execution environment[47] (TEE);
- The security policies to be configured in NSSI.

Then se-NSST is ready and will be instantiated as a security enhanced network slice subnet instance (se-NSSI) by NSSMF using the domain orchestrator NFVO.

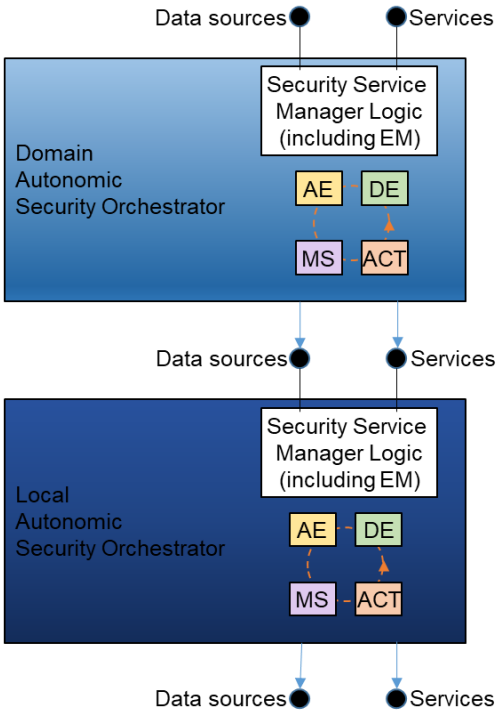

Figure 46. Generic interfaces between D-ASO and L-ASO.

5.2.4.3 NETWORK SLICE SUBNET OPERATION SUBCASE

At the commissioning phase of the network slice subnet, D-ASO updates L-ASO via the management services with the security configuration parameters, which include at least:

- The security policies for DE to manage incident response;

[47] A trusted execution environment (TEE) is a secure area of a main processor. It guarantees code and data loaded inside to be protected with respect to confidentiality and integrity. TEE as an isolated execution environment provides security features such as isolated execution, integrity of applications executing with TEE, along with confidentiality of their assets. Also see: https://en.wikipedia.org/wiki/Trusted_execution_environment.

- The locations where MS to perform continuous monitoring;
- The trained ML models for AE to detect anomalies and known attacks;
- The permissions for ACT to use the services exposed by PEPs via the authentication and authorisation procedures, and to execute the action plan issued by DE.

The deployment flavour of L-ASO VSF is computed by D-ASO during the preparation phase, which ensures that the security measures are sufficiently scalable to meet the workload of the network service.

At the domain scope, MS allows D-ASO to monitor network slice subnet activities from:

- The NFV MANO system using ETSI GS NFV-IFA 033 [61] interfaces to collect indicators and telemetries;
- The NSSMF to analyse the potential impacts on the protection caused by a management operation;
- Data source interface of L-ASO instances to collect security information (reports, indicators, etc.).

By analysing the activity logs, AE provides to D-ASO feedback information on the security status of the protected network slice subnets. Moreover, a local incident response activity reported by an L-ASO may need response actions at the network slice subnet level (escalation), as because of its impact is beyond NSSI. For example, in response to an incident occurring inside a NSSI-A, D-ASO may decide:

- To change the configuration parameters and manage the lifecycle of the NS and VNFs of NSSI-A;
- To change the configuration parameters and manage the lifecycle of the NS and VNFs of another NSSI-B that is exposed to the same risk;
- To update the design of the template NSST so all future instances derived from the model will benefit from the security improvement.

The global knowledge over multiple instances of network slice subnets is an advantage in learning from complex attacks and improving the protection process, the detection process and the response process. According to the cyber security framework [82][82][83], AE can carry out some of the recommendations of the UK NCSC to extract more intelligence by learning the lessons from:

- The incident itself:
    - To improve the protection to prevent the incident;
    - To identify other information that would help to detect earlier;
    - To search for vulnerabilities that can only be discovered through multiple incidents.
- The response:
    - To verify the successful and effectiveness of the response actions;
    - To identify the missing information that could be useful.

Based on the security policies in place, DE plans response actions that could have broader impacts ranging from the NSSI under attack to the NSSIs being exposed to the same risks and even to the template containing the defence structure design. For instance, these response actions may be:

- Updating the L-ASO configuration and policies;
- Managing the NSSI lifecycle;
- Modifying the design of NSST;
- Reporting the incident to NSSMF for correlation with possible loss or disruption of business services.

In order to execute the actions decided by DE, ACT as a policy administrator (PA) would be provided with the necessary information to discover and be granted permission to use the limited services exposed by the relevant PEPs. These PEPs are:

- L-ASO exposing its management service through its element manager (EM);
- NSSMF exposing limited capabilities to manage the lifecycle of network slice subnet;
- The NFV MANO system exposing some of the ETSI GS NFV-IFA 033 [61] interfaces.

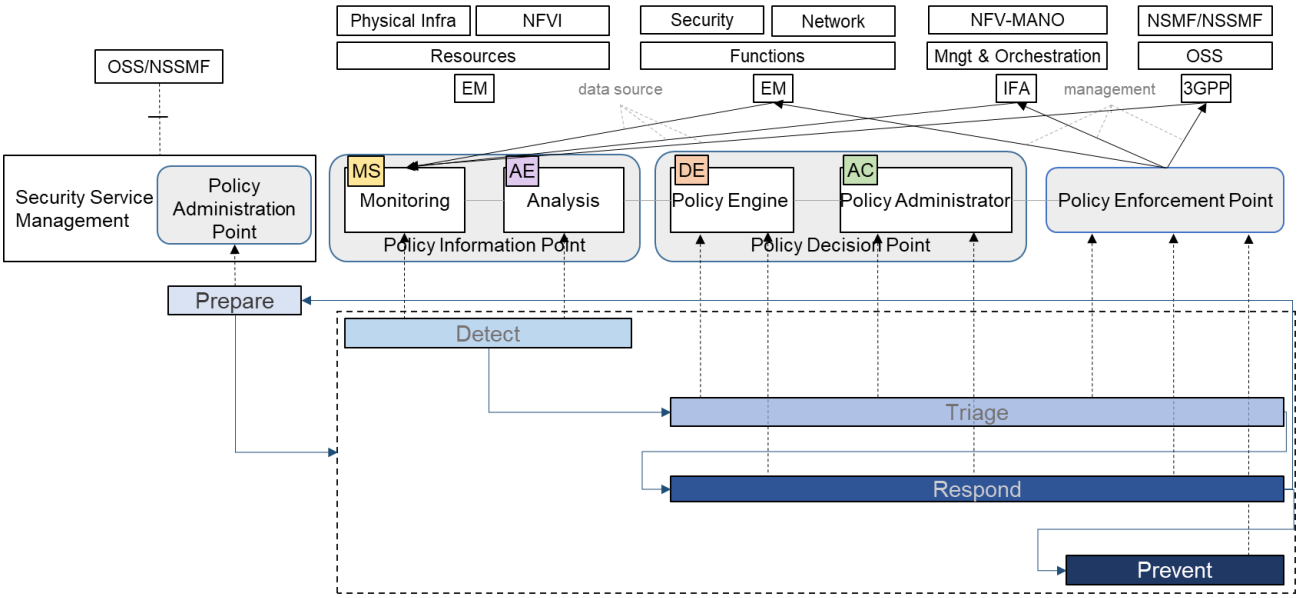

Figure 47. Integration of MonB5G components MS, AE, DE, and ACT in the security management processes of D-ASO and Interaction with the 5G system.

As presented in Figure 47, PIP can collect and analyse reports from L-ASO and indicators from NFV-MANO and NSSMF, while PDP can execute management services exposed by L-ASO, NFV-MANO and NSSMF.

5.2.4.4 CONTINUATION OF THE ALTER USE CASE AT THE DOMAIN LEVEL

As a mitigation to the aLTEr attack, the security policy shall require the integrity protection of PDU session UP (cf. Figure 48). However, this feature may not be supported in some cases, such as the non-standalone 5G network, or when gNB or UE does not support it. Several alternatives exist to reduce the impact of the response on SLA, either L-ASO instructs the Session Management Function (SMF) to block the UE communication to contain the threat or D-ASO deploys a security function (firewall) to deny such bad DNS traffic.

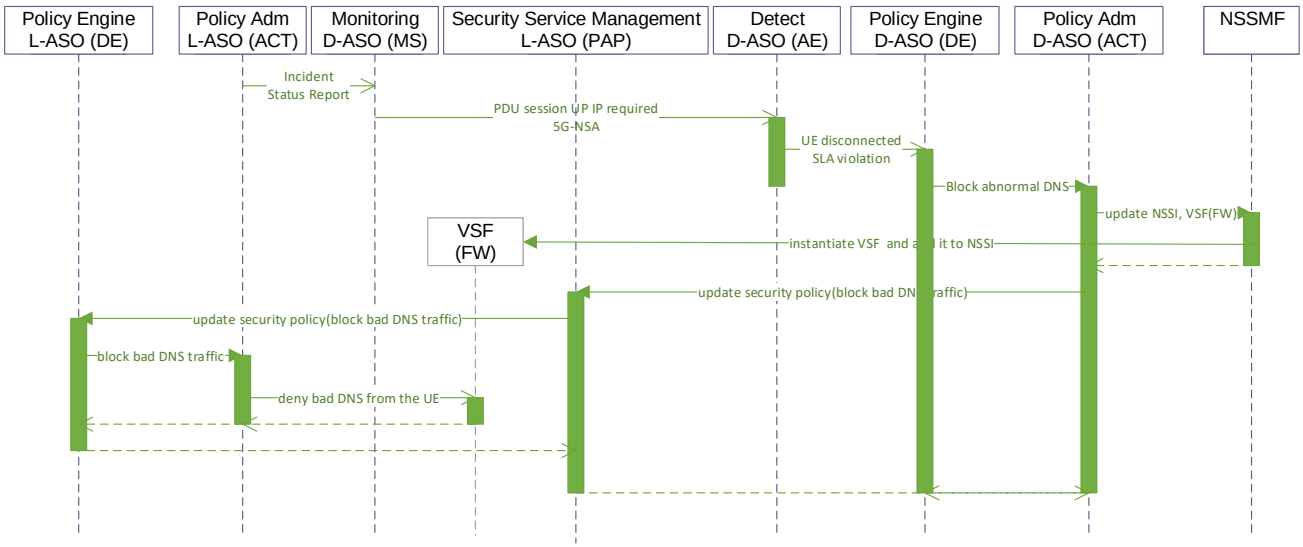

Figure 48. Instantiation sequence diagram of a new VSF by D-ASO to improve the response of L-ASO which consists of denying only the bad DNS traffic instead of dropping the UE connection.

In addition, D-ASO can generalize the incident mitigation solution from a NSSI to any other instances that are exposed to the same risk as a prevention action.

5.2.5 *END-TO-END AUTONOMIC SECURITY ORCHESTRATION*

5.2.5.1 DESCRIPTION

The E2E security of the network slice provided to the customer shall be ensured by the Inter-Domain or End-to-End Autonomic Security Orchestrator (E2E-ASO) regardless to the number of network slice subnets the service has crossed. As it is the case with D-ASO, E2E-ASO must be initially prepared and onboarded by the security expert as an NSD in the domain he/she has trust. E2E-ASO NSD embeds MS, AE, DE, and ACT as VNFs to respectively perform security activity logging from domains, security analysis, policy engine (PE) function, and policy administrator (PA).

E2E-ASO (cf. Figure 49) has the same modular VNF-based architecture as D-ASO to provide security management services. For a better slice security isolation, multiple instances of E2E-ASO can be created in order to separate slice security data and activities for different customers with respect to criteria such as organization or industry standards. Moreover, E2E-ASO is naturally scalable to cope with a large number of slices thanks to the VNF construct.

D-ASO exposes limited data source and management services to E2E-ASO, such as NSSI security activity reports and security policy administration services. E2E-ASO analyses these reports to extract new insights, then makes new decisions to improve the security of E2E services.

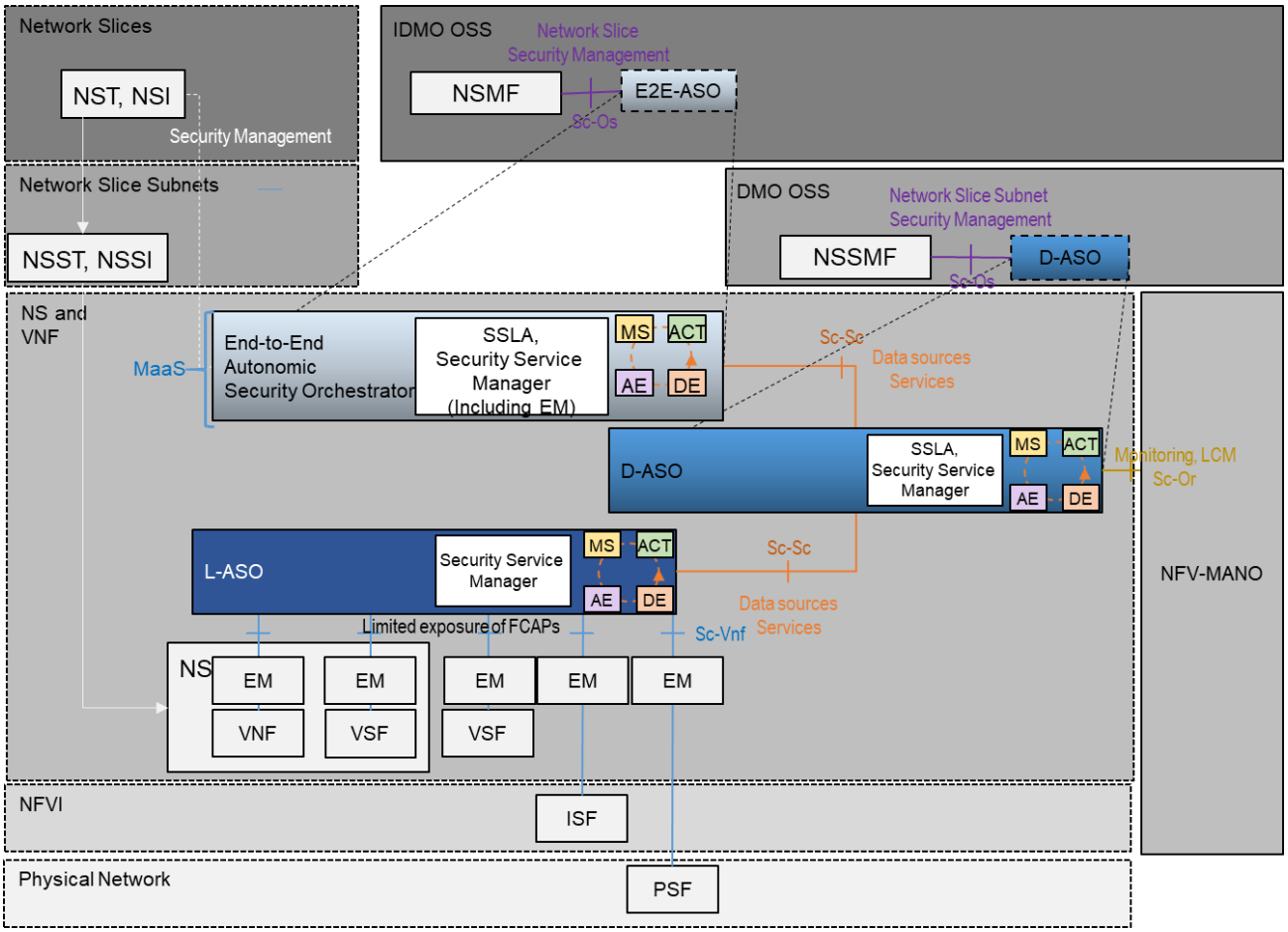

Figure 49. High level architecture of E2E-ASO managing the slice security.

5.2.5.2 NETWORK SLICE PREPARATION SUBCASE

At the inter-domain management layer, the assets E2E-ASO shall consider are network slices and their constituent network slice subnets for their lifetime. In the preparation stage of the network slice, E2E-ASO identifies the security needs that are derived from:

- the customer's input expressions in terms of security SLA (SSLA) or security intents;
- the industry-specific standards;
- the structure of the baseline network service descriptor (NSD) referenced by the network slice subnet template (NSST);
- the threat, vulnerability and risk assessments (TVRA);
- the promised performances.

Based on the security needs, E2E-ASO selects the appropriate protection solution according to the risk mitigation strategies defined by the security experts of the network slice provider, it derives the security requirements for NSSMF D-ASO which provide the composite network slice subnets.

In the feedback direction, E2E-ASO subscribes to notifications from NSSMF D-ASO to receive its activity reports. E2E-ASO analyses all received data to assess whether the security requirements are fulfilled by each network slice subnet or whether they need to be modified. Any security policy update might be generalized to the other network slices managed by E2E-ASO as it has an impact on the security of not only the current instance but also all instances sharing the same security policies.

As a result, E2E-ASO produces security enhanced NST, called se-NST composed of se-NSSTs, each of them fulfils the security requirements sent by NSMF to NSSMF of the crossed-domain.

The se-NST is ready and will be instantiated as a security enhanced network slice instance (se-NSI) by NSMF delegating the lifecycle management of the security enhanced network slice composite subnets to NSSMF D-ASO using the exposed management services. The generic interfaces between E2E-ASO and D-ASO are depicted in Figure 50.

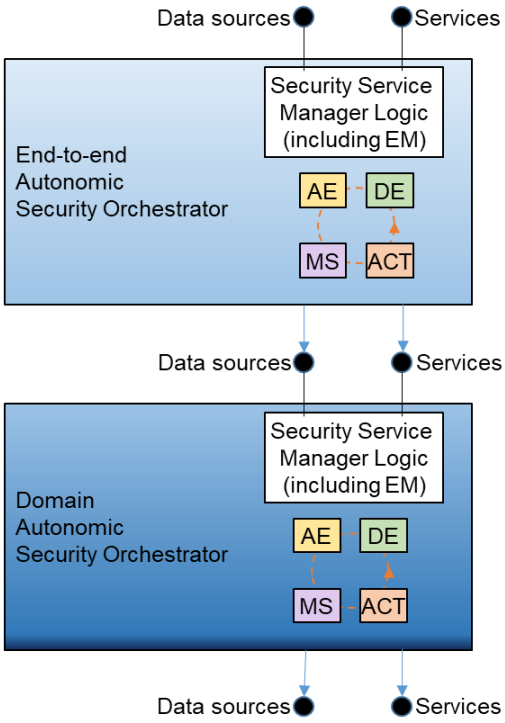

Figure 50. Generic interfaces between E2E-ASO and D-ASO.

5.2.5.3 NETWORK SLICE OPERATION SUBCASE

At the commissioning phase of the network slice, E2E-ASO derives the security requirements for each domain that supports the E2E network slice for NSMF. NSMF sends to NSSMF of the support domain a request including network slice subnet related requirements and the security needs.

Using a MS instance, E2E-ASO monitors network slice subnet security activities by subscribing to notifications enabled by:

- NSMF, to analyse the potential impacts on the protection caused by a management operation;
- the cross-domain NSSMF, to collect network slice subnet security activities via indicators or reports.

By analysing the activity logs, AE provides feedback to E2E-ASO on the security status of NSI it protects, a security incident reported sent by a NSSMF can lead to an escalation because of its impact. For example, the security incident could affect:

- NSSI that needs an update of the security requirements;
- Another NSSI which can improve security upstream of the current NSSI;
- Another NSSI which is exposed to the similar risk;
- NSI which needs an update of the E2E security requirements;
- The security feature listed in NST.

Through architectural similarities, E2E-ASO, like D-ASO, can leverage AE to extract from incidents that have occurred on multiple managed instances to identify new risks, verify alignment with security requirements, and improve the protection rendered.

According to the security policies DE has received, it plans response actions that could have impacts on NSI that has been under attack, the other NSIs that shares security policies, or the se-NST that contains the security requirements. For instance, these response actions may be:

- updating the requirements related to NSSI;
- managing the NSI lifecycle;
- modifying the design of NST;
- reporting the incident to NSMF for correlation with possible loss or disruption of business services.

In order to execute the actions decided by DE, ACT as a policy administrator (PA) would be provided with the necessary information by the domain security factory to discover and be granted permission to use the limited services exposed by PEPs. These PEP are:

- NSSMF exposing its management API through its standard 3GPP API (3GPP TS 28.531 [87]/3GPP TS 28.541 [88]);
- NSMF exposing limited capabilities to manage the lifecycle of NST and NSI.

5.2.5.4 THE ALTER USE CASE AT THE INTER-DOMAIN LEVEL

Subsequently, as shown in Figure 51,the aLTEr incident indicates that the PDU session UP is exposed to a higher risk of MitITM attack than initially expected when integrity protection is not enabled. Therefore E2E-ASO seeks to strengthen the protection of communication services for critical devices in other managed network slices by increasing the level of security required. As a result, E2E-ASO lists NSSIs realizing the 5GC network and requests the relevant D-ASO to enable the integrity protection option.

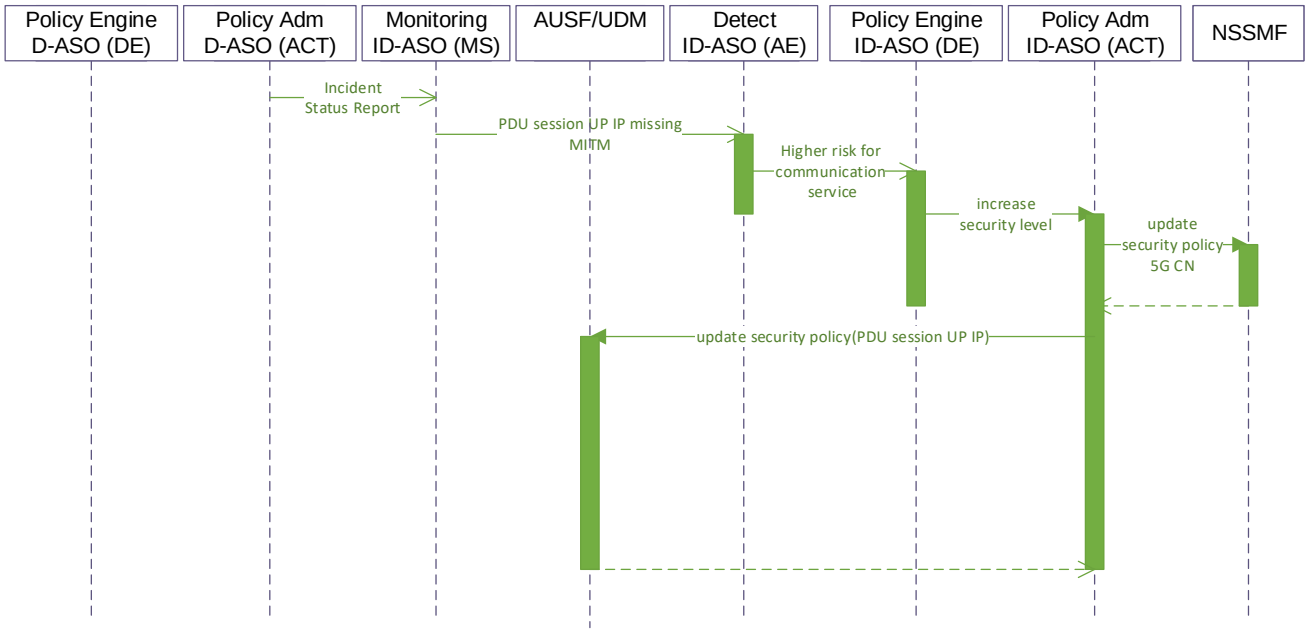

Figure 51. Sequence diagram of lessons learned from the incident in one NSI to improve the protection of the other NSIs.

6 Remarks on the MonB5G architecture

6.1 MonB5G architecture vs. ETSI ZSM requirements

The MonB5G architecture presented in section 3 is distributed, AI-driven and programmable. The mentioned features make the proposed approach scalable and flexible, with a goal towards appropriately supporting zero-touch management of network slices and services. The ETSI ZSM group has already defined an extensive list of requirements for zero-touch management, containing over 170 topics related to autonomic management [55]. In this section we outline some key requirements and point out how they are fulfilled by the MonB5G architecture. While the ETSI ZSM requirements list is flat, without any grouping, we have grouped a key set of requirements under different categories related to the specific aspects of autonomic service management. The categories include:

- monitoring and data analytics, i.e., requirements related to the collection of the performance data, their aggregation and ways of data usage to fuel analytic engines (AE);
- management actions, i.e., requirements related to network maintenance, coordination of management, recovery actions, etc.;
- management operations, i.e., requirements that relate to access to network slicing management services, LCM, management data policies, etc.;
- control loops, i.e., requirements devoted to the operation of control loops, disabling the control loops in terms of faulty operation, etc.;
- other important requirements not belonging to previous categories, e.g., requirements related to security, etc.

Table 6 presents these categories, with the respective functionalities/characteristics, and comments on how they are supported by the MonB5G architecture. Please note that these functionalities and characteristics are aligned with the goals we set out in section 3.1. For example, we are looking at key characteristics such as:

- collection of data at different granularities to support closed loops at different levels in the architecture;
- ability to continuously monitor resources, including. running slices, to detect and/or predict changes in behaviour;
- automated closed loops for optimisation and repair during runtime, with decisions based on the analytics results;
- programmable management functionalities;
- E2E management and orchestration across different technological domains;
- security capabilities when delivering automated network and service management.

The functionalities and characteristics presented in Table 6 are not exhaustive. One very important goal for MonB5G is to provide a scalable solution for zero-touch slice management and orchestration to facilitate the deployment of a massive number of slices in different administrative and technological domains. We are supporting this in our architecture through a distributed approach, which ensures automatic local decisions and minimisation of data transmitted across the network. More details on scalability are given in section 6.2.

Table 6. Key ETSI ZSM requirements supported by MonB5G.

Category	ZSM Requirements fulfilled by MonB5G	MonB5G support
Monitoring and Data Analytics Functionalities	Collecting performance data and fault data for a network instance at different granularities and aggregation of VNF/PNF raw measurements to calculate, e.g., slice-level KPIs	Hierarchical control loops and sets of MS, AE, DE, ACT components that perform optimisation and management-related tasks on each level. The performed operations concern both service-related events (SML, IDSM) as well as resource utilization (DMO, IDMO, IDM, IOMF).
	Performing data analytics for predicting KPI changes and failure conditions	
	Storage of historical data needed for the prediction and its exposure to the analytics	
	KPIs measurement	
	Analysis of the collected data to detect the undesired states and derive the root cause. The past, current and future states of the managed entities can be modelled to help to detect the undesired states and move the state of managed entities to the desired state	
	Prediction of the growth or reduction of traffic volume for managed resources for a Customer-Facing Service[48] (CFS) and over a given time period	
	The capability to predictively detect abnormal behaviours of the managed networks and services	
	The ability to demand forecast for capacity planning	
	Monitoring of managed services (network as a service including network slicing as a service) originating from different network/infrastructure domains including but not limited to NFVI, IP/SDN networks, fronthaul, and Radio	
	The capability to analyse conditions to detect root causes	
Management Actions	The capability of taking actions to perform predictive maintenance of a network instance	MonB5G management & orchestration for different administrative and technological domains (RAN, CN, Edge, etc.) by introducing abstractions and entities enabling concatenation of slices deployed in different domains (IDMO, IDSM) into an E2E slice. Also, a "MonB5G Operator" is supported to intervene in case of critical failures and to perform recovery actions.
	The management coordination across different technical domains, including at least Core Network domain, RAN network domain, transport network domain and virtualisation part to support network slicing management	
	The capability to perform recovery actions based on KPIs of the managed networks and services	
	The capability of zero-touch, E2E management and orchestration of 5G networks and services covering network slicing and edge computing	
	Managing the complete lifecycle of the network services/capabilities exposed per management domain, and shall provide an interface that hides internal details (such as the resource layer)	
Management Operations	Access to network slicing management services exposed by the framework for authorized vertical industry customers	Access to the MonB5G framework capabilities is provided via portal
	The capability to provide the interfaces' exposures for the automated management of the services	

[48] Customer facing services represent the commercial view of the services a legal entity (i.e., a network operator) exposes to its customers (a service represents the way that a product is realised and delivered to a customer). The same customer facing service can be used to fulfil different but similar product offers.

Category	ZSM Requirements fulfilled by MonB5G	MonB5G support
	Data availability inside management domains and outside of them so that it can be exposed to any authorized consumer within the framework belonging to one operator	interfaces described in section 3.6.
	The capability of automatic installation of management software	
	Automatic configuration of management software parameters	
	Automated detection of management services offered by a management domain	
	Automated lifecycle management of the framework functional components	
Control Loops Support	Capability to allow different sets of collected data to be used in different closed loops inside a domain and cross-domain	Control loops defined on VNF-, slice-, domain- and inter-domain level. The slice level (SML Sublayer) enables detection & conflict resolution between different loops. Each SML sublayer provides the interface enabling modification of properties of each separate sublayer by the Slice Manager. Fully manual management in case of the control-loops malfunction is supported.
	Detection and conflict resolution between different closed loops inside a domain and in different domains	
	Nested closed loops	
	Reconfiguration of any domain services as required, e.g., in support of closed-loop assurance	
	The use of automated decision loops, with different characteristics and scope, as a means to perform network and service management	
	Provision of an interface for the purpose of bringing decision criteria to the decision loops, i.e., triggers, policies	
	The collection of all available, relevant data and contextual information for a specific decision loop	
	The ability of the network owner to disable any automation function in case of malfunction	
Other functionalities	Scaling of network slice instances within available network resources	Supported by IDM, SML, AI-driven AEs and the security orchestrator
	Configuration of network slice instances during runtime without disruption	
	Status monitoring of all the network slice instances and identification of the network slice instances causing high utilization of network resource(s)	
	Automatic performance of FCAPS management for compute, storage and network resources, NFs, slices and services	
	Automatic configuration of physical and virtualised network function parameters	
	Taking decisions regarding actions to take, time of their execution, and the execution itself based on the analytics results	
	Security capabilities when delivering automated network and service management	

6.2 Scalability of the MonB5G architecture

6.2.1 SCALABILITY CONCEPT AND DEFINITIONS

As the Universal Scalability Law (USL) [89] implies, managing a massive number of parallel slices by only increasing the processing resources is not a guarantee for scalability since it also increases the complexity of their management, the degree of contention in the system, as well as suffering from a lack of collaboration between distributed decision entities. As presented in Figure 52, a scalable architecture would therefore achieve a trade-off in:

- utilisation of shared resources to minimise contention but also to avoid complex management of unnecessary resources;
- information flow by sharing only compressed parameters instead of raw data;
- degree of collaboration by enabling the exchange of inferences between decentralised analytics/decision engines while avoiding that they fall in competitive or too cooperative situations.

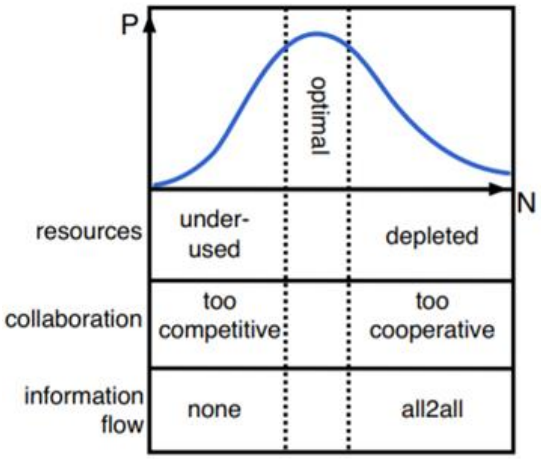

Figure 52. Scalability trade-offs [90].

Therefore, MonB5G targets at least a *linear scalability*, where the network is capable of processing a high number of concurrent slices without sacrificing neither lifecycle management KPIs nor AI performance. This means that by increasing the network resources, the architecture is able to handle a higher number of slices while LCM/AI KPIs do not degrade in a superlinear way (e.g., exponential).

From a design viewpoint, the architecture fulfils the above trade-offs since it presents the following features:

- **Hierarchical architecture per technological domain**: Intuitively, in flat architectures, a considerable increase of admitted slices would lead to performance degradation since a high number of peer-to-peer message flows or AI-based inference exchange occurs between the different distributed local management entities. Similarly, in purely central architectures, a large amount of raw monitoring data stemming from a high number of parallel slices has to traverse several technological domains to be analysed and used in the central decision. This challenges transport and processing queues and induces contention and large delay/overhead. In contrast, MonB5G defines a fine-grained distributed yet hierarchical slice management architecture, where analysis and decision are placed close to the monitored resources, while also considering an upper level for escalation (slice-level, domain-level or IDMO-level) wherein a certain LCM coordination, inference aggregation or large scope policy takes the lead based only on compressed parameters sent by local entities. In particular, the architecture avoids managing micro-slices by crossing different (unnecessary) domains thereby allowing a fast and low footprint local scope.
- **MS/AE/DE defined per slice,** which means less contention compared to centralised approaches.
- **Decentralised and collaborative analysis/decision**, where the scope of collaboration is limited to a subset of AEs/DEs (selection policy, per slice type or per domain).
- **Compressed models sharing** (trade-off between raw data sharing and no data sharing).

6.2.2 *MONB5G ARCHITECTURAL FEATURES THAT CONTRIBUTE TO SCALABILITY*

The main feature of the MonB5G architecture is the scalability of slice management and orchestration. In section 2.1 of the deliverable, the critique of the existing ETSI MANO and 3GPP approach to network slicing has been provided. In this section, we will outline key features of MonB5G that contribute to the scalability of the architecture. Please note that implementation-specific procedures and algorithms that

are not described here also have an impact on the scalability of the MonB5G approach. In general, MonB5G uses the AI-driven distributed approach that naturally contributes to the architecture. It is worth mentioning that MonB5G "has its price". In most cases, we are reducing the volume of the management data exchange between the nodes, but we are increasing the number of resources needed for the implementation of complex MonB5G algorithms (computing and storage). This approach is, however, justified by the more flexible scalability of the computing and storage resources rather than connectivity. Below given we present a discussion on the MonB5G mechanisms that contribute to scalability. It has to be noted that a quantitative evaluation of the MonB5G scalability is impossible to achieve, due to:

- the plethora of different slice templates with a huge variance of their complexity (number of nodes, topology, node's complexity);
- the number of management events is dependent on slice template and external events that are impossible to predict (users' mobility, etc.). Handling of such events is dependent on management procedures which in most cases are not standardised;
- the infrastructure resources (datacentres) can be used differently for each slice deployment. In one case, a single datacentre can be used, while in other cases, multiple data centres (and/or orchestrators) are used.

It is worth mentioning whereas the performance of user or control planes has been monitored for decades, the performance of the management plane has not been addressed so far explicitly.

6.2.2.1 RUNTIME MANAGEMENT SCALABILITY OF MONB5G

In the devised MonB5G architecture, several implemented mechanisms have improved runtime management in comparison to ETSI MANO:

- The exchange of the monitoring data between the nodes/functions and the management system has been reduced by the use of the embedded management of nodes and slices. In both cases, the producer and the consumer of a significant part of the monitoring data can be the same, i.e., node/function or a slice. MonB5G MS improves the scalability of slice management and orchestration by reducing:
    - Distance between producer and consumer of monitoring services:
        - Instead of EM-> OSS; VNF(EM) -> VNF(EM) consumer is local in case of EEM;
        - Instead of EM-> OSS; VNF (EM) -> VNF (MS) – NFVO can be co-located on both nodes.
    - Amount of transmitted data:
        - Monitoring: instead of EM-> OSS; VNF(MS) -> VNF (AE) – placement can be dynamically optimized;
        - Actuating: Instead of OSS-> EM; VNF (DE) -> VNF (ACT);
        - Monitoring: Instead of EM -> OSS, VNF (KPIs, SM) -> VNF (IDSM, Tenant Portal) Actuating.
- The use of AE deployed as a part of a slice contributes significantly to the reduction of the monitoring information processed by other management entities. Such reduction can be in order of 1000.
- In the ISM concept, the management of each slice is a part of the slice and therefore, it is implemented as a set of VNFs. That allows dynamic resource allocation to the management functions (i.e., VNFs) and that way improving that performance if needed. In the ETSI NFV MANO framework, such functions are implemented in the centralised OSS/BSS, not as VNFs but as management platform components.
- ISM is implemented as part of each slices template and the management information exchange between ISM and DMO/IDMO is rather marginal. It impacts a nearly linear increase of the management system load as a function of the number of slices.
- ISM provides a native management interface to slice tenants instead of an indirect, centralised one provided by ETSI MANO, which uses the central OSS/BSS publish/subscribe mechanism. Due to the proposed mechanism, the slice management is more efficient and involves tenants (or Slice

Management Providers) in the human-operated part of the management. Such involvement is possible due to the use of AI-driven management, intent-based.

- Due to much better separation of concerns and the management planes of slices, the number of nodes that has to handle bursts of requests coming from different sources is significantly limited and the probability of the congestion of some management nodes/functions is therefore also limited.
- Most of the slice reconfigurations are done by DEs that are part of the slice ISM. Therefore, the delay in taking the decision and its execution is reduced compared to the centralised deployment of DEs. It contributes to faster and more stable slice management by reducing the transient phase.

6.2.2.2 ORCHESTRATION SCALABILITY OF MONB5G ARCHITECTURE

The approach to the orchestration of MonB5G slices is different than the one proposed by ETSI MANO. The main differences and their contribution to scalability are the following:

- The orchestrator is agnostic to slices and its operations are focused on resources only. Therefore, it has fewer duties than in the ETSI MANO case.
- ISM of each slice has the ability to trigger an orchestration event (resource scaling, adding VNFs, etc.). It is no longer the role of the centralised OSS/BSS. Due to the distribution of the process the runtime orchestration (elaborating of the triggering events) is much more scalable than in the referred case.
- The ISM-triggered orchestration events can be proactive, for example, based on the number of logged to slice users opposite to reactive resource scaling provided by MANO. Moreover, the orchestration operations can be adapted (SML resource consumption prediction algorithms) in a way to reduce the number of orchestration events (at the cost of resource utilization effectiveness).

It has to be noted that the MonB5G architecture presented in this deliverable is a reference one and it can be implemented in many ways. The implementation specifics may impact the architecture's scalability and flexibility.

6.2.3 SCALABILITY EVALUATION

To accurately demonstrate the scalability, MonB5G will rely on extensive testbed measurements to devise an analytical model of the trend of evolution of some KPIs vs. the number of admitted slices on one hand and the supported slices vs. network resources on the other hand. These trends models will be valid and directly extrapolated to the massive slicing regime. In this respect, several evaluation methodologies can be considered:

- Characterizing e.g., the slice setup time and showcasing that it presents a linear/sublinear behaviour compared to a state-of-the-art central MANO solution as depicted in Figure 53.
- As exemplified in Figure 54, demonstrating that the overall AI performance (e.g., accuracy, loss) is not degraded while guaranteeing that the induced overhead does not increase super-linearly

with the number of slices as a result of MonB5G low footprint decentralised AE/DE architectures and algorithms.

- Ensuring that the trend of the number of supported slices with fulfilled SLA scales at least linearly with the increase of allocated resources compared to centralized MANO that could present sublinear scalability.

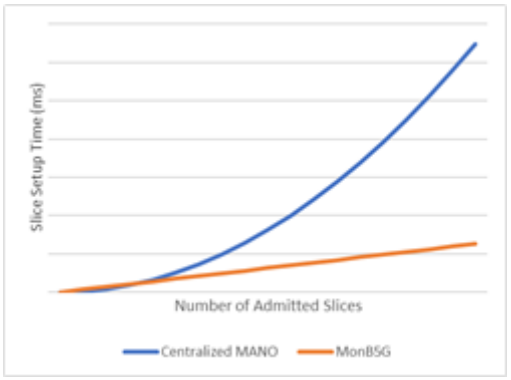

Figure 53. Slice setup time scalability.

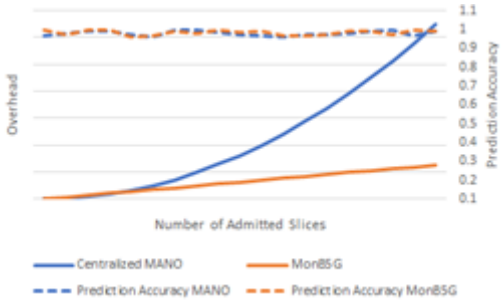

Figure 54. Overhead/prediction performance scalability.

7 Remarks on the implementation of the MonB5G architecture

This section provides a mapping between the reference architecture described earlier and a set of existing tools for future implementation.

7.1 Tools to be used for the implementation of MonB5G architecture

The architecture depicted in Figure 55 represents three Technological Domains hosted by different infrastructure virtualisation solutions (AWS, OpenStack[49], Docker[50], Kubernetes), which represent the infrastructure layer.

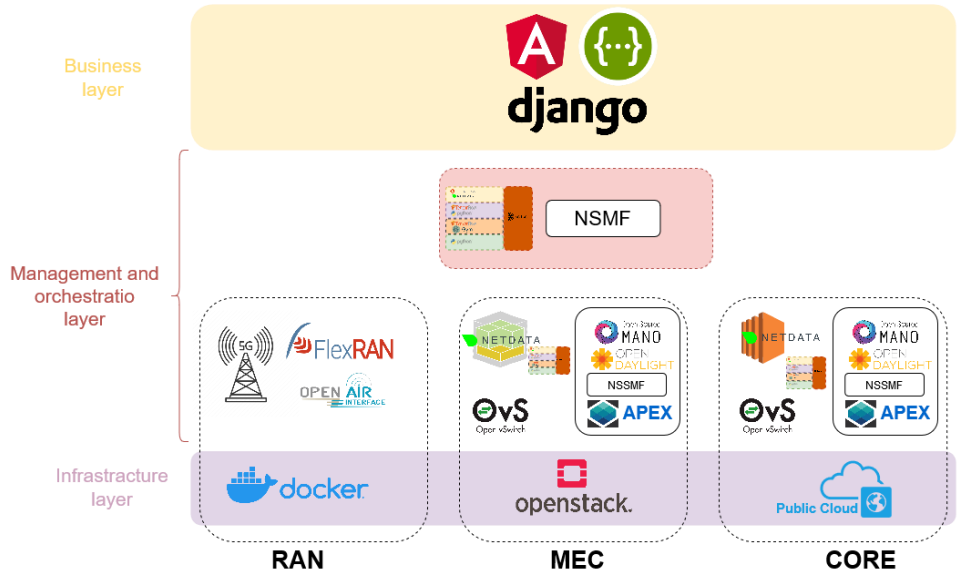

Figure 55. Mapping of MonB5G reference architecture and existing tools.

For the Management and Orchestration Layer in the Core (Cloud) and MEC (Edge), we have mainly ETSI OSM[51] for VNF management and orchestration in addition to the OpenDaylight (ODL)[52], an SDN controller that controls the links between different nodes using Open vSwitches[53]. CNFs, which are the nature of MonB5G Slice Management Layer (SML) components (e.g., MS, AE and DE functions) are deployed on top of Kubernetes, which acts as a Platform as a Service (PaaS) aligned with ETSI NFV-IFA 029 [58]. Additionally, APEX[54] for policy management and execution is deployed. The RAN domain in our approach uses the FlexRAN[55] on top of OAI[56] for controlling and slicing the RAN domain. NSMF and NSSMF contain the slice management functions that are supposed to be developed since there is no open-source tool that can provide the required functions to the best of our knowledge. The Business Layer has web interfaces and APIs with a database to store authentication credentials. Many tools are available on the market, e.g., Django[57]/Angular[58] and Swagger[59] are among the most popular stable tools and specifications. MonB5G is heavily based on the AI-driven management (underpinned by MS, AE, and DE components) described earlier for autonomous management. For the implementation, the NetData[60] tool can be used to gather and expose resource-related and network telemetry of the deployed VNF instances

[49] OpenStack, [Online]. Available: https://www.openstack.org/.
[50] Docker, [Online]. Available: https://www.docker.com/.
[51] ETSI Open Source MANO, [Online]. Available: https://osm.etsi.org/.
[52] OpenDaylight, [Online]. Available: https://www.opendaylight.org/.
[53] Open vSwitch, [Online]. Available: https://www.openvswitch.org/.
[54] APEX, [Online]. Available: https://docs.onap.org/en/dublin/submodules/policy/apex-pdp.git/docs/APEX-Introduction.html.
[55] FlexRAN, [Online]. Available: https://mosaic5g.io/flexran/.
[56] Open Air Interface, [Online]. Available: https://openairinterface.org/.
[57] Django, [Online]. Available: https://www.djangoproject.com/.
[58] Angular, [Online]. Available: https://angular.io/.
[59] Swagger, [Online]. Available: https://swagger.io/.
[60] NetData, [Online]. Available: https://www.netdata.cloud/.

and PNFs. Prometheus[61] might be used to automatically scrape custom metrics from SML components themselves, leveraging information from several customised agents to derive slice-specific KPIs. AEs and DEs are mostly AI-based and are planned to be developed using the ML frameworks such as Python TensorFlow[62], Pytorch[63] and Open-AI gym[64], as presented in Figure 56. The communication between layers MS, AE and DE can be based on publish/subscribe tools like Kafka[65].

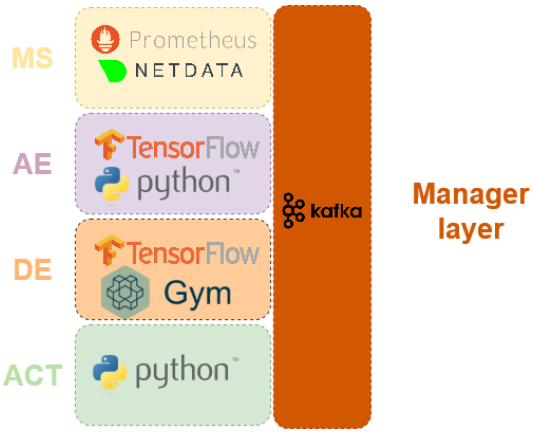

Figure 56. MLaaS instantiation (example).

7.2 Implementation of MS/AE/DE functions

In this section, we will provide more information about the implementation and interactions between different MS/AE/DE components of the architecture. The description is just exemplary as MS/AE/DE sublayers and components are part of a slice (SML or MLaaS) orchestrated as a PaaS NFV Object. There are, however, some recommendations for the implementation of the components. We will show an example implementation of the components on selected MonB5G Use-Cases (UCs).

7.2.1 MONITORING SYSTEM ROLE AT DIFFERENT LEVELS OF MANAGEMENT HIERARCHY

In MonB5G, the management system is distributed for scalability reasons. It especially concerns the monitoring part of the management. The proposed separation splits the slice-specific monitoring from resource-oriented monitoring, but also focuses on placing MS components closest to the monitored SFL elements, preventing raw telemetry from saturating inter-domain network links.

7.2.1.1 SLICE-RELATED MONITORING

For the slice part, the producers of monitoring data are typically its composing virtual functions (i.e., EMs or EEMs); moreover, the resulting aggregated and correlated monitoring information contributes to the estimation of slice-wide telemetry collected at the Slice Manager of SML. In the case of multi-domain slices, the monitoring information about (sub)slices that compose the E2E slices are processed by AE functions and published to IDSM via streaming busses. It has to be noted that each level of the autonomic slice management hierarchy i.e., node/function level, slice level and inter-domain level; may also be consumers of telemetry, and generally, such monitoring data are not exchanged among them as a primary (raw) format. The embedded intelligence implemented at different levels of the management hierarchy allows for the use of the concept of intent-based interfaces.

At the slice level, the monitoring information is consumed after pre-processing for the purpose of the self-management based on multiple AE and DE functions of its respective SML (or namespace in a MLaaS).

---

61 Prometheus, [Online]. Available: https://prometheus.io/.
62 Tensorflow, [Online]. Available: https://www.tensorflow.org/.
63 Pytorch, [Online]. Available: https://pytorch.org/.
64 Open-AI gym, [Online]. Available: https://gym.openai.com/.
65 Kafka, [Online]. Available: https://kafka.apache.org/.

As envisioned in MonB5G, the principal consumer of MS telemetry are AEs. Indeed, AEs oversee triggering the monitoring of needed information from MS. The latter starts the monitoring process by connecting to the appropriate source, i.e., infrastructure or Function. Accordingly, MS exposes two types of APIs: Control and Data Collection APIs. The MS Control API may be used by AE (or other SML components) to request specific metrics to monitor the sampling periodicity, metrics resolution, sampling duration, data format, etc. Additionally, it admits configuration specifying telemetry retrieval methods, i.e., publish/subscribe, request/response. Data Collection APIs are the interfaces from which metrics are exposed to AE in the manner requested via the Control API. These are, in turn, segmented into bulk retrieval and streaming interfaces exposing a local Time Series Data Base (TSDB) or a streaming bus, respectively.

Besides collecting monitoring data and providing API for AE to control and consume monitored data, MS implements functions to treat the collected data as described in section 3.4.2.1. MS may transform data by adding semantic and context information, such as the timestamp, source of data (e.g., DMO, function, type, etc.), metric name, and value. Moreover, MS may use persistent storage to store monitored data, their interpolation and extrapolation for future requests.

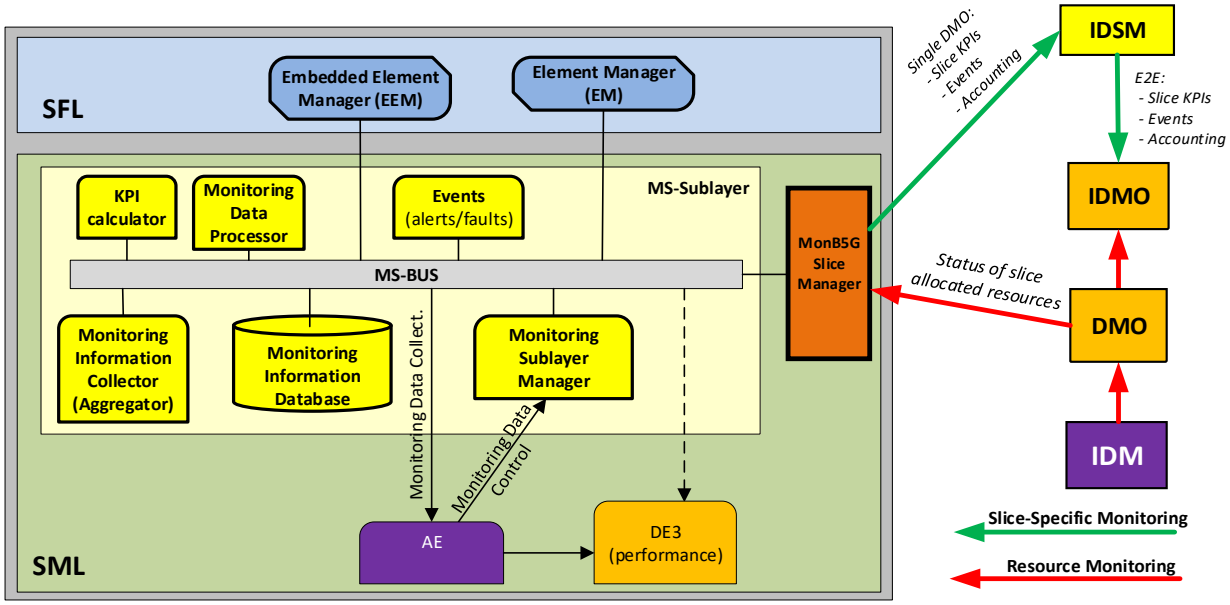

Figure 57. Overall E2E monitoring information exchange in case of a multi-domain slice (an example).

Pre-processed MS information may be retrieved from outside its Technological Domain by means of DMO, IDMO or the Slice Tenant Management Interface (or Slice Management Provider Interface). This external SML information exchange is mostly used for the exchange of slice-related KPIs and accounting related information, security incidents and for passing information about faults. The slice-related KPIs are calculated by MS and transferred externally by the Slice Manager of SML. The Slice Manager is also a consumer of the fault-related information coming from EMs/EEMs (direct faults), respective AEs of SML (faults identified by AEs) or from DMO (concerns only infrastructure faults linked with resources allocated to a slice).

Faults and security-related incidents are in a direct form passed through all levels of MS hierarchy. In cases when proposed mitigation has been taken at any level, this information is reported by a corresponding MS function. It has been assumed that in case of resources, the mitigation actions are taken first by IDM, next by DMO and if DMO cannot handle the fault, it sends the information to SML, which will request a fault mitigation action at the slice level (i.e., by IDMO). When fault details are sent to IDMO, it will attempt to find a solution by deploying the slice or sub-slices in a different domain or terminating it if such operation is not feasible.

The information exchange between MS components of different levels of the MS hierarchy in the case of a single slice is presented in Figure 57. In the given case, a multi-domain slice has been shown. In the case of a single domain slice, IDMO does not exist, and the information from Slice Manager goes directly to DMO.

It must be noted that a significant part of MS is included in a slice template; therefore, the MS-related internal interfaces enabling SML interactions do not have to use standardised interfaces. However, many MS functions can be common for diverse types of slices; therefore, some implementation guidelines can be given. The form of the information exchange between SMLs, DMOs and IDMO can be, to a certain extent, universal. The issue is left for further study and will be addressed during the MonB5G architecture implementation.

Table 7 shows some examples of monitored KPIs by MS of SML for selected MonB5G use -cases. We organise this table according to the technological domain from which it has been generated and where it will be demonstrated. We recall our use case scenarios:

- UC1Sc1: Zero-Touch multi-domain service management with E2E SLAs.
- UC1Sc2: Elastic E2E slice management.
- UC2Sc1: Attack identification and mitigation.
- UC2Sc2: Robustness of learning algorithms in the face of attacks.

Table 7. Slice-related KPI examples.

Source	Technological Domain	Metric/KPI	MonB5G UC
	RAN, Edge, Cloud	Latency	UC1Sc1 and UC1Sc2
	RAN, Edge	Bit rate	UC1Sc1 and UC1Sc2
	RAN	Packet Loss Rate	UC1Sc1 and UC1Sc2
	VNF (Mobility Management Entity/ Access and Mobility Management Function)	Number of UE attach	UC2Sc1
	Cloud or Edge NFVI	CPU and memory consumption of used VNFs	UC2Sc2
	MEC Platform	User access	UC2Sc1
	RAN, Edge, Cloud	SLA Violations	UC1Sc1 and UC1Sc2
	RAN, Edge, Cloud	Reaction time to NS malfunction	UC1Sc1
	Cloud or Edge NFVI	Network Energy Efficiency	UC1Sc2
Function	–	Video Quality (QoE)	UC1Sc1 and UC1Sc2
	–	Latency	UC1Sc1 and UC1Sc2
	–	Service response time	UC1Sc1 and UC1Sc2

7.2.1.2 RESOURCE-RELATED MONITORING

Resource monitoring is separated from slice monitoring; however, some information between the two monitoring areas is exchanged. Resources monitoring at the infrastructure level is done for the purpose of determining availability, consumption and faults. Moreover, the information about the energy consumption by slice allocated resources is fed from IDM to DMO. The interactions between IDM and DMO allow DMO to collect information on:

- NFVI: such as computing platforms and hardware;
- Physical Network Function (PNF) running network functions on dedicated hardware: such as eNB/gNB, router, and UPF;
- VNFs running common virtualised network functions: such as Core Network (CN) functions or DNS (including DFS).

The collected information is used by DMO for LCM operations on slices, like admission control and resource scaling.

7.2.2 ANALYTICAL ENGINES

Similar to MS, Analytical Engines (AEs) in MonB5G are distributed in the different parts of the management system. They perform a key role in SML by looking for anomalies related to a single slice operation, in DMO by detecting resource anomalies, and in IDMO for spotting E2E-related anomalies. The AEs of different management components (SML, DMO, IDMO, IDM) do not cooperate as they typically prepare data for DEs.

As opposed to MS, AE does not store but processes data gathered from the same or lower-level MS or AE; and exposes the result to any requester (i.e., DE or another AE) in an on-demand or periodic fashion. AE to AE communication is possible, mainly to build a learning model using federated learning techniques.

The main functions of AE are to: (i) identify performance degradation or a fault of a network slice; (ii) optimise the performance of a network slice or the DMO resources; (iii) react to security threats. To this aim, AE subscribes to data types it is interested in, using the Control API exposed by MS. The data type will be determined according to the logic of the LCM application runs. Then, AE starts receiving the stream of data or uses a request/response mechanism (i.e., via MS Data Collection APIs), depending on the purpose of the analysis.

AE may adapt the monitoring data rate or stop the request and request for other related monitoring information. AE can complete an inference task locally, extract features, and analyse these features and send alerts and notifications to DE. AE may collaborate with other AEs to build distributed learning (based on federated learning) model to realise the analysis and notify DE accordingly. For the adaptation of MS, an AE has to interact with the Slice Manager of SML.

Table 8 shows examples of AE considering the MonB5G use-cases:

Table 8. AE examples.

Feature	PoC of MonB5G
Prediction of SLA violation	UC1Sc1, UC1Sc2 and UC2Sc1
Prediction of NS faults	UC1Sc1 and UC1Sc2
Attack identification	UC2Sc1
Anomaly detection	UC2Sc1
Prediction of service migration	UC2Sc2

7.2.3 DECISION ENGINES

Decision Engines (DEs) exist in different layers of the architecture. Moreover, there can be several DEs (i.e., DE functions) in the same layer that realise competing goals. In the MonB5G architecture, the conflicts between DEs can be solved by several mechanisms. The first one is a Decision Arbiter that selects the DE output to be enforced according to active policy.

Another separation is provided by the range of the DE-related reconfiguration. Some DEs operate on separate slices (i.e., SFLs), some may also operate on their SML, some DEs are oriented towards resources (IDM, DMO) others towards slice orchestration (DMO), or may be used during slice deployment preparation phase to provide an optimal split of a slice template into multiple domains.

The third kind of separation between DEs is the time scale. It is expected that the fastest control-loops will reside inside EEMs, slower ones inside SML, yet slower reconfigurations will be done by DMO. That way, somehow interfering reconfigurations will be separated, even if their impact is global.

As depicted in Figure 58, DE is the decision-making element of the MonB5G architecture. It analyses alerts and notifications from AE(s) and considers a decision to take. The decisions are either derived using a local ML algorithm, based mainly on Reinforcement Learning (RL), or a predefined policy enforced by the Tenant or DMO through Intent, or a combination of both. DE may collect notification from several AEs,

and may interact with MS monitoring of different technological domains, to consider a global decision on the E2E slice. Global decisions are mainly considered at the IDMO level.

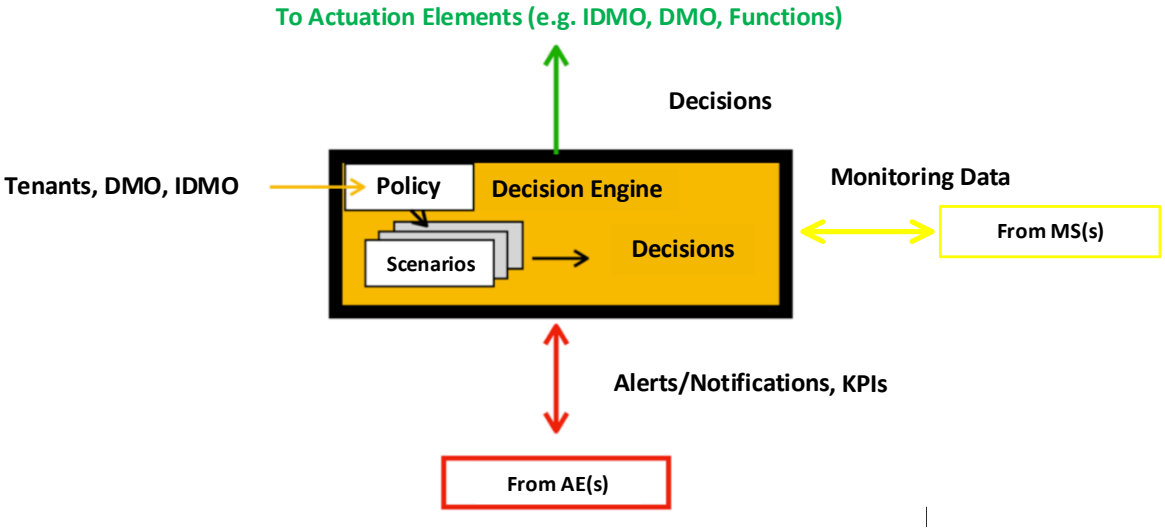

Figure 58. DE interactions.

DE interacts with actuation elements (function, DMO, or IDMO) to enforce the considered decisions. For local decisions, DE interacts with DMO and slice virtual function, while for global decisions, DE has to interact with IDMO. Table 9 shows examples of DE decisions in relation to MonB5G use cases. We classified the decision according to their scopes, i.e., local or global.

Table 9. DE examples.

Decision	Type	MonB5G UC
Scale a VNF	Local	UC1Sc1
Energy optimization	Global	UC1Sc1
Increase the RAN resources for an NS	Local	UC2Sc2
Block a UE connection	Global	UC2Sc1
Service migration	Local	UC2Sc2
VNF placement	Global	UC1Sc2

7.3 The use of PaaS in MonB5G

MonB5G leverages the concepts proposed in [58], particularly the definition of Platform as a Service (PaaS) as a new NFV Object. This concept is used for SML of a slice, which, if it is set to manage various SFL (i.e., Management Layer as a Service (MLaaS)), will contribute to the reduction of the slice footprint (in terms of resource consumption and cost). Relevantly, PaaS instances offered to MonB5G SFLs (as VNF Common/Dedicated Service) are referred to as Dynamically Shared Functions (DSF). Both concepts use the same implementation mechanisms and can be used simultaneously to the same slice. It has to be noted, however, that in most cases, DSF and MLaaS are not generic but SFL template specific.

DSFs as Common Services (DSF-C) must not be solely operated by a single Slice Tenant or by a DMO operator. Therefore, in both cases, there is a need to use the Slice Management Provider (SMP) as an operator. From a management perspective, DSF and MLaaS constituent system resources (i.e., VNFs) are considered slices. Therefore, their usage can be seen as vertical stitching of slices. Instances of such services (MLaaS, DFS) may be deployed by IDMO before the tenant resources.

ETSI NFV-IFA 029 [58] provides support for container-based services. In general, most of the mechanisms described there are applicable to the mentioned use cases of PaaS, especially those related to security (Section 8.4 [58]). The usage of PaaS comes with the mentioned above benefits (e.g., fast reconfigurability, proximity to SFL components, reduced resource footprint, etc.), despite having a main negative consideration regarding reduced security.

To increase the isolation and security of slices, the 3GPP recommendations included in [91] can be reused. To that end, Network Function (NF) discovery and registration shall be authorized and should support confidentiality, integrity, and replay protection of data. The mutual authentication between NF Service Consumer and NF Service Producer shall be supported. The deployment of some functions in NFVI, seen in a similar way as PNF (especially if they use hardware acceleration) must also be considered. Fortunately, recent advancements on popular Container Orchestration[66] Engines (COEs) allow fine-grain access control for these features[67]. The impact of PaaS as NFVI resource is left for further study.

7.3.1 DSF IMPLEMENTATION

Despite being defined for SFL use, DSF may also have an SML sub-layer holding MonB5G management components. Management of such combination of sub-layers may be delegated to a Slice Management Provider or be subject to policies from other components of the management system (SML/MLaaS of the corresponding Technological Domain(s)). Particularly, if the management of DSF is done by MonB5G SML, their functions must already be defined, and MonB5G security mechanisms used.

7.3.2 MONB5G MANAGEMENT AS A SERVICE IMPLEMENTATION

A tenet of MonB5G reference architecture is the concept of the fast local control loop and increasingly slower ones for wider-scope administrative domains (e.g., slice-level, tenant-level, etc.). Leveraging the proposed concept of MLaaS, it is evident that many software components (i.e., MS-C, AE-C and DE-C) need to be able to interact to realize such control loops. Moreover, IDMO/IDSM may require changes in its administrative components (e.g., lifecycle management operations, reconfiguration of parameters) to comply with new objectives or to ensure support for an increasing number of managed SFLs.

Figure 59 shows a simplified version of a slice according to MonB5G reference architecture. It includes a Slice Functional Layer (SFL) and MonB5G Layer as a Service (MLaaS). The former is composed of tenants' VNF and corresponding EEMs/Ems that allow MLaaS components to gather metrics and/or actuate upon such resources. MLaaS is in turn composed of VNFs providing the Container Infrastructure System (CIS) (i.e., Container Infrastructure Service Instance and Managers, CISI and CISM, respectively), and the PaaS itself (as a COE), which effectively acts as a runtime environment for MonB5G administrative elements and components. MLaaS management services are therefore conceived as cloud-native applications build-

[66] Container orchestration is the automation of much of the operational effort required to run containerised workloads and services. This includes a wide range of things software teams need to manage a container's lifecycle, including provisioning, deployment, scaling (up and down), networking, load balancing and more.

[67] See: https://kubernetes.io/docs/tasks/administer-cluster/topology-manager/.

out of interaction among several MonB5G components of SML (i.e., MS-C, AE-C, DE-C, Slice Manager). This is exemplified by NFV MANO and Infrastructure Optimizations in the figure.

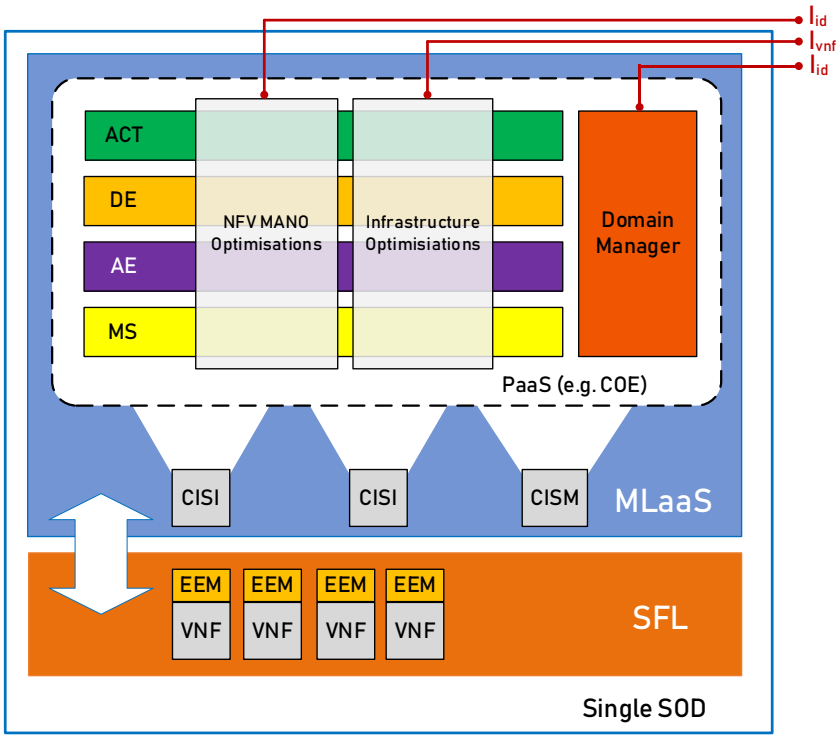

Figure 59. Single SOD DMO instance.

Leveraging COE for hosting MLaaS allows dynamic reconfiguration of components, custom resource scaling mechanisms for supporting an increasing number of SFLs, and compatibility with ETSI NFV via the models proposed in [58].

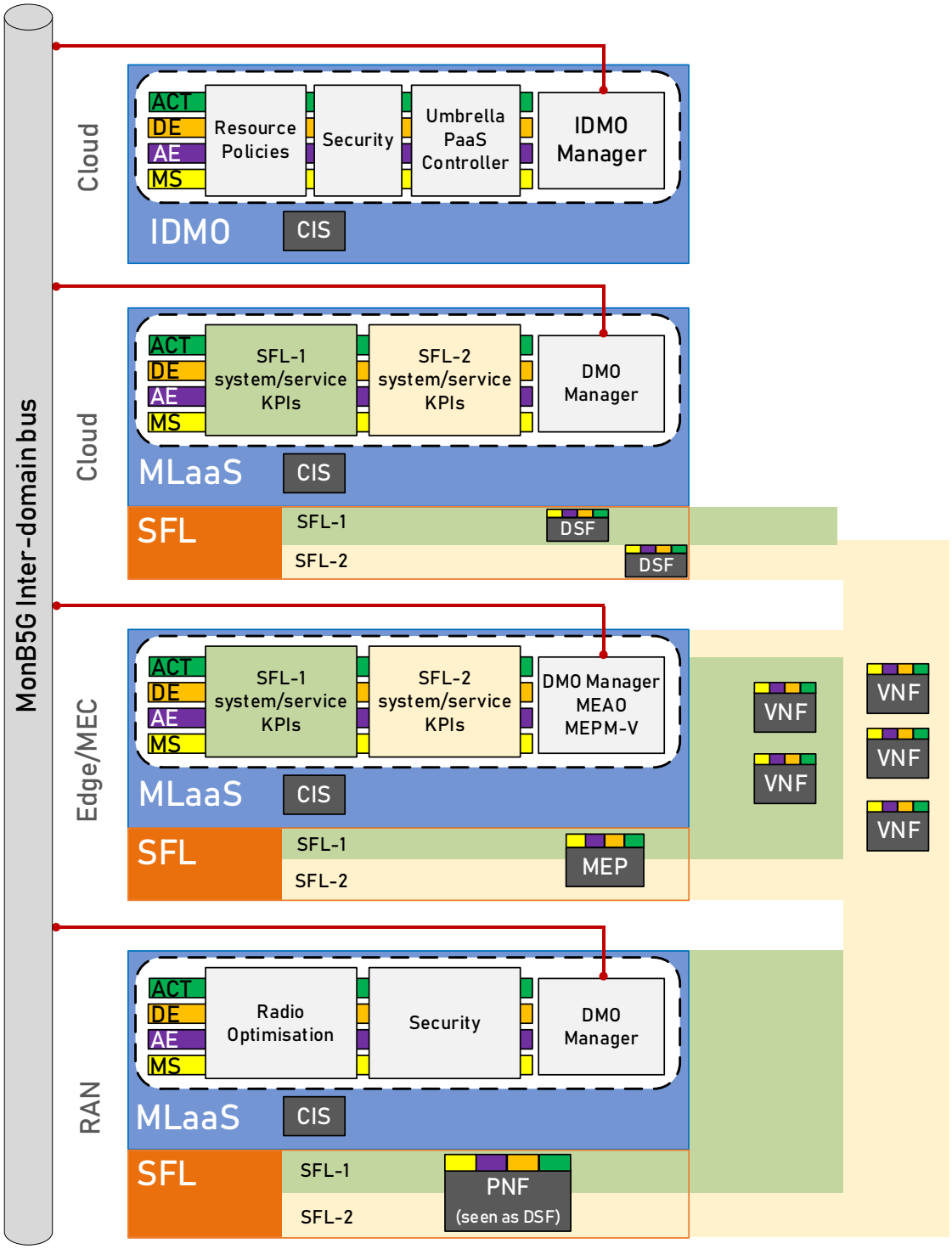

Figure 60. E2E network slice with MLaaS used at several levels.

The case of IDSM/IDMO, therefore, implies the instantiation of MLaaS at different technological domains (e.g., Edge, Cloud). Resources allocation is achieved via IDMO's NFVO, while a centralised provision of MonB5G software components is achieved leveraging CIS federation (e.g., Kubernetes KubFed API[68]). Figure 60 further elaborates by providing an instantiation of an E2E MonB5G Network Service, that is, containing MLaaS at several technological domains (e.g., Cloud, Edge/MEC and RAN).

The figure highlights the existence of two E2E network slices, denoted as SFL-1 and SFL-2. These Network Slice Instances have components (i.e., SFL) at each technological domain. Therefore, they are also

[68] See: https://github.com/kubernetes-sigs/kubefed.

accompanied by a corresponding management layer. In the case of the figure, each technological domain hosts an MLaaS managing both SFLs (i.e., NSSIs). Relevantly, IDMO in the Cloud domain holds a so-called Umbrella PaaS Controller, which is used to provision MLaaS across technological domains, respectively. Also, it is worth mentioning that DSFs are independent NFV Objects and are included at SFL in each technological domain to highlight the services offered to such layer.

The shown instantiation allows fast control loops at the VNF level by leveraging MLaaS and EEM, while wider-scoped and therefore slower control loops are exercised at the Slice (RAN, Edge and Cloud technological domains) and OSS/BSS (multi-slice) level.

8 Conclusions

In this deliverable, the final MonB5G zero-touch slice management and orchestration framework that facilitates the deployment of a massive number of slices in different administrative and technological domains is presented. In order to achieve the scalability goal, we have used a distribution of AI-driven management functions at all levels of the management hierarchy. Distributed management reduces the amount of the information exchanged for management purposes and taking some decisions locally reduces the management system response time. The use of distributed components with embedded intelligence has made it possible to use the intent-based interfaces that also contribute to the reduction of the information exchange between management functions and subsystems. Moreover, we have used a multi-domain orchestration solution, as well as separation of each slice's management from domain resource management. The use of the in-slice management concept (ISM) has reduced the number of slice external interfaces and provides a perfect separation of the slice management plane that cannot be achieved in the 3GPP approach to network slicing management. The implementation of slice management as a part of a slice (i.e., a set of VNFs) provides higher scalability of slice management. It allows for the programmability of slice management services on the fly. That includes slice-related security services. The in-slice management can play the role of its slice modification requester, typically based on slice-specific analysis, enabling proactive management operations and contributing to the agnostics of the slice orchestrator. In MonB5G, the slice orchestrator is mainly focused on domain resources and is linked with OSS/BSS that performs appropriate functions. The domain-based approach reduced the overall management traffic. To reduce it more, we have used the well-known, KPI-based approach to exchange monitoring information between domains. To include resource brokering and energy-efficient operations, we have proposed a modification of the existing interfaces between the infrastructure and other components of the architecture. The usage of PaaS is another novelty of the MonB5G architecture, while the MLaaS approach is a nice bridge between the existing concepts, like 3GPP one, and the self-managed slice as proposed by MonB5G. By using it, we have introduced a new entity that is called Slice Management Provider.

Even though the deliverable describes the final version of the architecture, we consider its update at the end of the project if the progress of other work packages and the experience gained during its implementation will require this, i.e., we will notice a need for architecture modification.

List of figures

List of tables

References

[1] NGMN Alliance, “NGMN 5G White Paper”, Jan. 2015.

[2] S. Sharma, R. Miller, A. Francini, “A cloud-native approach to 5G network slicing”, IEEE Communications Magazine 55(8), pp. 120–127, 2017.

[3] J. Elliott, S. Sharma, “Dynamic E2E Network Slicing unlocks 5G Possibilities”, Nokia (2016). [Online]. Available: https://www.nokia.com/blog/dynamic-end-end-network-slicing-unlocks-5g-possibilities/.

[4] I. P. Chochliouros, A. S. Spiliopoulou, P. Lazaridis, A. Dardamanis, Z. Zaharis, A. Kostopoulos, “Dynamic Network Slicing: Challenges and Opportunities”, In: Proceedings of AIAI-2020, IFIP Advances in Information and Communication Technology, vol. 585, pp. 47–60. Springer, Cham (2020), doi: 10.1007/978-3-030-49190-1_5.

[5] I. Idio, R. Rufus, A. Esterline, “Artificial registration of network stress to self-Monitor an autonomic computing system”, In: Proceedings of the SoutheastCon 2017, pp. 1–7, doi: 10.1109/SECON.2017.7925334.

[6] A. J. Gonzalez et al., “The Isolation Concept in the 5G Network Slicing”, in EuCNC 2020, Dubrovnik, Croatia, pp. 12–16, 2020, doi: 10.1109/EuCNC48522.2020.9200939.

[7] ETSI, “Report on Network Slicing Support with ETSI NFV Architecture Framework”, ETSI GR NFV-EVE 012, V3.1.1, Dec. 2017.

[8] 3GPP, “Management and orchestration; Architecture framework”, 3GPP TS 28.533, V17.0.0, Sep. 2021.

[9] 3GPP, “Management and orchestration; Concepts, use cases and requirements”, 3GPP TS 28.530, V17.1.0, Apr. 2021.

[10] S. Kukliński, L. Tomaszewski, “DASMO: A scalable approach to network slices management and orchestration”, NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, pp. 1–6, 2018, doi: 10.1109/NOMS.2018.8406279.

[11] T. Taleb, I. Afolabi, K. Samdanis, F. Z. Yousaf, “On Multi-domain Network Slicing Orchestration Architecture and Federated Resource Control”, IEEE Network, vol. 33, no. 5, pp. 242–252, 2019, doi: 10.1109/MNET.2018.1800267.

[12] L. Tomaszewski, S. Kukliński, and R. Kołakowski, “A New Approach to 5G and MEC Integration”, in: I. Maglogiannis, L. Iliadis, E. Pimenidis (eds.), Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops. AIAI 2020”, IFIP Advances in Information and Communication Technology, vol. 585, Springer, Cham (2020), doi: 10.1007/978-3-030-49190-1_2

[13] S. Kukliński, L. Tomaszewski, R. Kołakowski, “On O-RAN, MEC, SON and Network Slicing integration”, in 2020 IEEE Globecom Workshops, Taipei, Taiwan, pp. 1–6, 2020, doi: 10.1109/GCWkshps50303. 2020.9367527.

[14] ETSI, “Network Functions Virtualisation (NFV); Testing; Report on CICD and Devops”, ETSI GR NFV-TST 006, V1.1.1, 2020-01

[15] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, V. C. M. Leung, “Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges”, in IEEE Communications Magazine, vol. 55, no. 8, pp. 138–145, Aug. 2017, doi: 10.1109/MCOM.2017.1600940.

[16] S. Zhang, “An Overview of Network Slicing for 5G”, in IEEE Wireless Communications, vol. 26, no. 3, pp. 111–117, Jun. 2019, doi: 10.1109/MWC.2019.1800234.

[17] X. Li et al., “Network Slicing for 5G: Challenges and Opportunities”, in IEEE Internet Computing, vol. 21, no. 5, pp. 20–27, 2017, doi: 10.1109/MIC.2017.3481355.

[18] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, H. Flinck, “Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions”, in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429–2453, thirdquarter 2018, doi: 10.1109/COMST.2018.2815638.

[19] O. U. Akgul, I. Malanchini, A. Capone, “Anticipatory resource allocation and trading in a sliced network”, in 2019 IEEE International Conference on Communications (ICC 2019), pp. 1–7, 2019.

[20] L. Sanabria-Russo, L. Righi, D. Pubill, J. Serra, F. Granelli, C. Verikoukis, “LTE as a service: Leveraging NFV for realizing dynamic 5G network slicing”, in 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, 2019.

[21] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, J. Folgueira, “Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges”, IEEE Communications Magazine, vol. 55, no. 5, pp. 80–87, 2017.

[22] U. Paul, J. Liu, S. Troia, O. Falowo, G. Maier, “Traffic-profile and machine learning based regional data center design and operation for 5G network”, Journal of Communications and Networks, vol. 21, no. 6, pp. 569–583, 2019.

[23] M. R. Raza, M. Fiorani, A. Rostami, P.¨ Ohlen, L. Wosinska, P. Monti, “Dynamic slicing approach for multi-tenant 5G transport networks [invited]”, IEEE/OSA Journal of Optical Communications and Networking, vol. 10, no. 1, pp. A77–A90, 2018.

[24] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, X. Costa-Pérez, “Resource sharing efficiency in network slicing”, IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 909–923, 2019.

[25] B. Han, V. Sciancalepore, D. Feng, X. Costa-Perez, H. D. Schotten, “A utility-driven multi-queue admission control solution for network slicing”, in IEEE INFO-COM 2019 - IEEE Conference on Computer Communications, pp. 55–63, 2019.

[26] Next Generation Mobile Networks (NGMN) Alliance (2020): 5G White Paper 2, v1.0. [Online]. Available: https://www.ngmn.org/work-programme/5g-white-paper-2.html.

[27] V. P. Kafle, P. Martinez-Julia, T. Miyazawa, “Automation of 5G network slice control functions with machine learning”, IEEE Communications Standards Magazine, vol. 3, no. 3, pp. 54–62, 2019.

[28] ABI Research, “AI and Operations Automation in 5G Networks”, 2020.

[29] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, A. Banchs, “Mobile traffic forecasting for maximizing 5G network slicing resource utilization”, in IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, pp. 1–9, 2017.

[30] V. Sciancalepore, X. Costa-Perez, A. Banchs, “RL-NSB: Reinforcement learning-based 5G network slice broker”, IEEE/ACM Transactions on Networking, vol. 27, no. 4, pp. 1543–1557, 2019.

[31] E. Kapassa, M. Touloupou, D. Kyriazis, “SLAs in 5G: A complete framework facilitating VNF- and NS- tailored SLAs management”, in 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 469–474, 2018.

[32] S. Zheng, Z. Gao, X. Shan, W. Zhou, Y. Wang, “Tail latency optimized resource allocation in fog based 5G networks”, in 2018 IEEE Symposium on Computers and Communications (ISCC), pp. 00249–00254, 2018.

[33] C. Gutterman, E. Grinshpun, S. Sharma, G. Zussman, “RAN resource usage prediction for a 5G slice broker”, Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2019.

[34] M. Maule, P. Mekikis, K. Ramantas, J. Vardakas, C. Verikoukis, “Real-time dynamic network slicing for the 5G radio access network”, in 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, 2019.

[35] C. Luo, J. Ji, Q. Wang, X. Chen, P. Li, “Channel state information prediction for 5G wireless communications: A deep learning approach”, IEEE Transactions on Network Science and Engineering, vol. 7, no. 1, pp. 227–236, 2020.

[36] D. Clemente, G. Soares, D. Fernandes, R. Cortesao, P. Sebastiao, L. S. Ferreira, “Traffic forecast in mobile networks: Classification system using machine learning”, in 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pp. 1–5, 2019.

[37] S. Xiao, W. Chen, “Dynamic allocation of 5G transport network slice bandwidth based on LSTM traffic prediction”, in 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), pp. 735–739, 2018.

[38] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, J. Rao, “Ai-assisted network-slicing based next-generation wireless networks”, IEEE Open Journal of Vehicular Technology, vol. 1, pp. 45–66, 2020.

[39] N. Ferdosian, M. Othman, K. Y. Lun, B. M. Ali, “Optimal solution to the fractional knapsack problem for LTE overload-state scheduling”, in 2016 IEEE 3rd International Symposium on Telecommunication Technologies (ISTT), pp. 97–102, 2016.

[40] Q. Yang, Y. Liu, T. Chen, Y. Tong, “Federated machine learning: Concept and applications”, ACM Trans. Intell. Syst. Technol., vol. 10, Jan. 2019.

[41] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, D. Bacon, “Federated learning: Strategies for improving communication efficiency”, in NIPS Workshop on Private Multi-Party Machine Learning, 2016.

[42] M. Mohri, G. Sivek, A. T. Suresh, “Agnostic federated learning”, in Proceedings of the 36th International Conference on Machine Learning (K. Chaudhuri, R. Salakhutdinov, eds.), Proceedings of Machine Learning Research, (Long Beach, California, USA), vol. 97, pp. 4615–4625, PMLR, Jun. 2019.

[43] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, L. Liang, “Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications”, in 2019 IEEE 37th International Conference on Computer Design (ICCD), pp. 246–254, 2019.

[44] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, “Federated learning: Challenges, methods, and future directions”, IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50–60, 2020.

[45] X. Yao, T. Huang, C. Wu, R. Zhang, L. Sun, “Towards faster and better federated learning: A feature fusion approach”, in 2019 IEEE International Conference on Image Processing (ICIP), pp. 175–179, 2019.

[46] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, C. Miao, “Federated learning in mobile edge networks: A comprehensive survey”, IEEE Communications Surveys Tutorials, pp. 1–1, 2020.

[47] R. Fantacci, B. Picano, “Federated learning framework for mobile edge computing networks”, CAAI Trans. Intell. Technol., vol. 5, pp. 15–21, 2020.

[48] Y. Xiao, Q. Zhang, F. Liu, J. Wang, M. Zhao, Z. Zhang, J. Zhang, "NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning", in 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), pp. 1–10, 2019.

[49] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey", IEEE Communications Surveys Tutorials, vol. 21, no. 4, pp. 3133–3174, 2019.

[50] A. Othman, N. A. Nayan, "Efficient admission control and resource allocation mechanisms for public safety communications over 5G network slice", Telecommunication Systems, vol. 72, pp. 595–607, Dec. 2019.

[51] Z. Luo, C. Wu, Z. Li, W. Zhou, "Scaling geo-distributed network function chains: A prediction and learning framework", IEEE Journal on Selected Areas in Communications, vol. 37, no. 8, pp. 1838–1850, 2019.

[52] ITU-T, "Overview of TMN Recommendations", ITU-T M.3000 (02/00), Feb. 2000.

[53] IBM, "Autonomic Computing White Paper: An architectural blueprint for autonomic computing", 3rd edition, 2006.

[54] ETSI, "Management and Orchestration", ETSI GS NFV-MAN 001, V1.1.1, Dec. 2014.

[55] ETSI, "Requirements based on documented scenarios", ETSI GS ZSM 001, V1.1.1, Oct. 2019.

[56] MonB5G, "Deliverable D2.2: Techno-economic analysis of the beyond 5G environment, use case requirements and KPIs", Dec. 2020.

[57] ETSI, "Report on Architectural Options", ETSI GS NFV-IFA 009, V1.1.1, Jul. 2016.

[58] ETSI, "Report on the Enhancements of the NFV architecture towards 'Cloud-native' and 'PaaS'", GR NFV-IFA 029, V3.3.1, Nov. 2019.

[59] NIST, "Framework for Improving Critical Infrastructure Cybersecurity", [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf (Accessed: 14 Oct. 2021).

[60] Ch. Benzaid, T. Taleb, C.-T. Phan, C. Tselios, G. Tsolis, "Distributed AI-based Security for Massive Numbers of Network Slices in 5G & Beyond Mobile Systems", 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), 2021, pp. 401–406, doi: 10.1109/EuCNC/6GSummit51104.2021.9482418.

[61] ETSI, "Management and Orchestration; Sc-Or, Sc-Vnfm, Sc-Vi reference points – Interface and Information Model Specification", ETSI GS NFV-IFA 033, V4.1.1, Aug. 2021.

[62] S. Rose, O. Borchert, S. Mitchell, S. Connelly, "Zero Trust Architecture", Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2020. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=930420 (Accessed: 14 Oct. 2021).

[63] C. Benzaïd, T. Taleb, "AI for Beyond 5G Networks: A Cyber-Security Defense or Offense Enabler?", in IEEE Network, vol. 34, no. 6, pp. 140–147, Nov./Dec. 2020, doi: 10.1109/MNET.011.2000088.

[64] MonB5G D2.3, Trust Model and Trust Management Approaches. Nov. 2020. https://www.monb5g.eu/wp-content/uploads/2020/12/D2-3_FINAL_MonB-5G.pdf.

[65] Q. Zhou, H. Huang, Z. Zheng, J. Bian, "Solutions to Scalability of Blockchain: A Survey", in IEEE Access, vol. 8, pp. 16440–16455, 2020, doi: 10.1109/ACCESS.2020.2967218.

[66] D. Ding, X. Jiang, J. Wang, H. Wang, X. Zhang, Y. Sun, "Txilm: Lossy block compression with salted short hashing", arXiv preprint arXiv:1906.06500. 2019, [Online]. Available: https://arxiv.org/abs/1906.06500.

[67] X. Dai, J. Xiao, W. Yang, C. Wang, H. Jin, "Jidar: A Jigsaw-like Data Reduction Approach Without Trust Assumptions for Bitcoin System," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), pp. 1317–1326, 2019, doi: 10.1109/ICDCS.2019.00132.

[68] J. Poon, T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments", [Online]. Available: https://lightning.network/lightning-network-paper.pdf (Accessed: 15 Oct. 2021).

[69] J. Teutsch, C. Reitwießner, "A scalable verification solution for blockchains", arXiv preprint arXiv:1908.04756, 2019, , [Online]. Available: https://arxiv.org/abs/1908.04756.

[70] A. N. Al-Quzweeni et al., "Optimized energy aware 5G network function virtualization", in IEEE Access, vol. 7, pp. 44939–44958, 2019.

[71] 3GPP, "Service requirements for the 5G system", 3GPP TS 22.261, V18.4.0, Sep. 2021.

[72] 3GPP, "Management and orchestration; Energy efficiency of 5G", 3GPP TS 28.310, V17.2.0, Sep. 2021.

[73] ITU-T, "Energy efficiency metrics and measurement methods for telecommunication equipment", ITU-T L.1310, Sep. 2020.

[74] 3GPP, "Telecommunication management; Study on system and functional aspects of energy efficiency in 5G networks", 3GPP TR 32.972, V16.1.0, Sep. 2019.

[75] J. Opadere, Q. Liu, T. Han, N. Ansari, "Energy-Efficient Virtual Radio Access Networks for Multi-Operators Cooperative Cellular Networks", in IEEE Transactions on Green Communications and Networking, vol. 3, no. 3, pp. 603–614, Sep. 2019.

[76] A. El-Amine, M. Iturralde, H. A. Haj Hassan, L. Nuaymi, "A Distributed Q-Learning Approach for Adaptive Sleep Modes in 5G Networks", 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, pp. 1–6, 2019.

[77] I. Al Qerm, B. Shihada, "Energy Efficient Traffic Offloading in Multi-Tier Heterogeneous 5G Networks Using Intuitive Online Reinforcement Learning", in IEEE Transactions on Green Communications and Networking, vol. 3, no. 3, pp. 691–702, Sep. 2019.

[78] C.-W. Huang, P.-C. Chen, "Mobile Traffic Offloading with Forecasting using Deep Reinforcement Learning", arXiv preprint arXiv:1911.07452, [Online]. Available: https://arxiv.org/abs/1911.07452.

[79] Y. Xiao, J. Zhang, Y. Ji, "Energy efficient Placement of Baseband Functions and Mobile Edge Computing in 5G Networks", 2018 Asia Communications and Photonics Conference (ACP), Hangzhou, pp. 1–3, 2018.

[80] N. Toumi et al., "On Using Physical programming for Multi-Domain SFC Placement with Limited Visibility", in IEEE Transactions on Cloud Computing, 2020.

[81] ETSI, "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; NFV-MANO procedures specification", NFV-SOL 016, V2.8.1, Aug. 2020.

[82] NIST, "Framework for Improving Critical Infrastructure Cybersecurity", National Institute of Standards and Technology, V1.1, Apr. 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/cswp/nist.cswp.04162018.pdf.

[83] UK NCSC, "NCSC CAF guidance", National Cyber Security Centre, V3.0, Sep. 2019. [Online]. Available: https://www.ncsc.gov.uk/collection/caf.

[84] ANSSI, "Managing Cybersecurity for Industrial Control Systems", Agence nationale de la sécurité des systèmes d'information (National Cybersecurity Agency of France), V1.0, Jun. 2012. [Online]. Available: https://www.ssi.gouv.fr/uploads/2014/01/Managing_Cybe_for_ICS_EN.pdf.

[85] NZ NCSC, "New Zealand Security Incident Management Guide for Computer Security Incident Response Teams", New Zealand National Cyber Security Centre – Government Communication Security Bureau, May 2013. [Online]. Available: https://www.ncsc.govt.nz/assets/NCSC-Documents/New-Zealand-Security-Incident-Management-Guide-for-Computer-Security-Incident-Response-Teams-CSIRTs.pdf.

[86] D. Rupprecht, K. Kohls, T. Holz, C. Pöpper, "Breaking LTE on Layer Two", 2019 IEEE Symposium on Security and Privacy (SP), pp. 1121–1136, 2019, doi: 10.1109/SP.2019.00006.

[87] 3GPP, "Management and orchestration; Provisioning", 3GPP TS 28.531, V17.1.0, Sep. 2021.

[88] 3GPP, "Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3", 3GPP TS 28.541, V17.4.0, Sep. 2021.

[89] N. J. Gunther, "A simple capacity model of massively parallel transaction systems", In: CMG National Conf. (1993) 1035-1044.

[90] H. Hamann, "Superlinear Scalability in Parallel Computing and Multi-Robot Systems: Shared Resources, Collaboration, and Network Topology," ARCS 2018, Lecture Notes in Computer Science, vol 10793, Springer, Cham, Apr. 2018.

[91] 3GPP, "Security architecture and procedures for 5G System", 3GPP TS 33.501, V17.3.0, Sep. 2021.

[92] M. Alkasassbeh, G. Al-Naymat, A. B. Hassanat, M. Almseidin, "Detecting distributed denial of service attacks using data mining techniques," International Journal of Advanced Computer Science & Applications, vol. 1, no. 7, pp. 436–445.

[93] A. Dobson, K. Roy, X. Yuan, J. Xu, "Performance Evaluation of Machine Learning Algorithms in Apache Spark for Intrusion Detection", 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, pp. 1-6, 2018, doi: 10.1109/ATNAC.2018.8615314.

[94] Y. Manzali, M. Chahhou, M. El Mohajir, "Impure Decision Trees for Auc and Log loss optimization", 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Fez, pp. 1–6, 2017, doi: 10.1109/WITS.2017.7934675.

[95] A. Dobson, K. Roy, X. Yuan, J. Xu, "Performance Evaluation of Machine Learning Algorithms in Apache Spark for Intrusion Detection", 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, 2018, pp. 1–6, doi: 10.1109/ATNAC.2018.8615314.

[96] S. Jha et al., "Deep Learning Approach for Software Maintainability Metrics Prediction", in IEEE Access, vol. 7, pp. 61840–61855, 2019, doi: 10.1109/ACCESS.2019.2913349.

[97] RMSLE, [Online]. Available: https://towardsdatascience.com/11-evaluation-metrics-data-scientists-should-be-familiar-with-lessons-from-a-high-rank-kagglers-8596f75e58a7#5185.

[98] R square, [Online]. Available: https://towardsdatascience.com/11-evaluation-metrics-data-scientists-should-be-familiar-with-lessons-from-a-high-rank-kagglers-8596f75e58a7#ad72.

[99] R square, [Online]. Available: https://towardsdatascience.com/the-enigma-of-adjusted-r-squared-57b01edac9f.

[100] S. Kalyanakrishnan, "The Perceptron Learning Algorithm and its Convergence". [Online]. Available: https://www.cse.iitb.ac.in/~shivaram/teaching/old/cs344+386- s2017/resources/classnote-1.pdf.

[101] R. Basri, D. Jacobs, Y. Kasten, S. Kritchman, "The Convergence Rate of Neural Networks for Learned Functions of Different Frequencies", arXiv preprint arXiv:1906.00425, 2019, [Online]. Available: https://arxiv.org/abs/1906.00425.

[102] Reinforcement learning, available: http://www.it.uu.se/research/systems_and_control/education/2017/relearn/lec3.pdf

[103] Reinforcement learning, available: https://en.wikipedia.org/wiki/Sample_complexity

[104] T. Wang et. al., "Benchmarking Model-Based Reinforcement Learning". [Online]. Available: https://www.cs.toronto.edu/~tingwuwang/mbrl.pdf.

[105] S. C. Y. Chan et al. "Measuring the Reliability of reinforcement Learning Algorithms", arXiv preprint arXiv:1912.05663, 2020, [Online]. Available: https://arxiv.org/abs/1912.05663.

[106] "Convergence of Reinforcement Learning Algorithms". [Online]. Available: https://towardsdatascience.com/convergence-of-reinforcement-learning-algorithms3d917f66b3b7.

[107] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu. "Federated Learning", Morgan & Claypool Publishers, Dec. 2019.

[108] A. N. Bhagoji, S. Chakraborty, P. Mittal, S. Calo, "Analyzing federated learning through an adversarial lens", in: International Conference on Machine Learning, The Proceedings of Machine Learning Research vol. 97, pp. 634–643, 2019.

Appendix I: Metrics for AI algorithms evaluation

a) Supervised and unsupervised learning algorithms metrics

Metrics	Meaning	Formula	Target
Confusion Matrix [92]	A confusion matrix is an N X N matrix, where N is the number of classes being predicted. Confusion Matrix gives us a matrix as output and describes the complete performance of the model. The correct predictions fall on the diagonal line of the matrix. The Confusion matrix in itself is not a performance measure as such, but almost all of the performance metrics are based on Confusion Matrix and the numbers inside it.	4 important terms in Confusion Matrix: - True Positives (TP): The cases in which we predicted YES and the actual output was also YES. - True Negatives (TN): The cases in which we predicted NO and the actual output was NO. - False Positives (FP): The cases in which we predicted YES and the actual output was NO. - False Negatives (FN): The cases in which we predicted NO and the actual output was YES.	
Accuracy [92]	Classification accuracy or Accuracy is the ratio of the number of correct predictions to the total number of input samples.	Accuracy = Number of correct predictions/Total number of predictions made	Accuracy > 90% (Close to 100%) Accuracy and other ML metrics values depend on a number of factors, including the number of data used during training and test phases. Accuracy should be close to 100%, (above 90% is not bad, especially in complicated models).
Precision [92]	It is the number of True Positive divided by the number of positive results predicted by the classifier. To minimizing False Positives, we would want our Precision to be as close to 100%	Precision = TP/(TP+FP)	Close to 100%
Recall/ Sensitivity [92]	It is the number of True Positives divided by the number of all relevant samples (all samples that should have been identified as positive). To minimizing False Negatives, we would want our Recall to be as close to 100%	Recall = (TP/TP+FN)	Close to 100%
F1 Score [93]	F1 Score is the Harmonic Mean between precision and recall. It tells how precise the classifier is (how many instances it classifies correctly), as well as how robust it is, (it does not miss a significant number of instances). The greater the F1 Score, the better is the performance of our model. Range [0, 1].	F1 Score = 2*(Precision*Recall)/(Precision+Recall)	Close to 100%

Metrics	Meaning	Formula	Target
Logarithmic Loss [94]	- When working with Log Loss, the classifier must assign a probability to each class for all the samples. - Log loss measures the UNCERTAINTY of the probabilities of the model by comparing them to the true labels and penalizing the false classifications. - Log loss is only defined for two or more labels. - Log Loss gradually declines as the predicted probability improves; thus, Log Loss nearer to 0 indicates higher accuracy, Log Loss away from 0 indicates lower accuracy. - Log Loss exists in the range (0, ∞].	Suppose there are N samples belonging to M classes, then the Log Loss is calculated as below: $Log\ loss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{i=1}^{M} y_{ij} * \log(y_{ij})$ y_{ij} indicates whether sample i belongs to class j or not	Log Loss nearer to 0 indicates higher accuracy, Log Loss away from 0 indicates lower accuracy.
ROC, AUC [94][95]	ROC can be broken down into sensitivity and specificity. Choosing the best model is sort of a balance between predicting 1's accurately or 0's accurately. In other words, sensitivity and specificity - True Positive Rate (Sensitivity/ Recall): True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. - False Positive Rate (Specificity): False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points	**Draw curve:** True positive rate VS. False positive rate True Positive Rate= TP/(FN+TP) False Positive =FP/(FP+TN)	
Mean Absolute Error (MAE) [96]	Average of the difference between the Original Values and the Predicted Values. Does not give any idea of the direction of the error i.e., whether we are under predicting the data or over predicting the data. Robust to outliers Range (0, + infinity]	$\mathbf{MAE} = \frac{1}{N}\sum_{i}^{N} \lvert yi - yi\ \bigvee \rvert$ yi **: Actual output value** $yi\bigvee$ **: Predicted output value**	The smaller the MAE, the better is the model. (Close to zero)
Mean Squared Error (MSE) [93]	Takes the average of the square of the difference between the original values and the predicted values. As it takes the square of the error, the effect of larger errors (sometimes outliers) become more pronounced than smaller error. Model will be penalized more for making predictions that differ greatly from the corresponding actual value. Before applying MSE, we must eliminate all nulls/infinites from the input. Not robust to outliers Range (0, + infinity]	$\boldsymbol{MSE} = \frac{1}{N}\sum_{i=1}^{n} (y_i - y_i')$ y_i: Actual output value y_i': Predicted output value	Close to zero

Metrics	Meaning	Formula	Target
Root Mean Squared Error (RMSE) [96]	Because MSE is squared, its units do not match that of the original output. RMSE is the square root of MSE. Since MSE and RMSE both square the residual, they are similarly affected by outliers. RMSE is analogous to the standard deviation and is a measure of how large the residuals are spread out. Generally, RMSE will be higher than or equal to MAE.	$\boldsymbol{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - y_i')^2}$ y_i: Actual output value y_i': Predicted output value	Close to zero
Root Mean Squared Logarithmic Error (RMSLE) [97]	- It takes the log of the predictions and actual values. - What changes are the variance that we are measuring. - RMSLE is usually used when we do not want to penalize huge differences in the predicted and the actual values when both predicted and actual values are huge numbers. - If both predicted and actual values are small: RMSE and RMSLE are same. - If either predicted or the actual value is big: RMSE > RMSLE If both predicted and actual values are big: RMSE > RMSLE (RMSLE becomes almost negligible)	$\boldsymbol{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(\log(y_i' + 1) - \log(y_i + 1))^2}$ y_i: Actual output value y_i': Predicted output value	Close to zero
R Square. [98][99]	- In the case of a classification problem, if the model has an accuracy of 0.8, we could gauge how good our model is against a random model, which has an accuracy of 0.5. So, the random model can be treated as a benchmark. But when we talk about the RMSE metrics, we do not have a benchmark to compare. - This is where we can use R-Squared metric. - Shows how well terms (data points) fit a curve or line.	$R^2 = 1 - \frac{\sum_{i}^{N}(yi - yi')^2}{\sum_{i}^{N}(yi - y')^2}$, where: $y' = \sum_{i}^{N} yi$ yi : Actual output value yi' : Predicted output value	$0 \le R^2 \le 1$ Higher R^2 (close to 1) is better, which means the prediction is closer to the data
Adjusted R Square [99]	Both R2 and the adjusted R2 give an idea of how many data points fall within the line of the regression equation. However, there is one main difference between R2 and the adjusted R2: R2 assumes that every single variable explains the variation in the dependent variable. **The adjusted R2** gives the percentage of variation explained by only the independent variables that affect the dependent variable.	$\text{Adj\_}R^2 = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$. where: n: is the number of points in your data sample. k: is the number of variables in the model, excluding the constant.	$0 \le Adj_R{}^2 \le 1$ Higher adjusted R square value is better (close to 1).
Training Convergence Time [100][101]	For the supervised learning this is a time when for the training certain percentage of the final accuracy of classification is achieved. The time can be expressed in terms of iterations or in time units.	Analytic formula is dependent on the type of supervised learning algorithm.	Properly define the model as well as set the algorithm parameters in such a way that the convergence time is shortened while successfully learn the target function, and maximum precision is achieved.

b) Reinforcement learning algorithms metrics

Metrics	Meaning	Target values
Reward vs. time steps [102]	Draw a curve of the obtained reward by taking the actions over time.	The reward should be the highest possible. The value of the reward depends on the model considered, it can be maximised by properly adjusting the parameters of the algorithm, as well as properly defining the training time (e.g., the number of epochs ...).
Regret [102]	Regret is a commonly used metric. At each time step, you take the difference between the reward of the optimal decision versus the decision your algorithm actually took. You then sum this up for cumulative regret. The minimum regret is 0. With the optimal policy, the smaller the regret the better an algorithm has performed. Defined as the difference between the cumulative reward of the optimal policy and that gathered by π.	Zero or the lowest possible (towards zero).
Sample complexity [102][103]	Defined as the time required to find an approximately optimal policy. Well defined for any kind of RL problems.	Properly define the model as well as set the algorithm parameters in such a way that the training time is shortened while successfully learn the target function, and maximum precision is achieved.
Time Efficiency (Wall-Clock) [104]	Average time between the start and end of training (Timer). Wall time is thus different from CPU time, which measures only the time during which the processor is actively working on a particular task. The difference between the two can arise from architecture and -dependent factors, e.g., programmed delays or waiting for system resources to become available. Consider the example of a mathematical program that reports that it has used “CPU time 0m0.04s, Wall time 6m6.01s”. This means that while the program was active for six minutes and six seconds, during that time, the computer’s processor spent only 4/100 of a second performing calculations for the program.	Define a time threshold according to the requirements of the task.
Real-Time Criteria [104]	Whether the algorithm is fast enough to run in real-time at test time, i.e., if the action selection can be done faster than the default time-step of the environment.	Compare the test time with -time. The closer they are to each other, the better it is.
Short-term Risk across Time (SRT) [105] Related to CVaR	This metric allows to evaluate the reliability of an RL algorithm and measure the most extreme short-term drop in performance over time. It will help us calculate the differences on a specific performance metric (e.g., obtained reward) of an RL algorithm from one evaluation point to another on each training run. We plan to use the distribution of these differences to define the worst-case expected drop in performance. CVaR: Conditional Value at Risk or CVaR is derived by taking a weighted average of the “extreme” losses in the tail of the distribution of possible outcomes.	Short-term Risk across Time should be zero or very low.
Algorithm convergence time [106]	This is a time when the algorithm behaviour can be seen as stable, i.e., the steady state of the algorithm has been achieved, it means that the controlled system KPIs are within specified bounds and no significant further improvement is noticed.	Properly define the model as well as set the algorithm parameters in such a way that the convergence time is shortened while successfully learn the target function, and maximum precision is achieved.

c) Federated learning algorithms metrics

Metrics	Meaning	Target values
delta-performance loss [107]	VFED – federated accuracy VSUM – conventional accuracy $\|V_{FED} - V_{SUM}\| < \delta$ Denote the performance measure (i.e., accuracy, recall, F1-score) of the centralised and federated models, respectively. The centralised model is the one trained on the centralised dataset (i.e., the dataset collected from all participants at the server level);	δ is a non-negative real number that should be very close to 0. We say the federated learning algorithm has δ-accuracy loss [107].
Stealth metrics [108]	Check if the updates will improve or reduce the global model.	Set thresholds corresponding to the considered stealth metric (Accuracy checking or Weight update statistics). These thresholds must be low enough to guarantee the performance of the considered mode [108].
Update Time Efficiency	The time needed to update the global model	The time needed to update the global model must be less than or equal to a time threshold corresponding to the system requirement (to be defined according to the case).