Call: H2020-ICT-2020-2

Project reference: 101015956

Project Name:

A flagship for B5G/6G vision and intelligent fabric of technology enablers connecting human, physical, and digital worlds

Hexa-X

Deliverable D6.2
Design of service management and orchestration functionalities

Date of delivery:	29/04/2022	Version:	1.1
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D6.2
Document Title:	Design of service management and orchestration functionalities
Editor(s):	Ignacio Labrador Pavón, Adrián Gallego Sánchez, Ricardo Marco Alaez (ATO)
Authors:	Ignacio Labrador Pavón, Adrián Gallego Sánchez, Ricardo Marco-Alaez (ATO), Giada Landi, Giacomo Bernini, Pietro Piscione, Elena Bucchianeri, Erin Seder (NXW), José Ordoñez-Lucena (TID), Bessem Sayadi, Sylvaine Kerboeuf (NOFR), Mohammad Asif Habibi (TUK), Cédric Morin, Cao-Thanh Phan (BCO), Sławomir Kukliński (ORA), Antonio Virdis (UPI), Christos Ntogkas, Ioannis Belikaidis, Konstantinos Kokkalis, Iason Bitchavas (WIN), Milan Groshev, Jorge Martín Pérez, Jesús Pérez Valero, Vittorio Prodomo, Pablo Serrano (UC3).
Contractual Date of Delivery:	30/04/2022
Dissemination level:	Public
Status:	Final
Version:	1.1
File Name:	Hexa-X_D6.2_v1.1.pdf

Revision History

Revision	Date	Issued by	Description
V1.0	13.04.2022	HEXA-X WP6	First Full Version
V1.1	26.04.2022	HEXA-X WP6	Quality check – ready for releasing

Abstract

This Deliverable D6.2 describes the architectural design of the novel management and orchestration (M&O) mechanisms for the Hexa-X project, which represents one of the main project milestones and outcomes. The M&O architectural design described here considers the outcomes from the previous Deliverable D6.1, which identified the main features and enablers for Beyond 6G services M&O. Aligned with the work being performed in WP1 (E2E architecture) and WP5 (architectural enablers for 6G), the new M&O architectural design in this document proposes an evolution of the management mechanisms in the former 5G architectures through the extensive adoption of the cloud-native principles, but adapted to the new generation of 6G telco-grade services. The document addresses hot topics in the state-of-the-art that are considered to have a great impact on the architectures of future 6G networks, such as the usage of AI/ML techniques applied to M&O, continuum orchestration including different network domains, extreme-edge orchestration, or intent based management, among others. In the context of the Hexa-X project, the document is intended to serve as a basis for the next Deliverable D6.3, where the final evaluation of the service management and orchestration mechanisms introduced here are planned to be reported. Beyond the Hexa-X project, the ambition is also to provide a M&O architectural design that can also serve as a reference for future 6G projects and related SDOs.

Keywords

B5G, 6G, M&O, MANO, OAM, management and orchestration, AI/ML-based service and network management, continuum management and orchestration, extreme-edge, cloud-native, SBMA, SBA, architectural design.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101015956.

Executive Summary

This report is the second deliverable of the Hexa-X Work Package 6 (WP6), and focuses on the design of service management and orchestration (M&O) functionalities for the sixth generation of mobile telecommunications networks (6G), addressing the M&O architectural design of the novel orchestration and management mechanisms for Hexa-X, which is one of the main project milestones (MS5). The deliverable reflects the work done in WP6 from month 7 (July 2021) until month 15 (April 2022), according to the project execution plan.

The architectural design is presented here by means of three architectural views: (i) *Structural View*, describing the main building blocks in the architecture; (ii) *Functional View*, describing how these blocks can interact each other to provide the relevant functionalities; and (iii) *Deployment View*, describing how the architecture could be deployed in practice.

In the Structural View the document provides an architectural design with a clear split between managed and managing objects. Managing objects are in turn split in two: those oriented to perform the primary (basic) M&O functions (i.e., fulfilment, assurance, and artifact management functions), and those in charge of what are considered other complementary M&O functions: Artificial Intelligence (AI)/Machine Learning (ML) functions, security functions and monitoring functions. All components in this Structural View are distributed into four layers: Service Layer (business oriented), Infrastructure Layer (with all the infrastructure resources), Network Layer (including all the necessary Network Functions -NFs-), and the Design Layer (transversal, to integrate the services development processes with the operational scope). The document describes how all the components in the different layers can communicate each other using a specific Application Programming Interface (API) Management Exposure, following the Service-Based Management Architecture (SBMA) model, in order to support the orchestration of a wide variety of service definitions and decompositions, including lightweight microservices as well as other more conventional virtual appliances in all layers.

The Functional View focuses on the main management and orchestration processes. Besides the basic management mechanisms (i.e., instantiation actions, scaling actions, etc.), this view focuses on describing more complex functionalities, such as the End to End (E2E) seamless integration processes (which introduces the Device-Edge-Continuum M&O concept involving the extreme-edge and other external resources), the programmable processes (including intent-based mechanisms, monitoring processes and software integration processes using DevOps-like workflows), automation processes (including different forms of automation), and data-driven processes (considering the security aspects and focusing on the usage of Artificial Intelligence AI/ML-based data-driven techniques, together with the main challenges associated to them). Other mechanisms such as optimised placement, resource optimisation and dynamic allocation have been also considered here.

The Deployment View describes the main building blocks that can be used for the deployment of an architecture like the one described in this document. The view describes how these building block components can be grouped together from small racks up to large-scale datacentres, and explains how to integrate the extreme-edge resources in the orchestration domain to implement the device-edge-cloud continuum management concept.

In order to avoid strong alignment with specific Standards Developing Organisations (SDO) or other standards, the architectural design presented in this report does not align with a specific SDO or any other standard. However, possible alignments with relevant State of the Art (SotA) standards have been also provided in the document to showcase how the architectural design presented here could be implemented based on certain relevant SotA standards. Besides this, the document also identifies relevant M&O-related Key Performance Indicators (KPIs) and their relationship with the Key Value Indicators (KVIs) identified for the Hexa-X project.

Table of Contents

List of Figures

List of Tables

1 Scope

This document describes the M&O architectural design provided for the Hexa-X project, which ambitions to develop key technology enablers for the future 6G telecommunication networks. This design is part of the work addressed in the Hexa-X Work Package (WP) 6, which specifically addresses intelligent orchestration and service management for 6G networks. The document also describes how the architectural design lines up with relevant state-of-the-art standards and with the Hexa-X use cases. It also identifies relevant KPIs and how they relate with the main Hexa-X KVIs.

2 Abbreviations

5G	Fifth generation of mobile telecommunications technology.
5GC	5G Core
5GPPP	5G Infrastructure Public Private Partnership
6G	Sixth generation of mobile telecommunications technology.
AC	Autonomic Computing
AI	Artificial Intelligence
AIaaS	AI as a Service
AM	Autonomic Manager
AMC	Autonomic Management and Control
AnLF	Analytics Logical Function
API	Application Programming Interface
B5G	Beyond 5G
BPP	Bin Packing Problem
BSS	Business Support System
CaaS	Container as a Service
CAPIF	Common API Framework
CD	Continuous Delivery
CFS	Customer Facing Service
CI	Continuous Integration
CISM	Container Infrastructure Service Manager
CN	Core Network
CNF	Containerized Network Function
CRUD	Create, Read, Update, Delete
CSMF	Communication Service Management Function
CSP	Communication Service Provider
CT	Continuous Testing
CU	Central Unit
DE	Decision-making-Element

DevOps	Development and Operations
DL	Distributed Learning
DU	Distributed Unit
E2E	End to End
ETSI	European Telecommunications Standards Institute
FA	Federated Agent
FL	Federated Learning
FM	Federation Manager
GAN	Generative Adversarial Network
gNB	Next-Generation NodeB
GST	Generalised Slice Template
HA	High Availability
IBN	Intent-based Networking
IETF	Internet Engineering Task Force
IM	Information Model
IoT	Internet of Things
IRTF	Internet Research Task Force
ITU	International Telecommunication Union's
K3s	Lightweight Kubernetes
K8s	Kubernetes
KPI	Key Performance Indicator
KQI	Key Quality Indicator
KVI	Key Value Indicator
LCM	Life-cycle Management
M&O	Management and Orchestration
MANO	Management and Orchestration
MDAF	Management Data Analytics Function
ML	Machine Learning
MLP	Multi-Layer Perceptron
MNO	Mobile Network Operator
MO	Managed Object
MTLF	Model Training Logical Function
NE	Network Element
NEST	NEtwork Slice Type
NF	Network Function
NFVI	Network Function Virtualisation Infrastructure
NFVO	Network Function Virtualisation Orchestrator

NPN	Non-Public Network
NRM	Network Resource Model
NS	Network Service
NSD	Network Service Descriptor
NSMF	Network Slice Management Function
NSS	Network Slice Subnet
NSSMF	Network Slice Subnet Management Function
NWDAF	Network Data Analytics Function
OAM	Operations Administration and Management
O-CU	O-RAN Central Unit
O-DU	O-RAN Distributed Unit
ONIX	Overlay Network for Information eXchange
O-RAN	Open RAN
OSS	Operational Support System
OSSM	OSS/BSS Security Manager
P2P	Peer-to-Peer
PID	Proportional-Integral-Derivative
PNF	Physical Network Function
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RCA	Root Cause Analysis
REST	Representational State Transfer
RFS	Resource Facing Service
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RU	Radio Unit
SBA	Service-Based Architecture
SBMA	Service-Based Management Architecture
SDK	Software Development Kit
SDN	Software Defined Networks
SDO	Standards Developing Organisation
SLA	Service Level Agreement
SM	Security Manager
SotA	State of the Art
TIP	Telecom Infra Project
TMN	Telecommunications Management Network

TN	Transport Network
TSG SA3	Technical Specification Group Service and System Aspects Working Group 3
UPF	User Plane Function
uRLLC	ultra–Reliable Low Latency Communication
vCU	Virtual Central Unit
vDU	Virtual Distributed Unit
VIM	Virtual Infrastructure Manager
VNF	Virtualised Network Function
VNFM	VNF Manager
WP	Work Package
XAI	eXplainable AI
ZSM	Zero-Touch Service Management

3 Overview

3.1 Objective of the document

The objectives of this document are fully aligned with the main objectives defined for WP6, i.e.:

- To serve as the mean for verifying one of the main milestones in the project (identified as milestone 5), which requires providing the architectural design of the novel M&O mechanisms in Hexa-X.
- To fulfil the previous considering disruptive trends and technologies, based on the gap analysis performed in the previous deliverable D6.1 [HX21-D61].
- To describe how the architectural design also includes the necessary means for automation and network programmability of 6G infrastructures.
- To describe how the M&O architecture can provide intent-based mechanisms for elaborating on requirements, diagnosing the performance of networks and services, modelling/abstracting services/networks, or implementing corrective actions through CI/CD.
- To describe how the architectural design can support orchestration of a wide variety of service definitions, composed of functions deployed in different cloud execution environments, including (traditional) virtual appliances, microservices and containers, and serverless functions in all domains.
- To describe how cognitive service management and orchestration mechanisms would be performed, based on the context-aware placement, effortless portability (migration) and dynamic resource allocation of constituent NFs.
- To describe how data-driven management practices would be implemented across the device-edge-cloud continuum, with data collection and aggregation solutions ensuring the execution of distributed yet consistent closed-loops.
- To serve as a reference for other work packages in the project, and for the next planned Deliverable D6.3 in this WP6 regarding the final evaluation of the service management and orchestration mechanisms.

3.2 Methodology

Several means have been used to complete the definition of the M&O architectural design provided in this deliverable. Mainly, by interacting with other technical WPs in the project, specially WP1 (with focus on the E2E architecture design and the security aspects), WP4 (with focus on the AI-related methods and algorithms) and WP5 (with focus on the main architectural enablers for 6G). Also, by accessing to previous deliverables from other projects and research initiatives, and of course, to a diversity of publications and journal articles (see Section 13). Special attention has also been paid to the possible alignments of the architectural design with multiple Standards Developing Organisations (SDOs) and other standards (Section 9).

To address the work, the focus was first on the main building blocks that should be part of the architecture, considering the innovations that should be provided, ordering the pieces together to build out a coherent fabric of capabilities that allow meeting Hexa-X requirements and bridging gaps identified in Deliverable D6.1 [HX21-D61]. This view was also shared with the rest of technical Hexa-X WPs, which provided feedback to ensure the architecture covers all capabilities needed for their in-scope technologies. Once the main building blocks were identified the focus was also on how these blocks could interact each other, and how they could be deployed in practice.

The main features and enablers for designing the M&O architecture described here were mainly extracted from the previous Deliverable D6.1. The “View on 5G Architecture” whitepaper from the 5G Infrastructure Public Private Partnership (5GPPP) Architecture Working Group [5gp21]

has been also used as a benchmark. The analysis of these and other documents (those in Sec. 13), together with the information coming from the discussions among the different involved partners, were the main information sources to generate the architectural design described in this deliverable.

The outcome from this work has been the addressing of the architectural design itself focusing on three different aspects: The "structural" aspect (concerning the main building blocks in the architecture), the "functional" aspect (concerning the main functionalities those blocks could provide), and the "deployment" aspect (concerning how the architectural design would be deployed in practice). On that basis, the description of the architecture has been organised into the three architectural views described above, namely, "structural", "functional" and "deployment" views. The intention is to provide a fairly complete description of the overall M&O architecture while focusing on specific aspects stressed in the individual views.

3.3 Structure of the document

The document is structured as follows:

Section 1 describes the overall scope of the document.

Section 2 includes a list of abbreviations to support the reading process.

Section 3 describes the main objectives of the document, the methodology used for addressing the M&O architectural design, and the structure of the document (this section).

Section 4 introduces Hexa-X WP6 main work topic from a general perspective, i.e., services M&O in mobile telecommunications networks.

Section 5 presents an overall description Hexa-X M&O system architecture. First, it presents an actor-role model to illustrate governance on Hexa-X operation and analyses the functional and non-functional requirements that this model imposes to the design of the system architecture. Based on this analysis, this chapter then outlines the architectural design, mapping these requirements into capabilities that will be accommodated through the different architectural views. These views will be described in the following chapters.

Section 6 is devoted to the Hexa-X M&O Structural View, which identifies the collection of management services building out the M&O architecture, and the Management Functions producing/consuming them. These management services (management objects) allow manipulating and operating different resource types (managed objects), according to the semantics and relationships captured in a well-defined Information Model (IM).

Section 7 describes the Hexa-X M&O Functional View. It describes the basic orchestration actions and more complex orchestration mechanisms (e.g., intent-based mechanisms, data-driven orchestration mechanisms, among others) that can be performed through the management services/functions introduced in Section 6.

Section 8 describes the Hexa-X M&O Deployment View, which explains how the architectural design would be deployed in practice.

Section 9 is devoted to analyse the alignment of the M&O architectural design with relevant standards.

Section 10 reports the most relevant KPIs that have been identified for the M&O architecture, and their relationship with the Hexa-X KVIs.

Section 11 describes the alignment with the initial set of use cases that have been identified for Hexa-X, respectively.

Finally, Conclusions are provided in Chapter 12.

4 Services Management and Orchestration

Services M&O is the central point of interest for WP6 in Hexa-X, and therefore, for this document. However, the "management and orchestration" topic is sometimes used in a very general and ambiguous manner. This section tries to define the use of this concept for the present document, and in general, for the Hexa-X scope.

A well-known reference used for a long time that can help to clarify on this is the International Telecommunication Union's (ITU) Telecommunications Management Network (TMN) reference model [3010]. Although it could seem a bit outdated, it is considered the key concepts regarding network management in this model are still valid. To deal with the complexity of telecommunications management the model considers the management functionality split into five logical layers:

- Network Elements Layer, representing the actual network devices. When the TMN reference model was written this just represented physical network elements. However, although a new generation of virtual and cloud-native elements is available now, they can still be considered Network Elements.
- Element Management Layer, which manages each network element on an individual or group basis (e.g., by applying a software update or configuration change on a specific device). Again, although this was defined for physical devices, this can be also applied to virtual network elements. Even if a network element is cloud-native, it still needs to be managed, so this fits into the TMN model as well.
- Network Management Layer. This layer manages connectivity across a whole network comprising a variety of network elements. To this layer belong those functions addressing the management of a wide geographical area. Typically, complete visibility of the whole network is required here. Here is where the ***orchestration*** concept comes into play, referring to the management of a high diversity of network elements that can be distributed on multiple network domains (e.g., Transport Network (TN), local breakout, radio, and core domains).
- Service Management Layer. This is concerned with the contractual aspects of services provided to customers. At the time the TMN reference was written, the term "customer" referred mainly to the Mobile Network Operator (MNO) end-users. Today it is necessary to consider also other customers, such as vertical industries, hyperscalers, large-scale content providers and application developers, among others. Service Management typically requires provision/cessation of services, Quality of Service (QoS)/Quality of Experience (QoE) fulfilment, or fault reporting, among others. This layer builds on the Network Management Layer, from one or more administrative domains. In fact, for those cases where the footprint and application span beyond the boundaries of one MNO (e.g., as it occurs in global coverage services or public-private network infrastructure services), it is needed to populate the service management layer with necessary capabilities to aggregate/compose orchestration resources from different Network Management Layer instances.
- Business Management Layer. It represents the overall management of the MNO customer, supporting the decision-making process for the optimal investment and usage of human and technical resources.

Since services M&O is the central point of interest for WP6 in Hexa-X, the architectural design presented here targets the Service Management Layer which, as said, consumes orchestration resources from the Network Layer, and provides additional orchestration capabilities to allow the management of services running atop, in an E2E manner, across domains.

Services M&O system is responsible for keeping communication services, digital services, and network slice services, including operations associated to their constituent network service (NS). Examples of these operations include NS onboarding and instantiation, growing or reducing their capacity, updating their configuration, or requesting their termination in a consistent and orderly manner. In our case, this would be done by exploiting all the available infrastructure resources envisaged for future 6G networks. In other words, not only core or edge network resources within

the MNO administrative domain will be considered, but also resources at the extreme-edge network (vertical or end-user resources) and resources from 3rd party facilities (e.g., in-factory networks, transportation hubs, or hyperscaler cloud nodes). The main enablers to make this high degree of orchestration possible (automation, programmability, usage of AI/ML techniques, Intent-Based Networking -IBN-, etc.) will be presented through the document.

5 Hexa-X M&O Architecture Insights

5.1 Stakeholders in the 6G ecosystem

The stakeholders involved in the 6G ecosystem are envisaged to be quite similar to the ones already in 5G, defined in [5gp21]. However, some new entities may successfully enter the business chain covering some or part of the existing roles, but with specialised types of offers. For example, in the category of the software suppliers, the developers of AI/ML solutions for intelligent network optimisation could become more and more relevant. In the category of Network Operators, leveraging the integration of Non-Terrestrial Networks in 6G infrastructures, Satellite Operators or Virtual Satellite Operators can be also involved in the whole value chain. In fact, they can provide an alternative network access, extend the coverage towards remote areas or provide backup connectivity for the backhaul/midhaul segment to guarantee the reliability of the E2E connection. Moreover, Public and Non-Public Network (NPN) Operators can be present and interact in some scenarios. Figure 5-1 represents this updated approach, with the new entities highlighted.

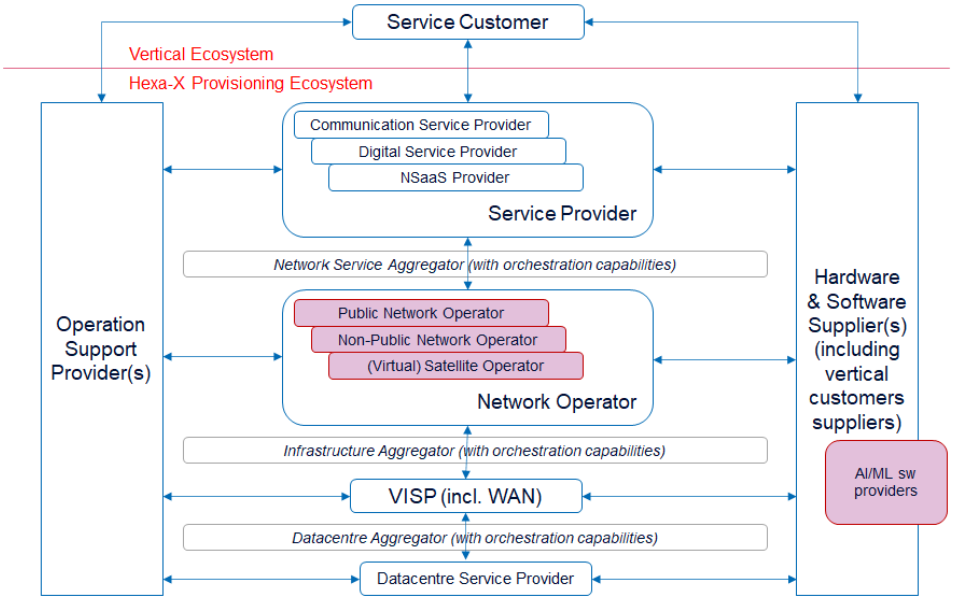

Figure 5-1. Stakeholders in the 6G ecosystem.

5.2 Requirements

In this section the main requirements for the Hexa-X M&O architectural design are provided. They are split in functional and non-functional requirements in sections 5.2.1 and 5.2.2 below, in the form of different lists. All these requirements come from the Goal State defined in the previous Deliverable D6.1 [HX21-D61] and the overall Hexa-X objectives in the project GA. The objectives will be addressed through the M&O architectural design itself, described in this document.

5.2.1 Non-functional Requirements

The main non-functional requirements for the M&O system are those derived from the Goal State described in Deliverable D6.1 [HX21-D61] (see Section 3). We summarise them here:

1. Regarding its implementation, the system shall be designed as a Distributed Platform.
2. Every segment of the network shall be operated following the cloud native principles.
3. Development and Operations (DevOps) techniques (e.g., CI/CD workflows) shall be also available, in order to enable the dynamic building, testing, merging, delivery and deployment of network/service management solutions, in an automated and secure way.
4. The M&O system shall provide interfaces to interwork with other external orchestration frameworks, access the infrastructure capabilities, and enforce decisions by leveraging on full network programmability.
5. The M&O system shall be enabled with an advanced monitoring system able to collect, aggregate and dispatch data from all managed network segments, integrating infrastructure, user plane and control plane related data from different sources, following model-based telemetry solutions.
6. The system shall provide means for automation and network programmability, aimed to facilitate the dynamic adaptation to changing network situations and requirements for utmost efficient use of resources.
7. The M&O system shall be enabled with security functions by design, providing accountability for security risk mitigation, forensic analysis and threat prevention.
8. The M&O system shall be enabled with AI/ML capabilities in order to manage complexity associated to the M&O processes.
9. The system shall be enabled with eXplainable AI (XAI) algorithms in order to avoid the orchestration function becoming a sort of black box able to perform actions on the network that could not be interpreted by humans.
10. The system shall be enabled with intent-based mechanisms for supporting regular operational tasks from both: MNO and Verticals.

Besides the previous requirements, additional non-functional requirements have been identified considering the KPIs also in the previous Deliverable D6.1 [HX21-D61]:

a) The M&O system shall be able to orchestrate seemingly infinite capacity, being able to orchestrate a seemingly infinite number of resources/devices. More specifically, the M&O system shall be able to scale to support high number of devices (>100 bn).
b) The M&O system shall be able to provide zero-perceived latency regarding service instantiation time.
c) The M&O system shall be able to contribute to reduce energy consumption and CO_2 emissions based on optimised resource management, at both provisioning time (minimise energy consumption when allocating workloads on infrastructure nodes) and operation time (granular, selective data collection for assurance and AI/ML training activities, minimising the movement of data across the system).
d) The M&O system shall be enabled to contribute to provide high efficiency in terms of QoE/service and cost, and societal goals, such as sustainability.

Besides requirements from [HX21-D61], other specific requirements have been included to be aligned with the overall Hexa-X project objectives. They are the following:

i. In predictive management scenarios, network reconfiguration (creation, composition and scaling times) shall be performed by (>10%) of the prediction horizon.
ii. The system shall exhibit an improvement by (>90%) in time to onboard new resources from other domains and manage the addition/removal of elements in/from the network.
iii. Compared to the former 5G networks, the system shall enable increasing service continuity by reducing downtimes by (>80%).
iv. The system shall be able to increase network energy efficiency by (>50%) applying predictive orchestration techniques.

Finally, some other non-functional requirements were also identified after releasing [HX21-D61]. They are the following:

I. The Hexa-X M&O architectural design shall be able to integrate solutions from multiple technology providers, e.g., different software vendors, hyperscalers, or solution integrators, avoiding vendor lock-in.
II. The Hexa-X M&O architectural design shall avoid strong alignment with a specific M&O-related standard (or a few of them). On the contrary, it shall be flexible and abstract enough to allow implementations based on multiple SDO's or other standards.
III. The M&O architectural design shall be aligned with the more general Hexa-X E2E Architectural Design addressed in WP1 (E2E Vision, Architecture and system aspects), and the 6G architectural enablers defined in WP5.
IV. The M&O architectural design shall provide a controllable capability exposure, enabling the communication among the different M&O resources within the MNO scope, as well as with resources in external administrative domains.

5.2.2 Functional Requirements

Besides those overall functional requirements typically expected from M&O systems (e.g., regarding provisioning, performance management or fault management), the following more specific functional requirements have been identified, also based on the Goal State defined in [HX21-D61]:

1. The M&O system shall provide continuous orchestration i.e., the orchestration function should cover the *continuum*, from the end devices (extreme-edge) through the edge, up to the central cloud with all the tiered cloud nodes in between. This requires performing service management operations on a high heterogeneity of devices, as well as blurring the borders through the different network and administrative domains from the M&O perspective.
2. The system shall support enhanced network abstractions/models and service description models.
3. M&O processes shall be able to support a wide variety of service definitions and decompositions based on modern lightweight microservice-based functions (e.g., containers), but also considering other physical and virtual appliances, in all domains.
4. The system shall provide multi-stakeholder orchestration. This means the M&O system shall be able to process requests from different actors (see Section 5.1), providing a unified architectural approach where they all can coexist and take part in the service value chain.
5. The system shall use AI/ML-driven orchestration functionalities to:
    - Enhance service M&O operations, such as the provisioning of NFs, including resource allocation and application layer configuration.
    - Automating network tasks, supporting data-driven and zero-touch approaches.
    - Provide cross-layer predictive orchestration.
    - Support proactive and dynamic self-optimisation of network slices.
    - Support the management of collaborative AI components across the network.
    - Support intent-based management processes by providing intelligence for reasoning regarding service requirements, network capabilities and external non-network factors. This includes high-level intent-based means for expressing application/service requirements, as well as for realising interactive human-machine interface methods.
    - Interpret and enforce sustainability policies, in order to reduce cost and energy consumption, as well as improving the service efficiency of the network infrastructure.

5.3 Novel Capabilities regarding M&O

Considering the requirements described in Section 5.2 the following main novel capabilities have been identified for the future 6G M&O systems:

- **Unified orchestration across the "extreme-edge, edge, core" continuum.** The end devices contribute to provide resources which can be leveraged to deploy 6G services through the M&O system. The nature of these extreme-edge nodes may require the adoption of resource orchestration mechanisms specialised to deal with their particular constraints, e.g., in terms of lower power capability, limited available resources that are also usually shared with user-controlled applications, volatile behaviour, and mobility patterns. This novel capability is addressed in Section 7.2.1.1 (Device-Edge-Cloud continuum management and orchestration).
- **Unified management and orchestration across multiple domains, owned and administered by different stakeholders,** characterised by heterogeneous technologies and potentially deploying specialised tools, platform and management systems with their own different interfaces. This feature involves the definition of converging interfaces, mechanisms to dynamically register and expose the resources and capabilities offered by each domain as well as access control procedures to regulate the consumption of the various primitives and services. At the multi-domain coordination level, the M&O system will combine hierarchical and peer-to-peer (P2P) federation strategies. This capability is specifically addressed in Sections 7.2.1.3 (Integration with other networks) and 7.2.3.4 (Automation in multi-stakeholder scenarios).
- **Increasing levels of automation** in the functionalities of service and network planning, design, provisioning, optimisation, operation and control, leveraging closed-loop and zero-touch solutions that strongly reduce the required manual intervention. Starting from a continuous monitoring of different aspects of network and service performances, the M&O system becomes able to automatically identify, detect or predict potential issues, failures, bottlenecks or inefficient configurations, triggering and coordinating dynamic reactions in the short or medium terms. This will be also enabled through the extreme programmability of network and computing resources. This capability is specifically addressed in Section 7.2.3 (Automation Processes).
- **Adoption of data-driven and AI/ML techniques in the M&O system**, supporting frameworks for distributed and collaborative AI, AI/MLOps, pervasive monitoring of service and network KPIs, with support for scalable data and trained models sharing along the "extreme-edge, edge and core" continuum in multi-stakeholder environments. The scope of AI/ML techniques will cover several optimisation aspects and lifecycle actions regarding the services M&O, including resource allocation and slice sharing at provisioning time, service composition, scaling, migration, re-configuration and re-optimisation of NFs and resources.
- **Intent-based approaches for service planning and definition**. The M&O system will implement automated mechanisms for translating service specifications and commands based on intents, which could be expressed even in natural language. This capability is specifically addressed in Section 7.2.2.1 (Intent-based means for expressing application/service requirements).
- **Adoption of the cloud-native principles in the telco-grade environment.** This involves three aspects: (i) the usage of micro-services for implementing NFs, i.e., lightweight self-contained, independent, and reusable components from different sources and suppliers (e.g., software containers), (ii) the implementation of a service mesh, which involves optimising the communication between applications and reducing downtimes using an specific built-in infrastructure layer, that facilitates fulfilling the extreme service requirements expected for 6G networks, and (iii) the enabling mechanisms for the NSs to be deployed and updated using DevOps practices, by implementing continuous integration and continuous delivery (CI/CD) pipelines with a very high automation degree. This third aspect is probably the most innovative in a telco-grade environment, involving the fact of putting together development and operational teams, which can be challenging in this scope, where services development is typically a collaborative effort involving multiple vendors. This capability is addressed in different sections in the document: the Structural View described in Section 6 relies on this concept itself; also Section 6.2.4, which describes the Design System based on the DevOps principles.

5.4 Baseline Architecture

To address the Hexa-X M&O architectural design considering the previous requirements and novel capabilities, the former 5G network architecture design has been used as a benchmark. Specifically, the 5G Architectural View from the 5G-PPP Architecture Working Group [5gp21] was used as the baseline (see Figure 5-2).

This architecture relies on the most recent advances in the softwarisation of the mobile network ecosystem, as well as relevant standards for access, core, management, and orchestration. As it can be seen, it comprises three main domains: one for verticals, other associated to the network itself, and another one corresponding to the infrastructure.

The Service Domain for Verticals includes all architectural innovations that help to include the business-related considerations to the offered services. The Network Domain represents the different operated NFs (e.g., access functions, core functions and M&O functions). The Infrastructure Domain considers specific innovations, such as the integration of 3rd party nodes (e.g., factories and transportation hubs), or drone-based access, among others.

The architecture also supports network automation, which is achieved through two main control loops: the first one at the service domain level (vertical control loop), which steers the behaviour of the network; the second one, within the network domain (operator control loop), with specific functions related to network and management data analytics.

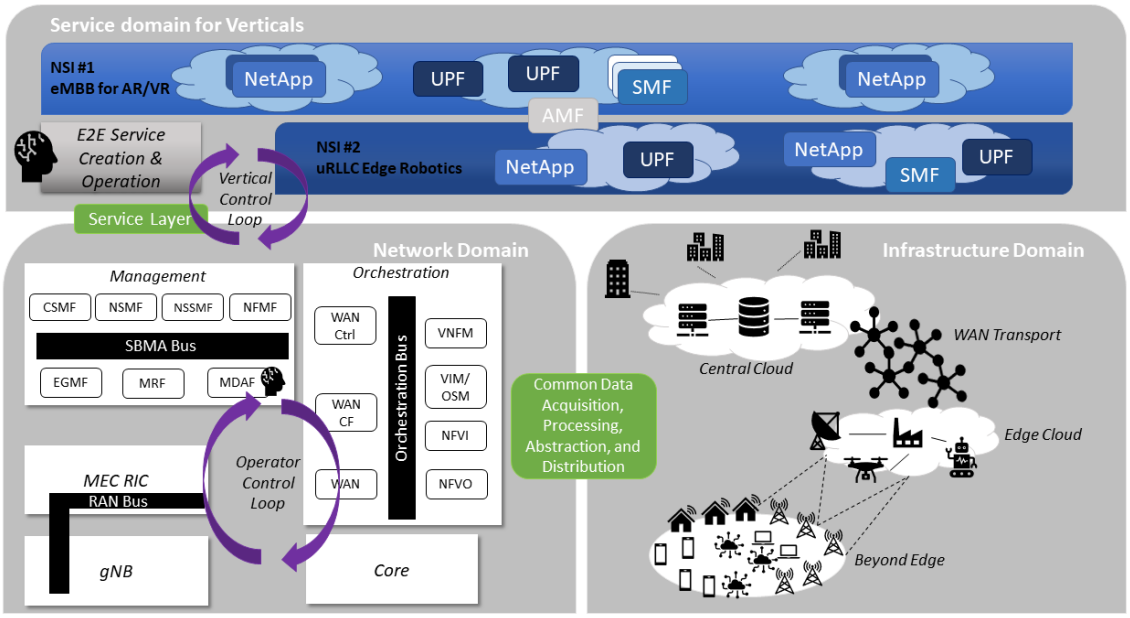

Figure 5-2. View on 5G Architecture. 5GPPP Architecture Working Group [5gp21].

The following chapters describe how the M&O architectural design presented in this document builds on top of this baseline architecture to include the envisaged innovations for the future 6G networks.

5.5 Architecture Description

The High-Level Hexa-X M&O Architectural Design in this document will be described by means of different "architectural views". This is a common practice when it comes to describing the architecture of complex systems [Kru95] [CAA09] [CBB+10]. The intention is that these views can together provide a coherent and complete description of the system's architecture, with each one conveying relevant information about different and more specific aspects.

Three different views have been chosen, focusing on providing an overall high-level description of the Hexa-X M&O architectural design: Functional View, Structural View and Deployment View (see Figure 5-3).

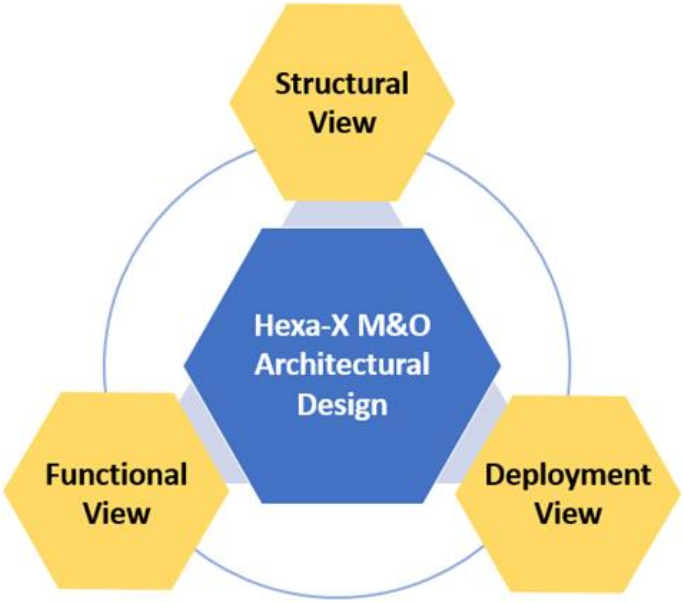

Figure 5-3. Hexa-X M&O Design. Architectural Views.

The Structural View (described in Section 6) will present the main building blocks to make up the system, and the interfaces that make communication among them possible. The Functional View (Section 7) will describe system behaviours, focusing on what are considered the most relevant functionalities that could emerge from this architecture, illustrating how the different functional blocks presented in the previous Structural View can interact each other to provide the different functions. Finally, the Deployment View (Section 8) will describe how the components in the Structural View could be deployed in practice, considering infrastructure resources and topological aspects.

The objective is that these three views, together, can provide a high-level yet complete description of what a 6G network M&O system could look like. Notice that none of these views address specific implementation details. This is intentional: since the development of the 6G technology is now taking its first steps, it is considered that, at this early stage, it is more important to provide a general (although complete) overview that can be easily generalised, instead of dwelling on specific implementation details that could become quickly outdated. However, information on possible alignment with relevant State-of-the-Art (SotA) standards is also provided in Section 9.

6 Structural View

Figure 6-1 represents the Structural View of the Hexa-X M&O architectural design. This view is intended to represent the main architectural building blocks. Associated functionalities and deployment details will be later provided in the specific Functional and Deployment views (in sections 7 and 8 respectively).

In general terms what this figure represents is that NSs and slices at the Service Layer are of course executed on the network elements (physical or virtual) at the Infrastructure Layer, being "made of" the network functions represented at the Network Layer. All those elements (network functions, services and slices) are designed and provided from the Design Layer.

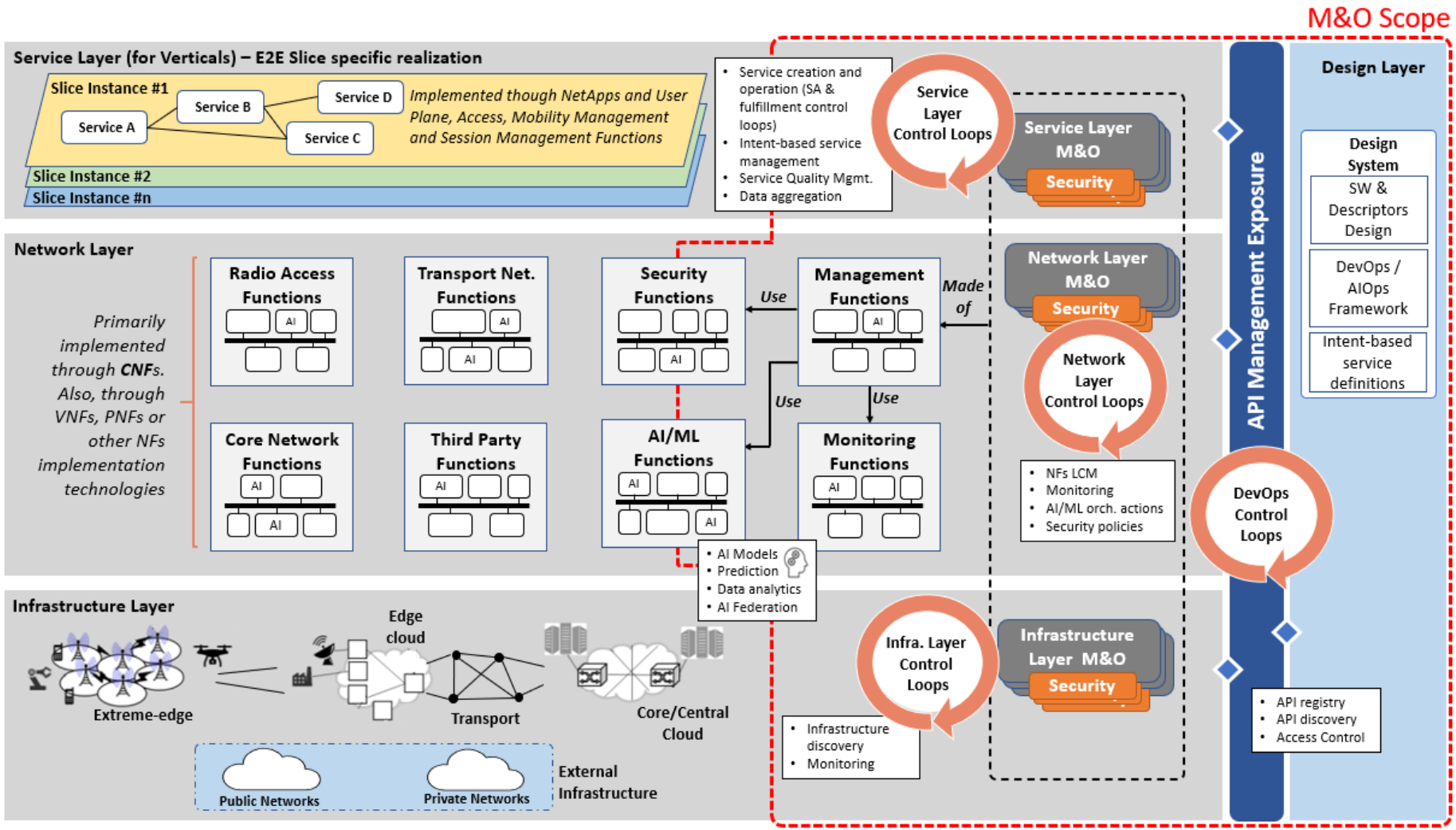

Figure 6-1. Structural View.

As it can be appreciated, this Structural View is inspired in the 5G baseline architecture mentioned in Section 5.4, but a simple glance already informs us of some of the main innovations that have been introduced, e.g.:

- There is a clear separation of concerns regarding the M&O resources (i.e., what is in charge of managing – see right-hand side of Figure 6-1) and the managed resources (i.e., what is managed, i.e., the left-hand side of Figure 6-1). This is aligned with the OSI management protocol [9595:98] [9596-1:98], which clearly separates these two types of software artifacts. On one hand, managed resources represent classes describing the properties of Hexa-X resources (e.g., NFs, network slices) that can be commissioned, operated and de-commissioned. The relationships existing across the managed resources are captured in a class UML diagram, called Information Model (IM). Instances of managed resources (classes) are referred to Managed Objects (MOs). On the other hand, M&O resources offer the management capabilities (e.g., provisioning, monitoring, etc.) that are used to act on the Managed Objects.
- A new layer, namely the Design Layer, has been included to represent the M&O-related operations involving third-party software providers.
- Hyperscalers, private networks, and extreme-edge have been included as part of the Infrastructure Layer[1].
- New control loops are also included:
  - The "DevOps Control Loop", which represents the automated continuous iterations (e.g., CI/CD iterations) between the MNO scope (grey colour) and the external Design Layer (light blue colour).
  - The "Infrastructure Control Loop", which automates the infrastructure discovery processes and associated monitoring. This is targeting the extreme-edge resources integration, that may have a high level of asynchrony that could require special processes in terms of management.
- Functions are associated in different groups at the Network Layer (e.g., Radio Access Functions, Core Network – CN– Functions, M&O Functions, AI/ML Functions, etc.). These functions would be primarily implemented through Containerised NFs (CNFs), but also through Virtualised NFs (VNFs), Physical NFs (PNFs), or other NFs implementation technologies (e.g., to ensure backward compatibility). As it can be seen, some functions work only as managed resources (e.g., CN functions or third-party functions) while others are specific M&O resources (e.g., the Monitoring Functions or the Management Functions themselves). Other functions are *hybrid*: they can support M&O resources (e.g., certain AI/ML- or security-related functions) or work as *pure* managed resources (e.g., certain AI as a Service –AiaaS– functions or security functions not directly involved in M&O processes).
- Functions in the Network Layer are generic, i.e., instead of referring specific functions as in Figure 5-2 (e.g., CSMF, MRF, NFVO, etc.), just generic blocks are provided. This is intentional, to avoid explicit alignment with a specific SotA standard, and to consider other possible functions that would be probably defined for the future 6G stack.
- A new set of AI collaborative components have been distributed across the network (this is aligned with the work being performed in WP4).
- M&O functions can be instantiated in the three different layers (service, infrastructure and network layers). They also include specific security-related functions.
- A new cross-layer Application Programming Interface (API) Management Exposure block has been included to communicate the different network elements in the different network layers.

[1] Although the 5G baseline architectural design already represents the "beyond the edge" domain (which could be understood as the "extreme-edge" mentioned throughout this document), this is never referenced in the 5G-PPP Architecture Working Group whitepaper [5gp21].

Even taking into account these main innovations, this Hexa-X M&O architecture is considered as an evolution, built on the same overall principles as the previous 5G architecture. One of those most remarkable common principles is the adoption of the Service Based Management Architecture (SBMA) model, already introduced in [28.533] and [zsm-002]. This model still represents a paradigm shift on the telco stack design, based on moving from traditional network/service management systems (hard-to-evolve, and with siloed managers connected with point-to-point protocol interfaces) to a cloud-native management system (built out of modular composable management services that are offered for consumption using HTTP-based RESTful APIs). Based on this SBMA model, it is possible to have a collection of management services, each representing a particular management capability (e.g., provisioning, performance assurance, trace control) that allows manipulating a particular resource (e.g., network slice, CN function, etc.). Management services are produced and consumed by management functions, which are mappable to vendor solutions.

The following sections describe in more detail the main innovations introduced here: First, Section 6.1, introduces the Managed Objects. Then, Section 6.2 will explain the M&O resources, i.e., those assets in charge of managing the MOs.

6.1 Managed Objects

As it can be seen in Figure 6-1, there is a clear separation between the M&O Scope and the MOs. MOs represent common resources, such as NFs, NSs or Network Slices that can be managed. In Figure 6-1 MOs are all those outside the dashed red line, i.e., those elements that can be managed by the Managing Objects inside the red dashed line. Figure 6-2 pictures a class diagram with all the MOs identified in the Hexa-X M&O architectural design, representing the hierarchical relationships with the generic 3GPP *Subnetwork* construction [28.622].

Taking as a starting point the 3GPP Slicing model [28.541], the Hexa-X project proposes to extend such model with three main innovations:

1. Including the *External Facing Subnetwork* object.
2. Including the *NetApp* construction[2].
3. Up scoping NF construction to *CNFs.*

These three items are represented in Figure 6-2. Further details on them are captured below.

External Facing Subnetwork

The *External Facing Subnetwork* construction (on top) builds upon the classical definition of the subnetwork concept, extending it for the "network of networks" support. This paradigm is one of the main topics associated to the network evolution and expansion towards 6G [HX21-D51].

The subnetwork is the baseline representation of a management domain. This construction groups managed objects (i) such that the group represents a topological structure which describes the potential for connectivity, (ii) with common characteristics, and (iii) subject to common administration.

As represented in Figure 6-3, the subnetwork object serves as a root artifact for the definition of other service-oriented construction such as Network Slices (NSs), and NS Subnets (NSSs). In fact, all these objects include attributes inherited from the subnetwork. It can also be defined recursively patterns, in the sense that a (parent) subnetwork can be built out of other (child) subnetworks, by applying appropriate pointer mechanisms in the IM tree. The number of

[2] Although NetApps are already considered in 5GPPP, they are still not in the standard, nor in the 3GPPP information model (or how to integrate NetApps in the current information model).

subnetworks and their parent-child dependencies is subject to the MNO decision, according to the distribution and compartmentalisation of assets in the M&O stack.

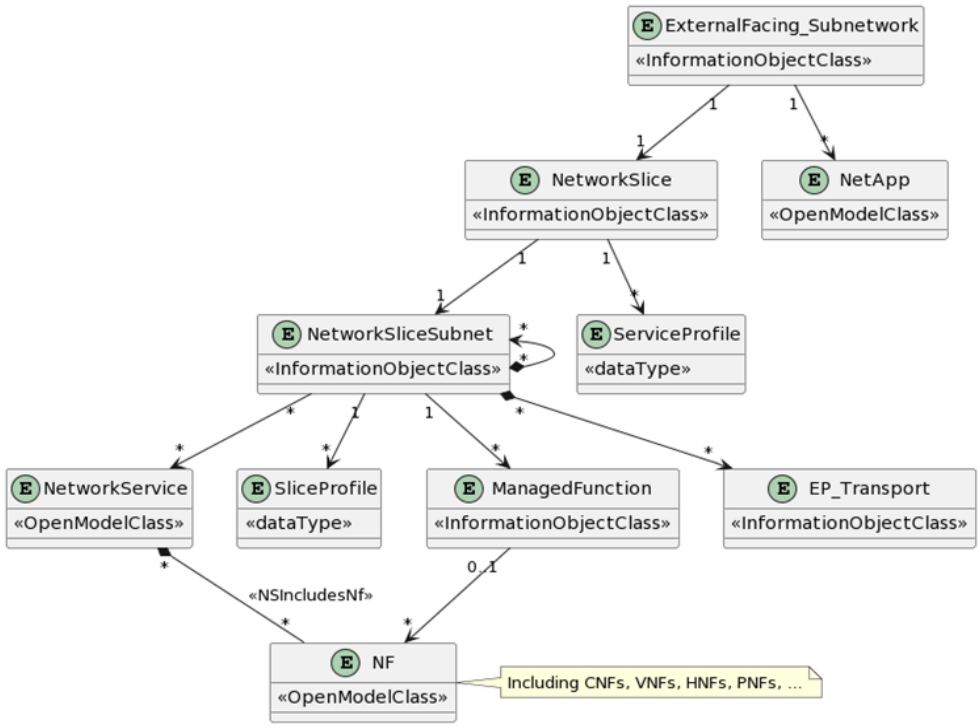

Figure 6-2. Information model hierarchies.

To realize the "network of networks" paradigm, the *External Facing Subnetwork* construction is defined. This construction captures the set of properties of a subnetwork that are allowed to be made available to third party consumers. In other words, it represents the view of a subnetwork that a MNO, when becoming a Communication Service Provider (CSP), provides to external administrative domains. The specification of this construction is as follows:

- *External Facing Subnetwork* results from filtering attributes of a subnetwork.
- One CSP may decide to expose one or more *External Facing Subnetworks*.

For the provisioning of advanced services in the "network of networks" paradigm (e.g., public-private networks with multiple providers in the value chain), it is required to combine subnetworks from different administrative domains. To allow for the stitching/chaining of these subnetworks, the *External Facing Subnetwork* constructions need to be appended in the IM tree of the stakeholder responsible for the E2E service management.

NetApps

The NetApp construction proposed in Figure 6-2 also inherits from the *External Facing Subnetwork* object for providing "network of networks" capabilities. NetApps (Network Applications) can be defined as assets where NFs may be chained across several domains to create applications tailored to the requirements of specific tenants. One of the features of NetApps is that they can be deployed as stand-alone entities or interacting with other NetApps to deliver more complex services. Usually, NetApps follow cloud-native design patterns, and they can be dynamically instantiated, scaled and terminated in edge and cloud computing resources through containers or VMs i.e., CNFs or VNFs. Such resources can belong to different stakeholders; for example, NetApps can be instantiated on the infrastructure of the network operator, on resources made available at the vertical premises (e.g., in the context of the integration with NPNs), on resources from cloud providers, and even over extreme-edge resources (e.g., on Internet of Things -IoT- devices or gateways).

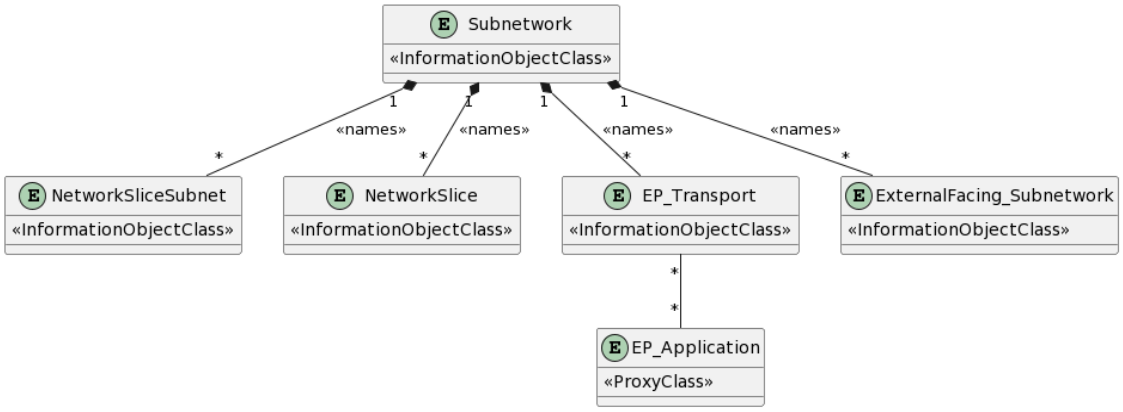

Figure 6-3. Information model hierarchies

As mentioned, NetApps are the second innovation introduced in Hexa-X regarding MOs. NetApps have additional constraints and characteristics, in terms of management and orchestration procedures, to be considered during its Life-cycle Management (LCM) procedures. For example, the NetApp dependencies on particular capabilities or services of future 6G networks would require the setup or re-configuration of Network Slices with the service profiles required to support the NetApp demands. The usage of localisation or network data analytics services requires, on the network side, the exposure of the related APIs towards the NetApp components. In terms of resource placement, the position of the NetApp virtual components could be optimised considering the expected coverage area, traffic patterns, and latency requirements.

Up scoping NF construction to Containerised Network Functions (CNFs)

The last novelty in which Hexa-X envisioned to extend the 3GPP slicing model is the NF containerisation. The Hexa-X approach is to increase the scope of the NF by considering the representation of NF generic objects, meaning that, although the preferable implementation would be based on CNFs, it is considered also important to provide compatibility with other NFs implementations (e.g., VNFs or PNFs).

6.2 Management and Orchestration resources

M&O resources represent the collection of the MNO assets that are in charge of managing and orchestrating the MOs described in the previous Section 6.1. As a whole, these resources are basically dealing with service assurance and fulfilling functions, involving service provisioning, performance management, lifecycle control, fault management, or intent control, among others.

As introduced above, the Hexa-X M&O architectural design described here follows the SBMA model. According to this, M&O resources are a collection of management services, each representing a particular management capability (e.g., provisioning, monitoring, performance assurance…) that allows manipulating MOs. Those management services can be produced and consumed by management functions, which could be mappable to vendor "boxed" solutions. Depending on their in-scope management functions can be clustered into two overall groups:

- Primary Management Functions (section 6.2.1).
- Complementary Management Functions (section 6.2.2).

6.2.1 Primary M&O Functions

Pictured with the "Management Functions" box in Figure 6-1, they represent the collection of Management Functions offering what are considered the basic management capabilities: *fulfilment* capabilities, *assurance* capabilities and *artifact management* capabilities. The first two correspond to extensions for the processes currently captured in the Business Process Framework

(eTOM), a common operating model for telecom service providers coined by the TM Forum [GB921].

These management services are applicable to all managed resources, including infrastructure layer, network layer and service layer resources. This would be done by instantiating the necessary functions to provide functionality on each layer (grey M&O blocks in Figure 6-1). Depending on vendor solutions and/or operator criteria, three archetypal flavours could be considered:

- One single primary Management Function per layer, providing all basic management capabilities for all managed resources in that layer.
- One single primary Management Function per capability type. This means, for example, that there will be one function providing fulfilment for all Hexa-X resources, another function providing assurance, etc.
- One single primary function per layer and per capability type.

Any other combination in between is also within the scope of Hexa-X. In the following subsections each of these capabilities are described.

6.2.1.1 Fulfilment Capabilities

Fulfilment refers to the collection of capabilities that allow provisioning an instance of a managed resource. The act of provisioning includes all the configuration management operations that are needed to manipulate managed resources throughout the entire lifecycle, from commissioning to decommissioning, with all the activation/deactivation and modification operations in between. Following SBMA principles, any provisioning operation over a particular Hexa-X managed resource can be implemented as a CRUD (Create, Read, Update, Delete) primitive over the object (class instance) that represents these managed resources. In practice, CRUD operations can be based on the HTTP methods GET, PUT, POST, DELETE and PATCH. Query parameters provide advanced features like scoping and filtering multiple resources or attribute selection. Support for multiple patch media types enables partially updating resources and patching multiple resources [Nok20].

Note that provisioning of Hexa-X resources in a given layer may have an impact on resources belonging from lower layers. For example, the instantiation of a network slice (service layer) may trigger the instantiation of new function instances and modification of existing ones (network layer). These actions may in turn trigger necessary resource orchestration activities (infrastructure layer), with the allocation of necessary resources (e.g., certain pods on K8s clusters), modification of radio resource quotas, and the set-up of connectivity services on the TN infrastructure. These workflows are executed according to the dependencies/relationships that exist across managed resources, and that are captured in the IM. Every management function providing fulfilment capabilities has a pointer to this model and enforce it accordingly.

6.2.1.2 Assurance Capabilities

Assurance refers to the collection of capabilities that allows network operators to continuously monitor and predict likely problems in the network (e.g., using advanced analytics). Assurance capabilities ensure a service is free of faults (service problem management) and meets the expected behaviour (service quality management). Unlike fulfilment, typically executed per request, assurance keeps executing in closed-loops once set up.

Hexa-X management services for assurance may include:

- **Performance management services:** they allow managing performance metrics production on managed resources and reporting jobs. These metrics could be reported with a file-based or stream-based approach.
- **Fault management services**: they allow collecting from managed resources raw information about faults or misbehaviour. The triggering of faults can be programmatically configured and subscribed to.

- **Trace management services**: they allow configuring and activate various trace jobs, including subscriber and equipment trace data, cell trace or MDT (Minimisation of Drive Tests), among other. These activities, done traditionally for PNF based networks, need to be extended for their applicability in softwarised networks such as Hexa-X.
- **Analytics services**, in charge of processing data and turn them into useful information for service problem and quality management purposes. Data sources may include:
  - Performance Management Functions: they produce performance measurements from individual managed objects (e.g., NFs, network slices, etc.). These measurements can be fed to an analytics function (analytics service producer) for further processing, turning them into useful KPIs.
  - Fault Management Functions: fault data can be aggregated and then stored in alarm records. By doing this, analytics function can turn raw information into root cause indicators (informing about the fault severity) together with proposed mitigations.
  - Trace Management Function, by composing them according to the dependencies captured in the IM.
  - Monitoring functions (see Section 6.2.2.3).
  - Inventory and catalogues (see Section 6.2.1.3)
- **Service problem solving and quality management services**, that take insights from the analytics functions and compare them against to Service Level Agreement (SLA) metrics, computing deltas and deciding on the necessary correction actions to mitigate them. For these decisions, service problem solving and quality management functions can be assisted with AI/ML functions (see Section 6.2.2.1). The decisions are communicated to the fulfilment functions described in Section 6.2.1.1, that enforce them by applying CRUD operations on the targeted resources.
- **Closed-loop management services**, in charge of defining and managing the control-loops that may govern the interactions across all the management functions. As happens with the fulfilment section, a single management function can produce one or more assurance services.

6.2.1.3 Artifact Management Capabilities

Artifacts refer to the set of assets that provide operators with assistance in their fulfilment and assurance activities. For their management, operator rely on catalogues and inventories, together with policies.

Catalogues are a key asset at design time. They store the descriptors/templates and software building blocks based on which managed resource instances are created and operated later on. These artifacts are designed with the purpose to allow effortless portability and ease replicability (i.e., using the *design-once-run-everywhere* approach). Likewise, they are tested and validated before being moved to production environments, following DevOps-like pipelines (e.g., CI/CD). Different types of catalogues can be found:

- Infrastructure layer catalogues: they store the specifications of physical resources (e.g., site, cluster, CPU or memory). Compute, storage and connectivity resources are stored here in order to instantiate the associated NFs.
- Network layer catalogues: they include model-based descriptors that allow characterising the resource requirements and semantics of the different managed objects, ranging from fine-grained, modular constructions (VNFs/CNFs) to upper objects resulting from their composition (e.g., NS, Network Slice Subnet (NSS) and subnetworks). Examples include Helm Charts [Hel22] or Network Service Descriptors (NSDs), among others.
- Service layer catalogues: including model-based descriptors for applications and (communication, digital, and network slice) services to be executed atop the network. Examples include models and templates such as those captured in the TM Forum Information Framework (SID) for Customer and Resource Facing Services (CFS/RFS)

[GB999], as well as Generalised Slice Templates (GSTs) and Network Slice Types (NEST).

Hexa-X acknowledges the problem of having such a number of catalogues, and agrees on the need to find alternative approaches, such as catalogue federation. The objective is to simplify catalogue management, addressing one of the main pain points faced in 5G: the need to synchronize catalogues, each keeping up-to-date copies of the Management Information Base (MIBs) from the rest of catalogues. The complexity grows as the number of catalogues increases, from different vendors, and instantiated at different layers.

Inventories host the run-time information of managed resources. This information would include administrative information (e.g., status, tenant, etc.) and operative information (e.g., key-value pairs for the different class properties) of all Hexa-X resources. In Hexa-X it is considered that the volatility of resources in the Infrastructure Layer can have a great impact on the topology and availability information managed by the inventories (see Section 7.2.1.1).

Additionally, policies capturing static rules related to business aspects or legal compliance (e.g., enforce GDPR, avoid certain services in particular locations by national laws, etc.) are available.

The different catalogue and inventory instances, together with the stored policies, can play a key role in implementing zero-touch real-time practices for fulfilment and assurance. In fulfilment, these instances participate in the feasibility check process for an optimal allocation (embedding) on all the atomic units building up the Hexa-X resources. In assurance, catalogue and inventory instances provide running instance info, that is used for the analytics services.

6.2.2 Complementary M&O Functions

Complementary M&O Functions are those sets of functions that can be used from the Primary M&O Functions described in the previous section, i.e., Security Functions, AI/ML Functions, and Monitoring Functions. As the name suggests, those functions can complement the primary M&O functions operation by providing additional resources (AI/ML, security, or monitoring resources).

6.2.2.1 AI/ML Functions

As it can be seen in Figure 6-1, the AI/ML functions block is split in two by the M&O Scope red line. That means certain AI/ML functions would be specifically designed to support the activity of management functions (within the scope of M&O), while others could be deployed for other purposes (e.g., to support other functions outside the M&O scope, such as Radio Access Network -RAN- functions, CN functions or third-party functions, among others). The main interest here is of course on those functions within the M&O Scope.

As AI/ML functions, those functions are intended to provide the mechanisms to build out the knowledge and the intelligence for controlling, managing and optimising the services deployed on the network, and to take decisions about the actions to be performed at the various architectural layers, implementing the “analyse” and “decide” steps of the various closed-loops.

But why use AI/ML functions in this M&O context? The short answer to this is: to deal with the *complexity* of processes associated with M&O. For low-complexity algorithmic problems i.e., problems requiring to deal with a small number of variables, traditional non-AI approaches are typically enough: Human programmers can easily understand the problem and generate an algorithmic model to solve it. However, high-complexity problems require dealing with a large number of variables that may be related in non-evident ways. This can make regular algorithmic solutions highly complex, or even unapproachable. For these cases, AI/ML techniques have proven to be a valuable resource, as they offer self-learning algorithms capable of dealing with lots of variables, being able to extract non-evident relationship among them.

Specifically for M&O the following sources of complexity are considered, which could be addressed using AI/ML functions:

- Time series: Time evolution of the multiple metrics measuring service KPIs or infrastructure usage parameters can be processed as time series. Isolated time series can be a source of complexity by themselves when it comes to perform forecasting (e.g., to implement proactive orchestrating actions). Multiple time series can be also a source of complexity, not only for forecasting, but also to find patterns and correlations among them that could not be self-evident. AI/ML techniques have demonstrated good performance regarding times series processing [LIM21].
- The extreme-edge integration: As mentioned in Section 5.2, to provide continuum device-edge-cloud management is one of the main novel capabilities envisaged for future 6G networks regarding M&O. In this regard, the integration of the extreme-edge domain is a challenge of paramount importance by itself, due to the high-number and heterogeneity of devices which will exist in this kind of environment. This adds a new source of complexity. In this context AI/ML can be used to address Big Data Analytics to process the big amount of data coming from the diverse extreme-edge devices, and trigger orchestration actions based on that. AI/ML has also demonstrated a good performance in terms of Big Data Analytics [LLF+21].
- Network Operations Management: Application of AI/ML techniques to this context is commonly referred as *AIOps* [MH19][DLH19]. Connected to DevOps, it has to do with automating and improving activities in the operations teams using AI/ML algorithms. Use case examples are alarms filtering (to support identifying relevant events), incident cause analysis, or collect and normalize high volumes of operational data from operational tools, among others.
- Intent-based networking oriented to non-skilled users (e.g., end-users or certain vertical customers), to allow them to deploy or configure their services by simply declaring high-level intentions, and preventing them from having to deal with complicated low-level configuration details. AI/ML functions would be used here to support the translation of those high-level intent declarations (that could be provided even in natural language) into the corresponding low level orchestration actions [SZF+18][SZI21].

AI/ML functions can cooperate and interact with each other, and also with the primary M&O functions, following the SBMA communication approach. Moreover, they can interact with other types of M&O functions, e.g., by consuming the monitoring data produced by the Monitoring Functions and triggering the Management Functions to coordinate the execution phase of the closed-loops.

AI/ML functions can be specialised for the target layers where they are applied, or adopt a joint and cross-layer scope. For example, some AI/ML functions can be dedicated to the network layer or any of its single segments, like the RAN or the TN, taking short-term control decisions (e.g., in the near-real time radio controller) or medium-/long-term decisions for optimisation, resource allocation at service/slice provisioning time or planning. Other AI/ML functions can operate with a wider scope, covering the service management. These functions can combine network KPIs with application and service level data, and have the capability to trigger actions not only at the network layer but also at the service layer, e.g., requesting the re-configuration of application parameters, the scaling of NetApp components, the sharing of NSSs among different service instances, etc.

AI/ML functions could help to support the following M&O related problems [5gaiml21]:

- The NFs placement problem (which is a well-known NP-hard problem) [TCP91][JAS02].
- Forecasting network characteristics and events, focused to trigger proactive M&O actions (e.g., scaling, healing or NF migration actions).
- Forecasting security incidents.
- Autonomous service and slice management, control and orchestration.
- Anomaly detection.
- Closed-loop automation (e.g., by using reinforcement learning techniques [YYY+19]).

- Data processing to support operational teams (e.g., for data validation, anonymisation, data filtering, or classification).

Besides the AI/ML-related logic, the AI/ML functions block in Figure 6-1 would also contain the management functions specific for those AI/ML-related functions, i.e., model training functions for creating and training ML models, or AI models management functions for storage, evaluation, validation, distribution, and discovery of trained models.

Distributed AI components

As illustrated in Figure 6-1, besides the AI/ML Functions block mentioned above, there are also other specific AI-related functions that can be distributed across the network. They represent distributed AI/ML components that would be also managed from the AI/ML Functions block. These components are being defined in the context of WP4 [HX22-D42]. Among other functionalities, they are intended to provide distributed AI-related functionalities (e.g., by means of specific AI-Agents or for implementing federated learning techniques). Anyway, although distributed through the network, dedicated M&O functions are needed also to coordinate them, e.g., collecting model statistics and providing model updates. In this respect, the AI/ML Functions block will need to have dedicated M&O processes for the distributed AI components. However, the association between the distributed AI components and these M&O processes can be however dynamic, e.g., depending on proximity, resource availability, performance of the underlying model (accuracy and/or explainability), etc.

6.2.2.2 Security Functions

The overall objective of the Security Functions is to protect the confidentiality and integrity of operations and data, and to ensure the continuity of the provided services. In order to achieve this goal, an effective method consists of complying with reference cyber security frameworks. Whereas many different cyber security frameworks exist [RKL13] [Anssi21] [Nist18], the general guidelines are similar through the following main functions:

- Identify the assets to be protected, and the security risks they are exposed to.
- Protect these assets by deploying appropriate tools and functions as countermeasures to reduce risks.
- Monitor and detect any signs of an undergoing attack on the assets.
- Respond to the attack with appropriate actions.
- Recover and learn lessons from the attack.

In order to accomplish these security tasks, both security enablers and security management functions are needed. In this document the focus is mainly on the management functions. As represented in Figure 6-1, the security management functions (orange blocks) can be seen as an extension of the primary M&O system, helping to ensure the security of the assets under its management. Those orange blocks represent different instances made of functions in the Security Functions block. Depending on the abstraction layer, different instances of M&O functions (grey blocks) can manage different types of assets, that also need to be secured, as represented in Figure 6-4. For example, in the Service Layer, the valuable assets would be network services and slices, while at the Network Layer the valuable assets would be NFs. For the Infrastructure Layer, computing, storage and network resources would be the main assets to be secured. At each layer, the generic M&O functions themselves (grey blocks) would be secured by the Security functions (orange blocks). Of course, there may be several instances of security functions associated to a single M&O functions block, each being devoted to secure a given group of assets managed by the M&O functions. For example, assets may be grouped by security level requirement.

When an entity requests any service related to a managed object via the exposed interfaces, the generic M&O blocks would be able to delegate the protection of the service to the associated security functions. This delegation would be transparent for the M&O resources and consumers. For example, on a demand for a service such as the provisioning of a network slice, the M&O functions could rely on the security functions to handle the security requirements included in the

request. The security requirement could be expressed in various ways, such as an abstract scale proposed by the M&O, technical security metrics, or high-level intents. When a customer requests a service to be instantiated, the M&O functions could apply LCM actions to instantiate the functions that compose that service. Based on its inner configuration and on the security requirements expressed for that service, the security functions may proactively suggest / impose additional LCM operations to the M&O functions. These may typically include modification of the configurations of the service functions, update of those functions, or addition of new security specific functions. These actions correspond to the "identify" and "protect" steps typically refereed in the cybersecurity frameworks mentioned above.

The measures to prevent attacks from compromising their targets may introduce vulnerabilities related to the presence of unknown bugs or incorrect settings. In addition to these potential vulnerabilities, there are residual risks that are not yet addressed, because of their high costs/benefit ratio. Adopting a defence-in-depth strategy helps to mitigate these vulnerabilities, which should incorporate monitoring, detection, and response measures, in addition to preventive protection approaches. As a consequence, during the lifetime of the asset, the security M&O functions shall continuously monitor and detect incoming attacks, either directly, or using the enablers deployed for this purpose, and take mitigation and remediation actions if a security incident were confirmed. Besides traditional security tools, as firewalls or signature-based traffic inspection, enablers to enforce security in future networks also includes AI/ML solutions for analysis and planning, or quantum-based security mechanisms for key distribution. Those AI/ML functions could be deployed as part of the AI/ML Functions block in the Structural View.

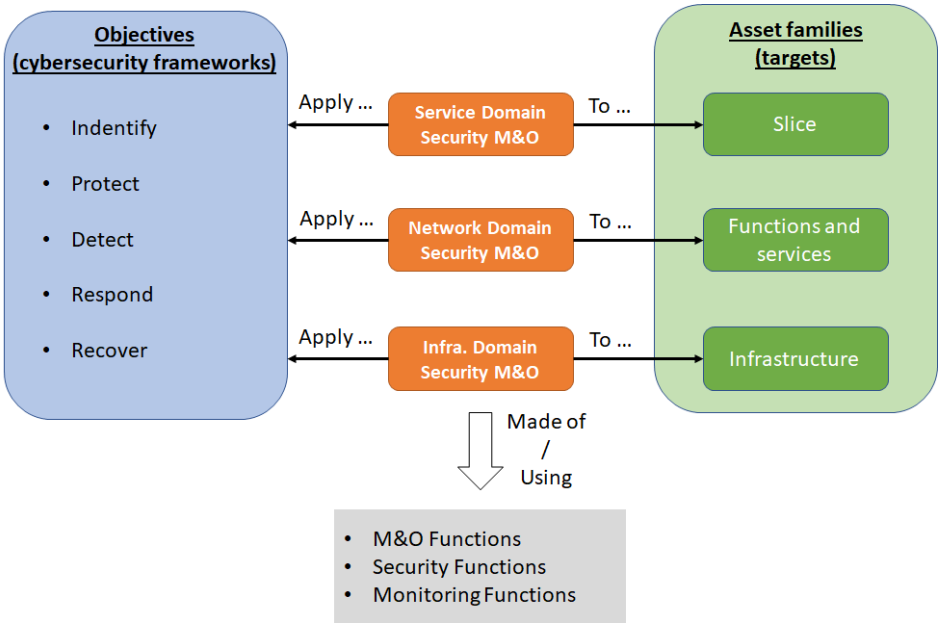

Figure 6-4. Security Structural View.

To accomplish its tasks, the security M&O blocks may use both: services exposed by primary M&O functions to manage the life cycle of assets, and services exposed by other security M&O functions to delegate security tasks or report security incidents. The LCM actions that the security M&O functions may have to apply to its security functions can be delegated to the regular M&O functions. Finally, the security M&O functions could also use the services exposed by other functions, such as monitoring services to collect data, and AI/ML services to obtain predictions and decisions. These services would contribute to the realisation of the automated closed-loops which are part of the security processes.

6.2.2.3 Monitoring Functions

Monitoring Functions are intended to provide information regarding the operational processes, in the form of trace files, alarms, KPI values, or usage parameters, among others. As depicted in

Figure 6-1 Monitoring Functions are used by the Management Functions, which process the monitoring information to perform M&O actions.

Of course, Monitoring Functions have always been there in one way or another related to M&O systems i.e., the SotA telco-grade M&O systems (e.g., [Osm22][ONAP]) already provide monitoring capabilities. However, Monitoring Functions have been typically based on providing monitoring for a fixed set of metrics (e.g., CPU consumption, RAM usage or certain network metrics, among others).

Beyond this, one of the requirements for the Hexa-X M&O system is to be enabled with an advanced monitoring system, able to provide monitoring, telemetry, and the handling of data ingestion from all network segments, allowing to integrate data from infrastructure through data and control planes (Section 5.2) to applications. This requires a higher degree of flexibility in the monitoring system, since it is no longer just about monitoring a fixed set of metrics: having to integrate metrics from data and control planes makes it necessary to integrate custom metrics that would be freely defined by verticals or NS suppliers, since they are who better know what metrics may actually be the most relevant for them. Application-based monitoring makes possible to perform advanced M&O actions in vertical services if they are under-performing or their resources are being over-provisioned. But beyond the Monitoring Functions themselves, this implies to define well-designed semantic data aggregators in charge of registering data sources and data consumers, formats, data aggregation rules, etc., i.e., a data fabric to manage the pipeline from data collection to data consumption. This will enable the data interchange considering these data could be provided from a diversity of sources and with different formats. Some work has been already done towards a monitoring system with features like those described in this section [VARYS], however this should be incorporated as part of the upcoming 6G M&O architecture.

This requirement about mixing application and infrastructure-based metrics is closely related with the AI/ML-related functionalities (Section 6.2.2.1), and it is considered here as a main enabler to provide data-driven orchestration functionalities. What feed AI/ML algorithms are data, there upon, the broader the available dataset, the better (more accurate) AI/ML models can be generated. Enabling the monitoring system to collect (and aggregate) data from the different network layers and domains is considered a key enabler to provide the advanced AI/ML functions needed for intelligent self-adaptation and self-optimisation decisions, based on correlating rich datasets with heterogeneous data.

In summary, functions in this Monitoring Functions block would be responsible for the following tasks:

- To gather raw data from different sources (monitoring probes, system/service logs, etc.) which could be scattered on the different network layers and network elements, including network slices at the service layer (data collection process).
- To provide the necessary APIs to the MNO, verticals, and the service developers make possible to monitor custom, user-defined metrics (and not just a fixed set of them).
- To act as processing elements (e.g., acting as data filters or data normalisation elements).
- To provide continuous monitoring of data towards the Design Layer, in order to support the CI/CD pipelines.
- To collect and store data for implementing model training on AI/ML systems.
- To provide interfaces to the operational information for both: humans (e.g., by means of real-time monitoring panels, or periodic reports) and other systems (e.g., M&O functions, Security Functions or AI/ML Functions).

6.2.3 API Management Exposure

The API Management Exposure represents the functional block enabling and regulating communication among the different M&O resources, within and across administrative domains, enabling the so-called capability exposure [5GVIN-D31] of network elements in the different architectural layers. Using this block all the various network elements in the different layers can

interact and communicate with each other using a variety of granularity levels, but following a unified pattern, i.e., exposing and consuming a subset of services and related management APIs which can be regulated using access control policies. Beyond the communication among the M&O resources, this model can be applicable also with a wider scope to represent potential federation-based interactions. In a nutshell, it mimics the behaviour of the Zero-Touch Service Management (ZSM) cross-domain integration fabric [zsm-002].

Within a single administrative domain, the API Management Exposure registers the endpoints (APIs) associated to individual M&O resources, (e.g., M&O functions, AI/ML Functions, Security Functions, etc.), and regulates their consumption, in service mesh topologies. This regulation span across layers.

In a multi-domain and federated scenario, resources and functions belonging to multiple administrative domains can also cooperate consuming APIs exposed by external domains. As consequence, a common and unified framework regulating the exposure and the management of the various APIs provided by different domains is required. Here, different levels of access and granularity should be supported for the various M&O resources, depending on the profile of the entity that invokes their API. Moreover, the dynamic nature of the environment, where new resources and functions can be added and removed at runtime, brings requirements for the dynamic discovery of the available APIs, and the level of exposure provided for the different categories of potential internal or external consumers.

This framework can exploit the concepts introduced by the CAPIF (Common API Framework), defined in the 3GPP TS 23.222 specification [23.222]. The M&O functions that expose APIs for potential consumers need to register and de-register their APIs, providing enough details to describe the different levels of exposure towards a variety of invokers. For example, for a slice management function, the list of slices may be available in read-mode for both internal and external entities, while the possibility to create new slices may be restricted to other functions of the same domains. Similarly, the possibility to request the modification of an existing slice may be available only for the entity that initially created it or an internal function of the slice itself, for example, an AI/ML component.

In order to support an effective cooperation among different actors, the framework should also allow third-party entities to register their own functions and APIs, in order to make them available in a dynamic manner. In addition, APIs should be deployed utilising an integration or CI/CD pipeline (from the Design Layer) to enable multiple advantages as, automated staging, testing and version management of the APIs, effortless new API management, and fast and reliable deployments.

Some of the features that the API Management Exposure should support include:

- Discovery of API, extended to dynamic and volatile resources and functions.
- Registration and release of APIs, for both internal and third-party functions. This action can be handled as part of the LCM of the function exposing the API.
- Routing across multiple API providers or related function instances, including failover management and load balancing.
- Access control, with management of access policies, authentication, and authorisation of API consumers. For the cases where API consumers are from external administrative domains, federation should be supported.
- Traceability of every request-response message exchanged between API providers and invokers, to allow auditability and further applications enabled by it.

6.2.4 Design System

The Design System, belonging to the Design Layer, includes the functions for the development, definition, modelling, and distribution of the software components which could be instantiated and operated in the 6G infrastructure.

The inclusion of this Design System in the 6G M&O architectural design is one of the main innovations introduced in Hexa-X. It represents the adoption of the cloud-native principles as regards bringing close together development and operational teams by using DevOps practices, facilitating how services are deployed and updated with a very high automation degree (e.g., by relying on CI/CD pipelines).

Although in recent years the adoption of DevOps techniques has been gradually happening in IT services, this has not been the case in telco-grade environments, where the adoption has been more challenging. One of the reasons for this lies in that, contrary to what happens in the IT arena, services operated by MNOs are typically implemented by multiple external providers. Consequently, the challenge is harder: in the usual DevOps approach in IT companies the main objective is to bring together operations and development teams within the company, but in the telco-grade environment it is not just a question of integrating departments within the same company, but rather integrating the MNO and different external vendors with different methodologies and corporate cultures. The inclusion of this Design System in this M&O architectural design targets addressing this challenge, in order to bring the benefits of the cloud-native techniques in terms of automation and agility in the software deployment processes. Although particular proposals are already in the SotA [OnDev22] or previous 5GPPP projects (e.g., the Dev-for-Operations model in [NGPaaS-D32]), it is considered necessary to make further progress in this area for the future 6G networks.

As illustrated in Figure 6-1 three different aspects are considered in the Design System:

- Software and descriptors design,
- DevOps/AIOps framework, and
- Intent-based service definitions.

The first aspect refers to the "Dev" leg in the DevOps model, i.e., those aspects regarding the SOFTWARE design and development itself, including the design of the deployment descriptors adapted to the MNO requirements. It comprises activities and tools regarding the coding, building, and testing activities at the SOFTWARE provider side. Overall, this includes two main aspects: (i) the internal structure of the network elements and their interfaces, and (ii) the specification of their requirements in terms of mobile connectivity, deployment in the virtual infrastructure, KPIs, dependencies on functionalities offered by the network, etc. The first aspect helps the service providers in the composition and delivery of complex services with multiple applications. The second one provides the guidelines to properly instantiate and operate the services within a 6G infrastructure, guaranteeing the required resources and connectivity in an automated manner during its entire lifecycle.

The DevOps/AIOps Framework includes two aspects: the "continuous" pipelines typically associated to DevOps, (i.e., pipelines for CI/CD, Continuous Testing, Continuous Monitoring, etc.), and the operational activities that can be supported by AI/ML techniques (AIOps), e.g., for alarms filtering, root cause analysis, data clustering tools for aggregating huge volumes of operational data, etc. This DevOps/AIOps Framework would help to increase automation level in the software releasing processes and in the regular operational tasks by putting close together the operations side (including MNO teams), and development side (including the involved vendors).

Finally, intent-based service definitions refer the tools to facilitate the intent definition regarding the services development and deployment processes. E.g., guiding the developers in selecting the required parameters and evaluating their consistency. Intents can be validated and translated by the system logic into slice descriptors (e.g., using the slice modelling from 3GPP or GSMA) and/or NSDs which would feed the M&O functions of Service and Network Layers. Alternatively, intents can provide complementary information with respect to what is declared in the NSDs provided by the operators, with customer-managed details related to the required service parameters (e.g., on-premises vs off-premises deployment, data protection strategy, and replicability for high availability, metrics to be collected and reporting period, etc.). In this case, intents can be directly translated into appropriate configuration and management actions.

The intent-based service definition block is intended to support NS developers since, in particular, verticals could provide only high-level specifications, defining just intents with the characteristics of the network capabilities defined from a service perspective. Such intents could be defined using a formal language with a well-defined IM, or even natural language (an example of model for the intent definition is the vertical service blueprint -VSB- initially defined in the 5G-TRANSFORMER project [5GTRA-D31], which has been adapted for NetApp blueprints in the context of VITAL-5G project [VIT5G-D21]). In Hexa-X this model is extended to encode information about (i) the desired placement of service components on edge and extreme-edge resources, (ii) a more advanced definition of metrics at the service level, with high-level definition of their impact on the network related requirements (as input for the network automation processes), (iii) KPIs, and (iv) requirements in terms of AI/ML functions and data the service would need to consume to drive the automated instantiation of the related AI/ML agents and functions.

7 Functional View

The Functional View is intended to describe the "dynamic" aspects of the M&O system from a high-level perspective, describing relevant behaviours or mechanisms that would be implemented through the interactions among the basic building blocks described in the Structural View (Section 6).

This view will be described in the following sections by means of the description of those mechanisms or processes that are considered the most relevant for future 6G networks. The "basic" orchestration actions are introduced first in Section 7.1; these include actions such as provisioning, scaling, NFs migration, etc. Section 7.2 describes what are considered the core orchestration processes, which are split into four: E2E seamless integration processes (Section 7.2.1), Programable processes (Section 7.2.2), Automation Processes (Section 7.2.3) and Data-driven processes (Section 7.2.4).

7.1 Basic Orchestration Actions

Basic orchestration actions can be understood as those M&O actions that could be used as the main building blocks (in terms of processes or mechanisms) that would be used to compose the other more complex M&O processes (described in Section 7.2). They can be considered as the most basic behaviours regarding M&O. They apply to all MOs outside the dashed red line in Figure 6-1, including the three layers: service, network and infrastructure. When applied to infrastructure they are typically referred as resource orchestration actions.

Typically, basic orchestration actions are already supported by regular SotA M&O systems, but it is important to mention them in this document as a matter of completeness. They can be summarised as follows:

a) Instantiation actions, i.e., actions regarding the MO creation using the necessary onboarding resources. It is assumed that during the instantiation process it must be possible to indicate the MO configuration parameters. Among others, *placement* parameters would be provided, indicating the specific infrastructure resources on which the MO would be instantiated.
b) Scaling actions, to increase or reduce the capacity of the MOs (e.g., available memory or bandwidth usage). When applied to infrastructure resources scaling actions are used to allocate/deallocate resources (such as VMs) on demand to adapt to workload changes.
c) Update (or configuration) actions, meaning the configuration changes on the already instantiated MOs. Particular update actions would be *migration actions*, when the update refers the MO placement.
d) Upgrade/downgrade actions, meaning the updating of the complete MO by newest or older software releases.

e) Data Gathering actions, meaning those actions targeting to get data from the MOs. Examples of Data Gathering actions are the *Performance Management Actions*, which are those in charge of collecting performance metrics; also, the *Fault Management Actions*, which are those related to possible alarms collection (alarms are typically asynchronously generated from the MOs, but they have to be collected and processed from the M&O system).
f) Terminating (or decommissioning) actions, such as the MO termination releasing all the consumed resources.

7.2 Orchestration processes

The orchestration processes would be those more complex processes that could be arranged using the basic orchestration actions introduced in the previous section. These processes are described in the following subsections, split into four general categories:

- E2E Seamless Integration Processes (Section 7.2.1)
- Programmable Processes (Section 7.2.2)
- Automation Processes (Section 7.2.3)
- Data-driven Processes (Section 7.2.4)

Please, note that not all possible M&O processes are described here. Information is provided for those processes which are considered the most relevant for future 6G networks.

7.2.1 E2E seamless integration processes

E2E seamless integration processes refers those processes in charge of managing the available infrastructure as a common pool of resources. The objective is to provide management teams or other processes a sort of *common facade* to allow them to deploy and operate NSs without much concern about the low-level details of the specific infrastructure on which they would run.

The main challenge of this topic is the high heterogeneity of the infrastructure resources. While it is still too soon to predict which will be the precise technologies to be in use, it is possible to foresee that there will be a significant number of network elements with a high level of distribution, from the extreme-edge up to the central cloud. The section is split into four subsections:

- Device-edge-cloud continuum M&O (Section 7.2.1.1).
- Network Slices Orchestration (Section 7.2.1.2).
- Integration with other networks (Section 7.2.1.3).
- Optimised placement (Section 7.2.1.4).

7.2.1.1 Device-Edge-Cloud continuum management and orchestration

5G enabled adaptive and flexible service M&O leveraging on the NFV and Software Defined Networks (SDN) technologies, making computing, communication, and storage resources highly flexible. That allowed the implementation of E2E network slicing, focusing on dynamically allocating NSs on the MNO core and edge infrastructure resources [FPE+17] [SZV+19] [Zha19] [ATS+18]. Hexa-X proposes to evolve this concept for future 6G networks by introducing the "Device-Edge-Cloud continuum M&O" concept, which considers expanding network slices beyond core and edge resources by integrating other network domains, from the extreme-edge (those end-user resources beyond the edge) up to the cloud, considering all the network resources in between.

In short, continuum M&O proposes to implement a common M&O framework able to efficiently deploy and manage network slices on a wide distributed computing platform, integrating:

- Heterogeneous network domains, considering not only the MNO public resources (e.g., mobile and fixed access networks, telco edge nodes, core cloud nodes and WAN

resources, providing connectivity services across them), but also 3rd party nodes (e.g., factories, transportation hubs, hyperscaler clouds), some of them forming NPNs.
- The extreme-edge domain, i.e., those end-user devices (personal devices, automotive devices, IoT devices or gateways, AR/VR and gaming devices, industry devices…) beyond the MNO access network.

The continuum M&O main objective would be to provide the necessary mechanisms to support creating and managing network slices with resources from these heterogeneous and distributed domains. In this context orchestration is quite challenging because NSs are composed of multiple and heterogeneous resources, which may have integration and interoperation dependencies [MGZ+19].

But continuum M&O is not just about incorporating new infrastructure resources to the M&O workflows. It is also about enabling the atomic NFs composing the NSs in such a way they could be efficiently moved among the different domains in a very agile way (effortless portability); this is what in fact implements the concept of continuity on the available infrastructure resources (e.g., one could imagine certain network or service application functions moving through the core, the edge or a certain device in the extreme-edge, to provide service to end-users moving through different locations).

Other relevant challenge comes from the extreme-edge itself: this domain must be considered as a totally different domain to the core or even the regular edge domains owned by MNOs, which typically are under strictly controlled conditions. On the contrary, extreme-edge devices are at the end-user's scope, meaning they are not necessarily under regular maintenance (they can be error-prone) or they could be unexpectedly moved or even switched off/on (they could behave asynchronously regarding their status). This applies not only for personal devices, but also for other devices that, although in corporate environments (e.g., vertical industry environments), are outside the MNO scope. Besides, it is especially remarkable the fact that extreme-edge environments will be comprised of a high heterogeneity of devices and data protocols, including devices with limited computing and storage resources. It will be also massive in scale, even exceeding human scale regarding regular operations activities. In order to make this happen, it would be necessary to align with extreme-edge device providers for them to offer the necessary Software Development Kits (SDKs) and well-defined APIs (e.g., to configure the devices for making use of slicing capabilities, like attaching certain applications to an operator supported slice).

To efficiently provide the previous features, continuum M&O would rely on the following main enablers:

a) Well-defined and standardised APIs, for enabling effective communication across all domains. These APIs would be made available for consumption by the API Management Exposure block (see Figure 6-1).
b) Leveraging mainly on stateless and lightweight cloud-native micro-service-based components for implementing managed objects, in order to ease their allocation and live migration on the different network domains, including the extreme-edge.
c) Highly automated processes for implementing the regular operational tasks in all domains, but with special focus on the extreme-edge domain. A large number of devices in this domain could make unfeasible to manually configure individual devices (as done in previous generations); therefore, automatic processes should be enabled to automatically apply massive configuration actions on large sets of devices.
d) Updated security resources, to mitigate the potential risks associated with integrating the aforementioned new network domains we are mentioning here (extreme-edge and other networks).
e) AI/ML capabilities, to help managing the complexity associated to the large amount of network elements and devices, mainly those on the extreme-edge domain. This would be oriented to help performing the regular operational LCM tasks on these large number of devices, and on the associated monitoring tasks.

f) A unified development SDK considering the different resources across the network continuum, enabling development pipelines across the different domains, also for the external stakeholders. This would be provided as part of the Design Layer (see Figure 6-1).

g) Beyond the SotA infrastructure managers, to manage the asynchronous and error-prone nature of devices in the extreme-edge domain. Besides the regular infrastructure management tasks (which are typically manual and hence slow operations), these components should provide new mechanisms for the infrastructure discovery and real time event-based status update, integrated as part of the Infrastructure Layer Control Loops (see Figure 6-1).

h) Connected to the previous, there should be mechanisms supporting management of resource volatility regarding the registry management in inventories, capturing the short-lived nature of the extreme-edge resources. This way M&O functions could consider this information in advance to take medium- and short-term decisions (such as planning and resource orchestration).

For providing these functionalities, the Management Functions operating at the Service and Network layers (Figure 6-1) should communicate with the different domains in order to: (a) collect information about capabilities and characteristics of available resources and (b) trigger the instantiation and LCM actions of service components and NFs. In turn, the various orchestration platforms should operate over the controlled nodes through a set of programmable interfaces enabling the reservation, scheduling, allocation, release and monitoring of the resources. The extreme-edge nodes at the UE side should provide also certain level of programmability, so that it could be exploited in order to run third-party service components.

Each infrastructure manager should also expose its own interfaces following the API Management Exposure mechanisms (see Section 6.2.3), so that their functionalities can be seamlessly consumed by the Management Functions to trigger the reservation, allocation or operation of the resources in each technological domain. Due to the specific exposure policies that may be applied by the various stakeholders, the level of visibility and programmability available in each domain may vary, even according to the requesting entity. For this reason, inter-domain mechanisms should be enabled for the implementation of procedures for dynamic registration and discovery of: (i) platform capabilities, (ii) exposed APIs and (iii) available resources in each domain contributing to the extreme-edge/edge/cloud continuum, combined with access control and accounting mechanisms. The first two aspects are quite static, and they can be easily advertised between domains based on pre-established service level agreements. On the opposite, the dynamicity of the resource utilisation requires mechanisms for automated publication and discovery. This become particularly relevant and challenging for the extreme-edge domain, where nodes are more volatile, and their resource capabilities can change quickly. This can have an impact on the synchronisation approach, which may require dedicated and secured communication paradigms and more frequent information exchanges. In this context, the adoption of AI/ML-based prediction can greatly improve the efficiency of extreme-edge allocation strategies, since it may help to automatically identify patterns describing in advance the dynamic and variable trends of the resources' availability.

Figure 7-1 shows an initial high-level approach on which M&O systems for 6G networks are envisaged to span across the multiple network domains, extending the traditional orchestration of cloud and edge resources towards the extreme-edge, and where the UE devices are equipped with computing and storage resources that can be allocated dynamically to run part of the service components. Moreover, following cloud-native patterns, the E2E vertical services may make use of serverless computing technologies (in a Function-as-a-Service paradigm), where resources are consumed dynamically on an event-based schedule without a continuous reservation, allocation, and maintenance.

As it can be appreciated, different kinds of orchestrators and management platforms can be deployed in the various segments to control the computing nodes at the extreme-edge, edge and cloud segments, each of them specialised to deal with the characteristics of the resources under

its scope. These orchestrators can be considered as "structural" ones, i.e., devoted to the resource orchestration within a particular domain or a particular platform. The Management Functions operating at the Service and Network layers of the overall architecture (see Figure 6-1) would be responsible for the coordination of the provisioning and LCM procedures across all these heterogeneous platforms, potentially deployed in different administrative domains. This kind of orchestration is operating at a "functional" or "service" layer, and spans across multiple domains and platforms, making use of hierarchical levels of abstraction.

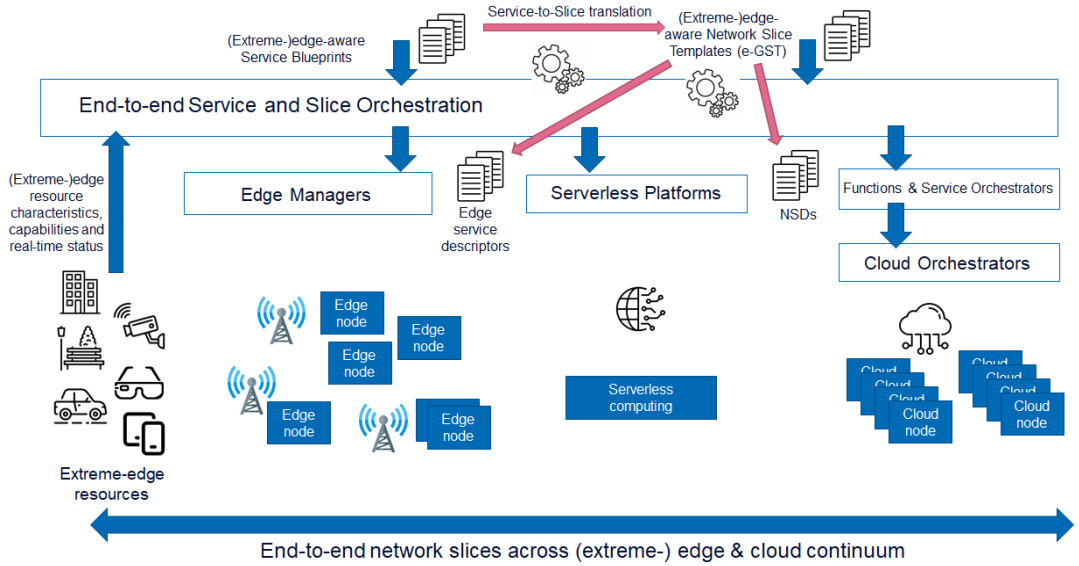

Figure 7-1. E2E orchestration of device-edge-cloud continuum.

The actions of the Management Functions, which can also operate when deployed in distributed environments and operating in distributed mode, and optionally with the support of AI/ML Functions, also distributed, must be coordinated in an E2E manner at both: service and network levels (i.e., Communication/Vertical Service Management, and Network Slice Management Function respectively). The interaction among Management Functions in different domains can be mediated through the API Management Exposure block.

The E2E management procedures are driven by the service requirements. These requirements could be expressed through intents which, in 6G networks, must be able to capture the needs and characteristics of the application components, that can be also running at the extreme-edge. For example, in case of service intent modelled through Vertical Service Blueprints, their IM could be extended with (extreme-)edge-aware items, e.g., to describe the placement of atomic service components in extreme-edge resources, their dependencies on IoT devices or platforms, their mobility and traffic patterns, the mobile connectivity requirements towards the edge or other extreme-edge nodes in a network mesh, etc. These service requirements would be then translated into GSTs, as specified from GSMA, but enhanced to consider extreme-edge requirements at the network slice level. The resulting slice definition, combined with the service and application information (provided through the intent declared from the service customers), would be finally used to derive the corresponding descriptors (defined at the operator level) for the deployment of the slice subnets in the various domains. At the NFs level, the descriptors would usually be in the form of an NSD (see Section 6.1), with multiple levels of nested NSDs related to the single domains. The vertical applications could be modelled through several cloud-native service-oriented descriptors, and could be specialised on the basis of the target platform type (extreme-edge/edge/cloud or serverless).

7.2.1.2 Network Slices orchestration

It is envisaged that in 6G networks slice orchestration would need to achieve higher levels of dynamicity and automation than in the previous generations, with provisioning customised on-demand on the basis of service intents and slice operation at runtime, enabling a continuous

service assurance. The strategies for slice management have the objective to optimise, also from an energy-efficient perspective, the overall usage of the resources, in compliance with the SLAs to be guaranteed to the running services. In this context it is fundamental a bi-directional cooperation between the service and the network layers, enabling the adjustment of network slices according to the service dynamicity. Moreover, the extension of network slices towards the extreme-edge requires an extension of the awareness of the network infrastructure at the slice management logic.

One of the most important issues regarding network slicing operations is the reliable deployment (admission) of network slices, as well as efficient and dynamic allocation of resources to running network slices in order to ensure slice SLA compliance while optimising resource usage, and therefore energy consumption. Such allocation should be made with some threshold to avoid too many resource scaling operations during slice lifetime. In order to make reliable admission as much as possible information regarding slice usage should be provided. The data-driven approach proposed by Hexa-X (Section 7.2.4) would enable the usage of the information coming from multiple domains and network segments. The first step in proper dimensioning of resources needed can be the slice template, that would include information regarding its usage. To that end the GST/NEST approach by GSMA for slice instance initial resource allocation could be used. This can be done by configuration of GST attributes before slice deployment. The attributes may include number of users, coverage, per-user throughput, slice availability, service KPIs, slice isolation level, throughput, latency, etc. The approach also defines network slice functions that can be owned by the vertical, and root cause investigation that is provided to the vertical in case of network slice KPI degradation. The 3GPP Network Resource Model (NRM) [28.540] [28.541] already allows mirroring GST into service profile attributes, which are decomposed into domain-specific attributes, 5G Core, RAN or TN. Based on these attributes, domain-specific configuration actions can be triggered accordingly.

In Hexa-X, using the NEST [Ng116] as input, the orchestration of network slices would be handled through a set of cooperating Management Functions in the M&O network layer (Structural View). An example of these network layer Management Functions could be the Network Slice Management Function (NSMF), which coordinates the E2E composition and the stitching of the slice components across multiple domains that, in 6G networks, should reach the extreme-edge. While this function was already introduced in the management system of 5G networks, some evolutions would be needed in 6G networks, e.g., to reach the extreme-edge, or even completely novel Management Functions could be included for this scope. For example, the NSMF decisions are expected to be increasingly supported by AI/ML techniques in 6G networks and, even if operating with a centralized logic and an inter-domain scope, they can be actually distributed across multiple cooperating elements. The intra-domain provisioning, which could be technology-specific, would be delegated to other specialised Management Functions, still operating at the network layer. In 5G networks, this role was covered by the NSSMF (Network Slice Subnet Management Functions), responsible for the provisioning and runtime management of single slice subnets within the access, core and transport domains. 6G networks may adopt a similar approach or introduce additional functions. However, the internal features should be updated to comply with the new 6G requirements in terms of extreme dynamicity and nomadicity of the resources, especially when considering the extreme-edge domain, and the characteristics of multi-domain environments where NSSs should be provisioned and stitched together, combining resources from multiple stakeholders. It should be noted that in 5G networks the hierarchy that maps a NEST to the corresponding Network Services, composed of VNFs, PNFs, and CNFs, and the associated resources, is still quite static and usually pre-defined by the MNO. In 6G networks, this approach should evolve towards a more business-oriented slice model which would trigger its automated and real-time translation into the underlying hierarchy of virtual functions and resources, required to provision the concrete slice instance.

Multi-tenancy is an additional key aspect that allows to share NSSs across multiple E2E slices, in compliance with pre-defined isolation constraints and the performances requested for the running services. For example, the control plane components of the CN could be shared between two

different slices, while each of them may have its own dedicated User Plane Functions (UPFs) to preserve the traffic isolation. In other cases, the anchor UPF at the core can be shared, while dedicated intermediate-UPFs can be instantiated at the edge, one for each slice. In order to guarantee the continuity of the E2E performance, the shared subnets may need to be scaled, terminated or modified as consequence of actions and events occurred at the parent network slice or in other subnets. While multi-tenancy was already considered in 5G networks, the extension of network slices to the extreme-edge, with resources owned and controlled by different actors, brings new challenges related to the definition of SLAs among multiple parties, to security issues, and to the need of policies for regulating the disclosure of multi-source information in such multi-actor context. Moreover, the constraints in terms of power consumption and resource capabilities of extreme-edge nodes impact the sharing decisions and lead to more dynamicity in scaling or migration of slice subnet components running there.

7.2.1.3 Integration with other networks

As introduced in Section 7.2.1.1 (Device-edge-cloud continuum M&O), it is expected that 6G networks will be able to cope with different access and backbone networks to provide E2E solutions. Typically, the integration concerns different technological or administrative domains in that context. Such integration can be dynamic, and it concerns the user plane primarily but, in some cases, has to be supported by control plane services for the E2E integration.

In the virtualisation era, integration also concerns M&O in order to support services that are deployed in the E2E manner. Following this, the runtime management (including service assurance) requires interactions between different, typically technology-specific, management systems. The integration of M&O systems is, in such a case, very challenging. Such integration can be simplified if a proper exposure of the management resources is done, and the inter-domain interfaces are implemented using the intent approach. In the case of multiple administrative domains, a limited or abstracted view of each domain can be provided only due to confidentiality and security reasons. Using E2E M&O, additional functionalities can be added to each subnetwork in order to support network or application-level integration. In the Hexa-X approach, the integration of the M&O of different subnetworks is possible via the use of the API Management Exposure block. This block can be used for exchanging the secure M&O information between subnetworks using the SBMA model (see details in Section 6.2.3). Dependent on the use case different levels of granularity should be supported with the dynamic discovery of the available APIs.

There are, at least, two ways of orchestration of E2E slices or services. The first one is based on a centralised orchestration (e.g., using a single orchestrator) which has the capability of direct orchestration of resources of different subnetworks. In such a case the Management API Exposure would be used for the exposure of resources, management data, and the management functions. The E2E orchestration may also be implemented using multiple orchestration domains (including extreme-edge orchestration) with dedicated orchestrator each. In such a case the Management API Exposure would be used not only for the exchange of the management data and the exposing of the management functions between subnetworks, but also in the exchange of information between the orchestrators (note in this case the importance of standards, such as those from 3GPP).

The integration of public networks with private networks can be made in a similar way as the integration of networks operated by multiple operators, but in such a case, it must be assumed that the private network management system is relatively simple and has not implemented some management and orchestration mechanisms that exist in public networks and are required for the integration of M&O systems (i.e., the API Management Exposure). In such cases private network management systems shall be provisioned with adaptors to consume Hexa-X offered capabilities. The easiest way of making E2E integration is to use the network slicing technology, with generic network slicing mechanisms for slice selection and authentication. In such approach, however, the exchange of management information would be limited, and would typically not concern the infrastructure domain.

Another important fact is the need of having a network service mesh which can significantly solve some of the aforementioned microservice challenges, e.g., the security. In this case, the service mesh architecture would be brought by design, which would be essential for the 6G ecosystem.

7.2.1.4 Optimised placement

One of the problems typically associated to M&O services is the placement problem. This is an optimisation problem consisting in finding the best (most optimal) way to deploy the atomic software components (e.g., NFs) that typically conform other more complex structures (e.g., NSs or Network Slices). Formally, this problem can be treated as a specific instance of the well-know "Bin Packing Problem" (BPP) [GKM18] [AMB20] [MRP11], which has been identified as an NP-hard problem. In practical terms the proper addressing of this BPP applied to datacentres can bring benefits in resource optimisation and energy savings (if certain functions can be deployed together on certain servers, perhaps other servers could be switched off).

The placement problem is closely related to the E2E seamless integration processes that are been considered in this section, mainly regarding the continuum M&O of services across the extended network domains considered in Hexa-X (from the device up to the cloud). However, the main issue for addressing this topic lies in the BPP being an NP-hard problem. In practice, this means that computing the optimal solution could take a huge amount of time. Despite its worst-case hardness, certain algorithms can be used to address even very large instances of the problem in reasonable times (e.g., [GKM18] or [AMB20]). Additionally, there are also many approximation algorithms [CEC+21], and even AI/ML-based algorithms [TCP91] [JAS02].

From the Hexa-X perspective, those algorithms would be executed by specialised placement functions in the Management Functions block (see Figure 6-1), perhaps supported by the AI/ML Functions block if required. These placement functions would drive not only instantiation actions, but live-migration actions also.

Regarding the infrastructure management, those algorithms should consider updated information on the available infrastructure for deploying software components, considering both: 1) the need to integrate information on extreme-edge resources, and 2) the need to handle volatility and short-lived features of some resources (not only from the extreme-edge, but also for 3rd party infrastructure nodes). This information should be updated in real-time into the artifact management capabilities inventories/repositories (Section 6.2.1.3) for the placement engines to take it into consideration when taking decisions, since not all the infrastructure elements could be always available or suitable for hosting certain software functions (especially considering the high heterogeneity and the limited capacity of certain resources at the extreme-edge). Also, to prioritise some infrastructure resources for running certain functions (e.g., those powered with renewable sources).

7.2.2 Programmable processes

Network programmability is one of the key enablers for coping with the multiplicity and heterogeneity of NSs, the diversity of the 6G infrastructure, and the need for utmost efficiency. Service platform programmability enables managing the network in an algorithmic way, leveraging on modern software virtualisation technologies. Network programmability will also abstract all the required network/service and resource configuration, as well as the generation and management of policy lifecycles, considering also that the number of local breakouts (public and private) are expected to grow exponentially.

Programable processes in this subsection list those processes enabling network programmability. The following specific processes are addressed here:

- Intent-based processes for expressing application/service requirements (Section 7.2.2.1)
- Processes for enhanced service description models and profiling (Section 7.2.2.2)
- Monitoring and diagnostic processes (Section 7.2.2.3)
- Processes for reasoning (Section 7.2.2.4)

- Software Integration processes (Section 7.2.2.5)

7.2.2.1 Intent-based means for expressing application/service requirements

The concept of "intent" is particularly interesting in the area of programmable and autonomous networks, since it allows to declare the requirements, constraints, objectives and context of the desired connectivity in an abstract and technology-independent manner ([CCG+21] [BPP+19] [ZT20]). The perspective is thus moved from the network operation and configuration to the requirements and expectation of the applications and services that will make use of the network connectivity. In this sense, the intent is something easy to interpret and manage without the need to have a deep technical knowledge of the networking aspects. As such, it is suitable for describing the intentions of vertical users following their own language and expertise, which does not need to cover the details of the underlying infrastructure or the knowledge of the low-level M&O resources. This simple and service-oriented language facilitates the verticals' approach towards the mobile networks and their interaction with the M&O system to declare their requirements. In general, intents can be defined at different layers applying increasing levels of abstraction. For example, an intent may describe the expected performance of infrastructure resources, or identify the characteristics of the connectivity a mobile network should provide, thus operating at the resource and network layers. Alternatively, the intent can provide service or even business-oriented indications, delegating the translation of such "upper layers" constraints into concrete policies and configuration actions to the lower layers of the architecture.

6G networks can further evolve and enrich the functionalities offered to process and translate the verticals' and customers' intents to drive the network configuration. For example, in order to model a secure connection between two networks, an intent could just declare the need to establish a secure tunnel between them. Starting from this, an operator would additionally identify the classifiers for the traffic using such a tunnel and any other parameter required to fully configure the tunnel itself. In addition, specific service requirements would need to be defined by the user, such as that high-resolution 1080p video would be streamed. Should the user wants to transmit multiple streams of 4K video in the future reconfigurations to the infrastructure, transport and access network have to be made to support these high throughputs needs. To avoid any service outage or other infrastructure disruption, assurance checks will be enabled to reconfigure anything that seems to be out of place. Intent-based means are the first step to enable full system automation and configuration of all software and hardware devices based on the enhanced service description components in use. The initial configuration of the system would be generated by the intent-based mechanism, and then the system would provide ongoing assurance checks between the intended and operational state of the network, utilising multiple data-driven processes. With the use of profiling techniques that classify, extract, and validate transmitted service data, the system could understand the service that is being used (e.g., video, its quality, etc.). In particular, the M&O system could provide enhanced support to automatically validate the intents defined by the users and increase the efficiency of the intent's interpretation. In fact, AI/ML techniques would enable a sort of "self-learning" about how to correctly interpretate and translate the customers' intents into concrete configuration and orchestration actions that would be applied through specific Management Functions in the structural view. Besides, the integration with the network Monitoring Functions would allow to collect and process service and network KPIs, to automatically detect potential changes in the service intent declared initially at the provisioning time, and react accordingly during the runtime. Finally, it is worth to mention that the current approach to translate intents into network configuration actions would be mostly based on static criteria and policies, configured manually by the network administrator. Introducing AI/ML to enable continuous self-learning procedures, these translation criteria, rules and policies could be automatically updated and modified in a dynamic manner, e.g., re-adapting them on the basis of the current network conditions, to better match and guarantee the desired performance.

Figure 7-2 shows a possible workflow for the management of service intents in the context of 6G networks, highlighting the interaction with the M&O functions operating at the network layers in Figure 6-1 (mostly Management and Monitoring Functions), and the applicability of AI/ML-

based closed-loop logic to various phases of the intent management. In particular, the management of the intent can be addressed introducing new functions at the design and service layers of the Structural View, as shown in the picture.

Starting from top of the picture, a typical intent-based architecture would provide tools and APIs to allow users to declare their intents and request the provisioning of the related service. Two major categories of intent definitions could be supported: (i) intents expressed in natural language, and (ii) intents expressed through a formal IM. In the former case, the description provided by the user should be firstly interpreted and transformed into a definition expressed through the formal language understood by the system. In this regard AI/ML techniques have historically demonstrated to be a valuable resource [KAR19][DK15], being the most relevant advances those derived from the GTP models [BMR+20], which could be applied in this area.

Regarding the second case, the compilation of the intent could be facilitated through tools and graphical interfaces that could guide users in filling all the required values, which could be then elaborated internally to generate the internal representation of intents. In this case, some templates or blueprints could be adopted to provide a generalised model for the intent's declaration. Some IMs have been proposed in this sense, both in standards (e.g., from Internet Engineering Task Force -IETF- [CCG+21] and TMF [1253a]) and in research activities (e.g., the Vertical Service Blueprints adopted in 5G-TRANSFORMER [5GTRA-D31], 5Growth and 5G EVE [5GEVE-D41] projects). Once filled with the specific values defined by the users, these templates would originate "intent descriptors" compliant with the expected format, which could feed the following steps of the intent processing.

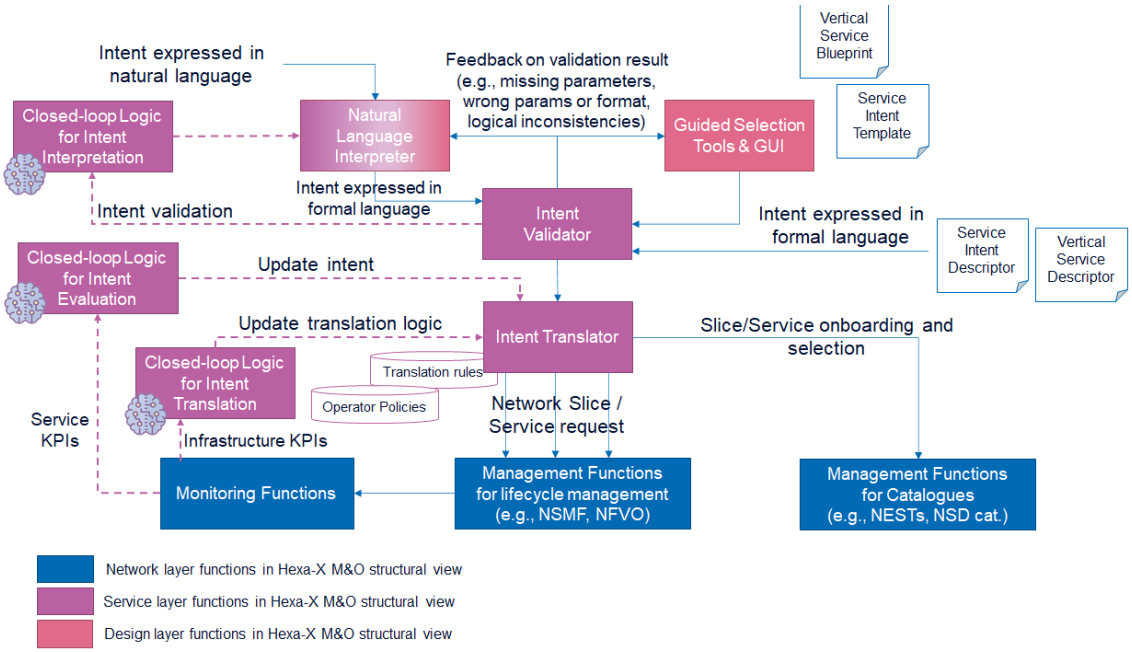

Figure 7-2. Intent management enhanced by AI-based closed-loop logic.

After its declaration, intents would need to be properly validated, and its consistency should be verified from a formal and a logical point of view. For example, it could miss mandatory parameters, define values out of the acceptable range, or define contrasting values that could lead to a logical inconsistency of the declaration as a whole. To avoid this, a verification would be performed by the Intent Validator block in the figure that, in case of errors, would return feedbacks and suggestions to help the users to compile their requests. At this stage, AI/ML techniques could be used to help the system to learn dynamically about how to interpretate the inputs provided by users, e.g., by elaborating the most common errors and correlating with the following user's attempts to fix them.

Once validated, intents would be elaborated by the Intent Translator (see figure), which would be in charge to transform the abstract connectivity requirements into a set of concrete technical

specifications, that could be directly used to configure the network and provision the functions and resources needed to guarantee the expected performance. This translation, for example, could lead to NSDSs or network slice profiles to feed the instantiation procedures at the life-cycle-management functions in the Management Functions block (Figure 6-1). Here, before moving to the actual instantiation, a preliminary interaction between the Intent Translator and the catalogue function of the M&O network layer would be required. Depending on internal policies, a push or pull model could be adopted. For instance, the Intent Translator may be allowed to generate autonomously new descriptors to onboard dynamically the catalogues, or it may be limited to the offer already available in the system. In the latter case, the Intent Translator would only be able to select the pre-existing descriptor that better could match the intent, without the possibility to create new ones. Once the target NSs or network slices were defined, a provisioning request would be issued to the corresponding M&O Management Function (e.g., an NFVO or a NSMF), which would be then in charge of establishing the required resource allocation, virtual function instantiation, and network configuration.

At runtime, the service, the network and the infrastructure components would be continuously monitored to detect potential issues or degradation of the expected performance. During this phase, some network automation mechanisms, optionally triggered by AI/ML-based decisions, would help to dynamically modify the system configuration in order to continuously meet users' expectation. Different types of misalignments may occur with respect to the initial declaration and translation of intents. For example, the initial service-level requirements defined in the intent could change due to an increasing service demand, a modification of the application context and configuration, etc. This could be easily detected through the "closed-loop logic for intent evaluation" depicted in the picture. Processing the service KPIs, the system could automatically derive a new intent specification to trigger an autonomous re-configuration, scaling, or update of the initial NS or network slice.

An alternative situation could be related to the poor quality of the intent translation logic, that is usually driven by configurable rules and policies which are manually defined, and could be prone to errors. In this case, the current allocation of the resources would be not able to properly sustain the service demands, i.e., in terms of traffic load or processing load in a specific service or network function, while the intent itself would remain valid. The "closed-loop logic for intent translation" block depicted in the picture could help to improve the efficiency of the translation approach. This situation could be detected elaborating jointly the service and infrastructure KPIs, since this usually impacts at the beginning the infrastructure metrics (e.g., due the high usage of computing resources, traffic congestion, etc.) with a following degradation of the service KPIs under the requested values. The countermeasure, in this case, would include the automated update of the intent translation logic, e.g., modifying the rules to transform the service-level requirements, and the adjustment of the service according to the new rules.

7.2.2.2 Enhanced service description models and profiling

The previous section introduced vertical service blueprints and descriptors as a technology-agnostic manner to define intents, following vertical users' perspectives. In 6G networks it is expected that additional information can specify characteristics, dependencies, or resource constraints for the services. For example, the target placement of the application elements can be extended beyond the edge, towards the extreme-edge, e.g., exploiting the low-power computing resources in IoT gateways, industrial devices or vehicles. The traditional network KPIs can be complemented with service-level metrics, potentially linked to automated actions to be triggered in case of particular alerts raised when the measured metrics and KPIs are out of predefined ranges. Such actions can refer to an update of the service-level requirements or can indicate explicitly countermeasures to be applied at the different layers of the architecture (e.g., network re-configuration at the infrastructure layer, or scaling slices at the network layer). Finally, the service descriptor may also declare explicit service-level AI/ML components and their related requirements, for example, the kind of input data to be consumed, or trained models to be applied, or AI/ML functions and engines to run as part of the overall service. The definition of the AI

components of the service can drive the joint management of resources, virtual functions, and monitoring sources, specifically dedicated to the distributed AI/ML processing, in support of data-driven service management.

From the network perspective, in order to provide the optimal services to verticals, an increase in programmability is required, both for the network and the service elements. The existing models used to describe these elements need to be extended to accommodate these new requirements. One such extension is the information pertaining to profiling the services and network slices.

Increased programmability in the use of the available resources can lead to different operational conditions for a service or network slice. Modifying the number of resources or changing the placement of different elements can have from none to significant impact on the observed performance. The profiling process, leveraging the Monitoring Functions components, and the diagnostic processes, a subsystem of the assurance functions components, would lead to the collection of the required information to generate static profiles, providing an estimate of the expected performance of a service or slice under a set of known conditions. This process could happen under preconfigured static conditions, or dynamically configured based on the available options for modifying typical variables, like network conditions, service traffic load, etc. This information would be inserted into the description of said services or slices, e.g., following the format of Service Profiles or Slice Profiles proposed in the 3GPP Network Resource Model (NRM) [28.541]. As a result, the descriptors would provide a way to quantify the desired state of the service or network slice.

Leveraging on the increased programmability provided by the Management Functions, and supported by the fulfilment control loops (part of the Service Layer Control Loops), the services and slices can become more elastic and, as such, their profiles can be extended to become more elastic as well, considering resource reallocation, element placement, energy consumption etc. During the profiling process, possible internal changes in the deployment of a service or slice, like dynamic reallocation of resources or element placement, could be tested and used to enrich these elastic profiles. Doing so would enable a more accurate estimate of the expected performance based on resource usage trends, and also provides another option to assist in the service quality assurance, in conjunctions with the performance diagnosis, by increasing the options of the fulfilment control loops to satisfy the service quality requirements.

7.2.2.3 Diagnostics processes

Taking into consideration the requirements of the requested services and the offered operations and mechanisms, the importance of systems that provide an overview of the various components and services becomes evident. In Hexa-X, these tasks are handled by the monitoring and diagnostic processes respectively. The diagnostic processes are an integral part of the Management Functions, as seen in the Structural View, and would be specifically implemented by the assurance subsystems. These assurance subsystems would be supported by the service quality management functions and AI/ML-driven orchestration actions, part of the Service and Network Layer M&O blocks (Figure 6-1). They would provide the link between the requirements and the assurance that these requirements are met in the offered services, as the name of the subsystems category suggests.

Existing M&O implementations, like the ones used in current 5G deployments, while effective, in this regard are limited in scope, since they focus mostly, if not exclusively, on the network or the infrastructure layer. In order to handle the diverse needs of future 6G services and networks, and provide a completely automated and programmable operation, a more holistic approach is needed, able to bridge the gaps between service, network, and infrastructure layers. Figure 7-3 shows a possible way of implementation to achieve this new cross-layer approach.

Typically, the first stage of a diagnostic process is the ingestion of data. This would be implemented leveraging the Monitoring Functions components to retrieve the current status and information of the examined deployment. This information would be used to get near real-time (or with a configurable time granularity) snapshot of the system, based on the monitoring system's

capabilities. This information could also be correlated and mapped to the various system components or service elements through the use of the M&O specifications and service descriptions available from the Management Functions, part of the Service and Network Layer M&O blocks in Figure 6-1. Tapping into this data flow, the diagnostic process would be then able to a create a cross-layer picture of the examined service. Utilising this information, the second stage of the diagnostic process (the analysis) would receive its required input. At this stage the raw data provided by the monitoring mechanisms would pass through various types of performance analysis processes in order to extract meaningful information. Commonly used analytics mechanisms for this purpose could be statistical analysis, historical trends, training of AI/ML models, and predictions to detect any anomalies in the behaviour of the system components or service elements, as well as detect any performance degradations. In parallel with the performance analysis, the performance profiling sub-process would be responsible for creating a dynamic profile considering the service components, resources, traffic load, etc. Utilising this mechanism, detection and prediction of issues or bottlenecks could be detected and verified. In the case of successful detection of said events, the third stage of the diagnostic process (the actions decision) would be activated. Based on the observed difference between the current and the desired state of the system or service, and after examining the results of the diagnostic sub-process used to locate the possible issues, such as an algorithm for Root Cause Analysis (RCA), a number of suggestions could be generated, in order to help to minimise or eliminate the delta between these two states. These suggestions could vary, for example, from scaling the software components of a service, to scaling Network Slices, redeploying network elements, or changing infrastructure elements, among others.

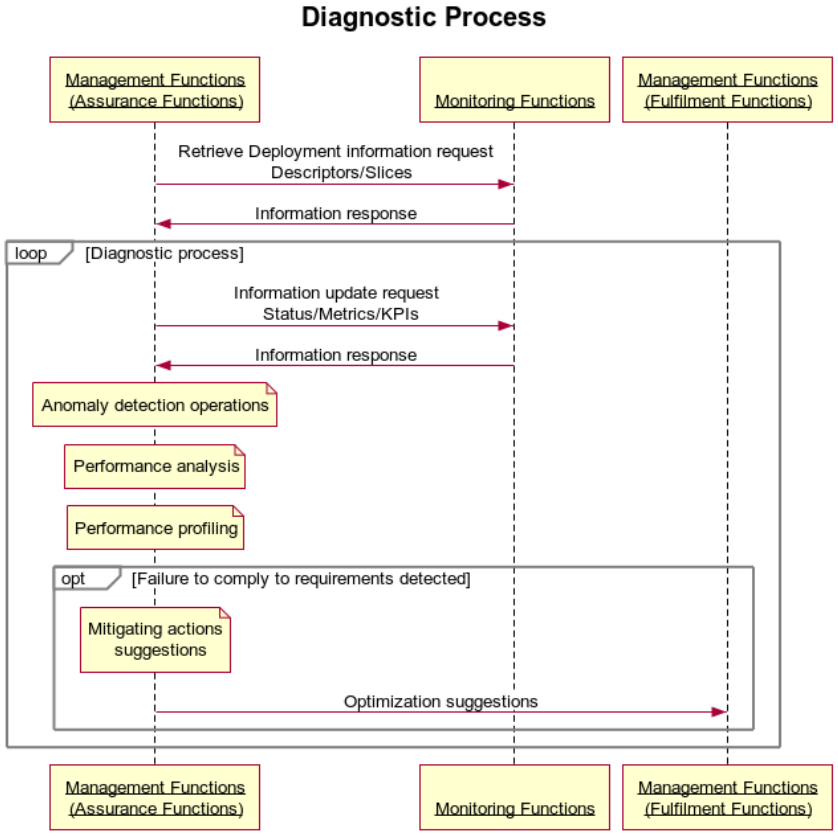

Figure 7-3. Diagnostic Process workflow.

All these stages would be continuous operations running in parallel with components and services as part of a closed-loop. To accomplish that, the diagnostic process should be interfaced with the appropriate mechanisms or components capable of executing the actions decided by the process using a model of requests or notifications. These capabilities would be provided by the fulfilment components (within the Management Functions block). These requests would be to be taken as suggestions for actions in order to alleviate any performance degradation, and would be not

immediately executed, since appropriate validity checks would need to be done beforehand. Depending on any infrastructure, vendor, security, or other related constraints, these requests should be filtered for viability and approved accordingly. In such cases, calibration of these requests or re-examination should be possible.

A possible issue with this approach would be the need for verifiable relations between the components and elements. This is something that could be handled effectively either by the respective M&O systems in each of the various domains, as they are guaranteed to have access to the description of the services and functions by centralised monitoring or by the management functions tasked with the overview of the system. These relations are the information necessary to enable the cross-domain correlation, and the matching of events and performance. Doing so, opens the path to other innovative mechanisms and functionalities, like automated per domain performance optimisation, optimised resource allocation and management, and automated response to unexpected conditions.

7.2.2.4 Programmable network enablers for reasoning

The programmability of 6G networks can be exploited by the network "intelligence" to trigger dynamically several automated actions following the zero-touch and data-driven patterns, as will be further discussed in Section 7.2.3. In particular, reasoning techniques combined with extensive network knowledge representations (e.g., based on ontologies) are getting attention in the area of AI/ML to build semantic learning strategies applicable to 6G networks M&O [CB21]. Still in this direction, ML and reasoning, together with data and knowledge management, are key enablers for cognitive networks that are expected to play a crucial role in E2E 6G architectures [RRB+22].

The creation of a representative knowledge base for 6G networks is one of the challenges to enable an efficient reasoning. In case of reasoning applied to M&O strategies, this knowledge base should capture a wide variety of parameters that may impact the performance and requirements at the service, network and infrastructure layers, since they should be jointly taken into account when taking decisions about network, slice, or service re-optimisation. In this context, the network programmability offers the capability to generate and expose a variety of monitoring data to build such extensive knowledge base, as required by the reasoning entities that consumes them.

Since in 6G networks AI/ML agents, where the reasoning engines run, are expected to be provisioned and re-configured dynamically, the need of monitoring data to feed such agents may vary in time, e.g., in terms of type of metrics to be monitored, frequency of metrics generation and collection, options for filtering, granularity and level of details and aggregation of the monitoring data. Thereupon, programmable interfaces at the infrastructure, network and service layers should be provided to enable on one hand the provisioning of new probes in different segments of the infrastructure and the configuration of heterogenous monitoring data sources and, one the other hand, the efficient retrieval of the collected data in a distributed, scalable and secure manner. The former must be orchestrated through the Management Functions at the various layers of the Structural View in the architecture, in strict coordination with the Monitoring Functions. The latter constitutes the enablers for the AI/ML Management Functions to feed their reasoning logic.

The following categories of parameters and monitoring data can be considered at the reasoning procedures, often mixing them together:

- **Parameters related to service demands and requirements.** These parameters can be declared in an explicit manner with different levels of details. For example, following a service-oriented perspective, the customer can define "intents" to describe the expectations from the network (see Section 7.2.2.1). An alternative could be the adoption of the NEST [Ng116], which provides an intermediate level of definition for the requested Network Slice, declaring its network requirements filling the GST parameters [Ng116] (e.g., uplink and downlink data rates to be guaranteed, their possible peak values, the maximum acceptable latency, jitter, or packet loss, the coverage areas, the level of desired isolation and security, etc.). A further level of request, more oriented to the technicians

and administrators of the network, can include the fine-grained directives for the network configuration, i.e., the number, type and dimension of the virtual NFs to be deployed, their specific configuration, the paths at the TN and their capabilities, etc. However, the service demands are not always declared explicitly, since they may be not known in detail a-priori, or they may change during the service runtime. For this reason, suitable monitoring mechanisms should be adopted, both at the service and network level, to help the reasoning procedures to automatically derive the traffic patterns and profiles, as well as to detect or predict possible changes in order to properly react.

- **Network capabilities, including resource availability and related constraints.** In 6G networks, this type of parameters are dynamic and variable with the context. For this reason, in some cases they cannot be modelled as static and well-defined inputs for the reasoning process, but they must be acquired on-demand or even continuously monitored and updated. For example, due to the multi-domain nature of the infrastructure, some scenarios involve multiple stakeholders cooperating together, where each of them applies their own policies and may offer resources, both at the computing and network level, with different SLAs and costs. Moreover, the presence of extreme-edge resources, with volatile nodes that may belong to a variety of users out of the MNO scope, requires mechanisms for dynamic discovery and, where needed, dynamic negotiation of SLAs. These nodes have also a number of characteristics, like their limited power constraints and their resource sharing with variable user applications, that should be carefully considered by the reasoning logic when composing the E2E service, when taking decisions about potential virtual functions migration, or computing offloading.
- **External factors not directly related to the network itself but impacting its performance or the service demands.** This includes a variety of information that may be collected through very different data sources. For example, the information about traffic congestions in a city may help the reasoning to identify and predict potential mobility patterns for the mobile users. In case of integration with Non-Terrestrial Networks, the weather conditions may impact the selection of the satellite gateways placed in different geographical areas. In Stand-alone NPNs deployed in smart factories, the information coming from the production lines may help to identify the profile of the traffic generated by IoT sensors and devices spread in the various areas of the factory.

All the examples described throughout this section highlight the fundamental role of a scalable, secure, multi-domain and distributed monitoring and data management system, able to collect, aggregate, filter and elaborate data coming from very different sources, to efficiently feed the intelligent decisions of the network.

7.2.2.5 Software Integration Processes

Software integration processes are those basically involving the well-known DevOps processes [EGH+16], such as Continuous Integration and Delivery (CI/CD). As already mentioned in Section 6.2.4, those processes are already widely applied in the cloud-native based systems, but are not so extended in the telco-grade industry.

However, integrating development and operations, as it is already done in the cloud-native industry, brings clear benefits. E.g.:

- Helps to get improvements in software quality by enabling frequent releases with new features, fixes and/or updates. Close collaboration between operational and development teams eases frequent capturing of user feedbacks, which can lead to rapid improvements in the quality of the deployed services.
- Provides a high degree of automation for repetitive management tasks.
- Provides fast and reliable problem-solving techniques.
- Improves MTTR (Mean Time To Recovery).

DevOps processes in the Hexa-X M&O architectural design would be implemented through the specific Design Layer included in the architecture (see Figure 6-1). These processes are closely

related to the M&O tasks, since they can trigger the basic orchestration actions in Section 7.1, i.e., service instantiation actions, configuration actions, scaling actions, etc. In the following list, the main DevOps processes that could be implemented through this Design Layer are described:

- Continuous Integration (CI). This is the main DevOps process, referring to the automation of the initial development stages. Different developers can simultaneously work on a common developer's repository to bring together their contributions to the code development. To improve efficiency, the development tasks are usually split in small fragments of code, which are being "continuously integrated" (hence the concept).
- Continuous Testing (CT). meaning that multiple test batteries are executed on the provided code, to ensure it meets specifications. The process is automated, so it can be also "continuously" triggered.
- Continuous Delivery (CD). This process is the logical next step after CI. In few words, it means that changes validated in the developer's repository would be transferred to a centralised artifacts repository in the operational environment, where the set of all changes would be continuously tested again and verified in a production (or production-like) environment.
- Continuous Deployment (Cd). This also refers to the logical next step after CD. Typically, artifacts generated during the CD stage are stored in a repository until somebody decides to deploy them in a production environment. However, this could be done automatically, and this automation is what is referred by this "Continuous Deployment" process, i.e., it refers the automatic deployment without (or with minor) human intervention. This enables direct delivery of new features to end-users. Nonetheless, Cd processes usually decouple deployment from activation by using the so-called "feature flags", which makes possible to activate certain functions only under controlled situations.
- Continuous Monitoring (CM). Refers to those monitoring processes oriented to use real production data in order to guide development and operational teams. Automation also has a meaning within this concept: autonomous responses to certain metrics or alerts can be implemented, instead of relaying only on human responses/reactions. Besides, automated CM enables the scaling of NSs without human intervention according to certain QoS/QoE metrics.

Figure 7-4 shows these five concepts (CI/CT/CD/Cd/CM) working together in a graphical way. Although other processes could be considered, these four are the main processes related to DevOps practices. They showcase how the DevOps approach breaks out the barrier between development and the operational scopes: although CI is still on the development side, CD and, especially, Cd and CM are clearly beyond the developer's scope, entering clearly in the operations area.

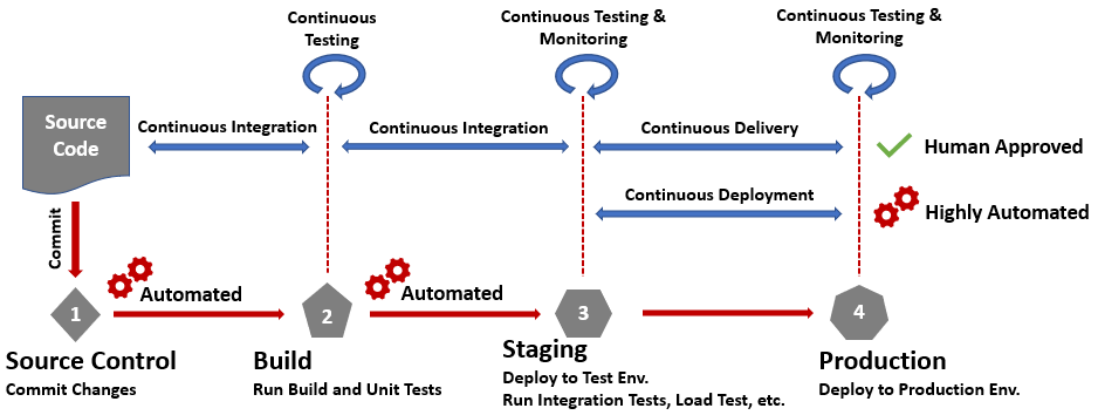

Figure 7-4. Continuous DevOps processes.

Implementing this in a telco-grade environment is highly challenging, since development and operational scopes are usually totally separated: MNO and software providers are typically different companies, with different interests and corporate cultures. Implementing this concept in

such environment will probably require addressing not only technical challenges (e.g., to enable shared repositories and open interfaces), but also cultural and procedural changes in the involved companies (similar to those already implemented in IT environments).

7.2.3 Automation processes

Automation makes possible to execute a defined set of tasks with minor or even no-human intervention. Automation is a natural consequence of programmability. Once a set of processes can be algorithmically programmed, it is possible to generate automatic tasks based on them. In this section, the automation processes that have been identified as the most relevant for 6G networks are addressed, including:

- Zero-touch automation processes (Section 7.2.3.1)
- Autonomic computing processes (Section 7.2.3.2)
- Closed-loop automation processes (Section 7.2.3.3)
- Automation processes in multi-stakeholder scenarios (Section 7.2.3.4)
- Processes for dynamic self-optimisation of network slices (Section 7.2.3.5)

7.2.3.1 Zero-touch Automation

The technology that underpins NSs is becoming too sophisticated for manual processes to be run effectively. Automation is required to support orchestration, and to accommodate the rapidly changing requirements associated with NFs and services. B5G and 6G systems are posing unprecedented engineering challenges in terms of disaggregation, complexification, scalability, and versatility. Automation in the most general terms, including machine learning, is considered as an essential ingredient for addressing some of these challenges. Even this could be done via scripting, the Zero Touch Automation removes the human intervention in handling the full life cycle of networks and the exposed services across multiple business and technology domains. Using closed loops and advanced machine learning, Zero Touch automation detects anomalies and provides the appropriate remediation.

This can happen through the new architecture design of closed-loop automation and embedding intelligence with data-driven AI/ML algorithms, which are the key enablers for self-managing capabilities, with lower operational costs, accelerated time-to-value, and reduced risk of human error. This closed-loop includes the process of collecting monitoring data from the services and networks, performing real-time data analytics for identifying events to handle, and taking proper decisions for optimisation and re-configuration of the system, such as auto-scaling, self-healing and fault-tolerance, anomaly detection and automated troubleshooting, automated authentication, and traffic management.

The generic automation framework components are summarised below:

- The monitoring part (provided by the Monitoring Functions in the Structural View). This is the framework to set probes and collect their measurements. The monitoring framework would be used by the automation framework, but not part of it. They are independent to facilitate the evolution of each without impacting the other (as long as their service API is unchanged).
- Data analytics/mining (part of AI/ML Functions block in the Structural View): extracts "meaning/insights" from raw data coming from the Monitoring Functions (and other sources, e.g., data lakes or historical data). This component is able to "understand" what is happening to the controlled system (from simple reactive threshold passing to classifications, predictions, complex correlations, etc.).
- Automation/Policy engine (part also of the AI/ML Functions block), also called Policy Decision Point: it would be in charge to select and trigger/request the automation action given what could be happening in the system (i.e., considering the input from the data analytics component). It may keep a state machine (i.e., state/event → action(s) → new state) and the policies (or may retrieve them from a DB).

- Automation/ML model marketplace (part of the design layer): this is where the models would be stored for the user's to pick them up to be used (alone or composed in complex chains).

7.2.3.2 Autonomic Computing processes

Autonomic Computing (AC) was a term coined by IBM in 2001 [Ibm01], although the concept was already known before this date. It originally refers to the biological domain, where many actions can be performed without any conscious intervention. Translated to the computing realm, it consists in designing an Autonomic Manager (AM) able to change the behaviour of a system, being the change driven by environmental awareness and high-level policies. Compared to manual management, AC makes it possible to process a large number of events, to provide fast and reproducible responses, thus freeing up time for humans to focus on high value-added tasks that cannot be easily automated. Moreover, AC can make prediction-based decisions to further reduce response times.

To carry out feedback actions, AC systems rely on the following two pillars: context awareness and self-awareness [AO17]. Context-awareness refers to the capability of the system to gather and consider the state variations of external entities. Self-awareness is composed of a set of so-called "self-*" properties; although different "self-*" properties can be defined [AO17], the four initially formalised by IBM are the following [Ibm05]:

- Self-configuration: To apply self-modifications in response to changes in the environment.
- Self-healing: To detect, understand and react to internal failures.
- Self-optimisation: To dynamically track and adjust resource allocation to achieve better performances.
- Self-protection: To enforce own cybersecurity measures against threats.

In the proposed Hexa-X architecture these properties will be enforced by the M&O system onto the services they manage. In this case the M&O system itself would effectively act as an AM. To enforce these properties the Hexa-X M&O system would rely on different functions in the Network Layer. In particular, it could use the Monitoring Functions to be aware of the state of the managed services, or any other element of relevance in the environment. It could also decide on the course of action using the AI/ML dedicated functions, and apply determined actions relying on the Management Functions. This system could be applied to any self-awareness property.

However, to efficiently integrate AC in future 6G networks, there are still remaining challenges to overcome [HM08]. Although the main function of the AC is to dynamically adapt the system to changes in the environment or policies, the overall system it supports should move towards a stable state. For example, state-flapping avoidance mechanisms should be proposed to prevent the system from oscillating between states as a result of an inability to determine the best state to align with the policy. It should also be possible to evaluate the performance and effectiveness of the AC instance without necessarily aiming for the optimal solution, for example by evaluating the benefit-to-cost ratio of the responses to a stimuli that could maintain a sustainable functioning of the customer's services. As the human operator delegates part of his management tasks to the AC, it would be necessary also to create the conditions for trust: on the one hand, they would consist of the understanding of the functioning of the system as a whole, i.e., the behaviour of the AC in response to a given stimuli, or the ability of the system to adapt itself in an expected way. On the other hand, by relying on a non-centralised trust model between components, more suited to a distributed environment.

Another important consideration is that AC is unlikely to operate in a centralised form. Rather, they would be part of an overall multi-layer, multi-domain system containing several AMs that would need to work together, which leads to a set of active research topics. This includes two important issues that future 6G networks should address: heterogeneity and scalability. Heterogeneity refers here to the diversity of domains that are involved in the delivery of valuable

services to the customer, e.g., including domains such as RAN, transport network and core. In each of these domains, several technologies should be implemented, such as multiple RATs (Radio Access Technologies), and different generations and flavour of cores, depending on the vertical requirements. NSs could be instantiated and run by means of various resources which could be physical or virtualised. Driven by constraints derived from customer needs, NSs would be deployed at various locations, including the edge or the extreme-edge. This decomposition in multiple deployment domains calls for a distribution of specialised AMs, interworking together. These AMs could be organised vertically or horizontally [AO17]. Horizontal separation would be intended to provide separation of concerns, or separation between tenants (two systems providing either very different services, or services to different tenants, may not be handled by the same AM). Vertical separation, on the other hand, would help to cope with scalability, which still remains as an ongoing research topic [AO17] (it would allow local AM to take fast and locally accurate decision, while higher level AMs would ensure E2E management). The distributed AC system would also raise another issue: the isolation between domains. Isolation should primarily ensure that the resources reserved and used by an AM would not be used by another AM [BBR+16], since this could jeopardize the quality of the service in the network. It should be also ensured that no conflict could result from the policies pushed in the AMs, whether between two policies in a single AM, or between two AM following their respective policies [BBR+16] [HM08]. The resolution of policy conflicts should lead to minimal lost in performance for all concerned domains. Within the Hexa-X M&O architectural framework, this de-centralised paradigm would imply that AC systems located in the different M&O blocks should be connected together, both between layers and within the same layer.

Although AC is a generic concept, it is strongly expected to be applied to 6G systems via a closed-loop architectures hosting AI/ML components. As such, AC would also inherit the different challenges and limitations of those closed-loops, which are detailed right in the Section 7.2.3.3.

7.2.3.3 Closed-loop automation

Closed-loop automation is a well-known concept from the control systems theory. The first formal analysis on this topic is commonly attributed to the pioneering work from J.C. Maxwell back in 1868 [Max68]. In a very general way, closed-loop control systems are those in which a feedback signal is back-propagated from the system output towards the input, so that the system output is taken into account to perform the control actions. Closed-control systems are also known as *feedback* control systems and are typically represented as shown in Figure 7-5, with a *closed loop* connecting output and input.

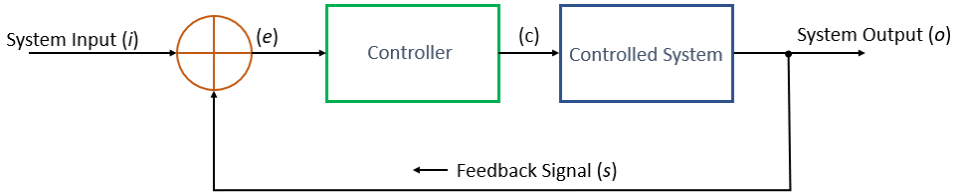

Figure 7-5. Closed-control loop automation system.

Closed-loop control systems are widely used in a variety of engineering fields (e.g., electronics, mechanics, hydraulics, robotics, etc.). A typical easy-to-understand example could be an electric water heater, in which the water temperature measurement (the system output) would be used as a feedback signal to activate/deactivate an electrical resistance used to heat the water.

Opposed to closed-loop control systems, there are also open-loop systems (simpler and generally less precise), in which no output-to-input feedback is provided (following the heater example, an open-loop system could be implemented using a timer to activate the heating resistance for a fixed time, i.e., without a direct measure of the variable to be controlled -the water temperature).

Probably the most-used feedback control systems are the so-called Proportional-Integral-Derivative (PID) systems [ACL05], where the control system continuously computes an error value (*e* in Figure 7-5) as the difference between the desired and the measured output, applying

corrections based on the so-called proportional, integral, and derivative values. In short, the proportional term represents the present error, while integral and derivative values represent past and possible future errors. These systems were typically implemented using analogue electronic systems, but the progress of modern control engineering made such systems increasingly implemented using computing resources, which help to implement improved control strategies even beyond the PID model itself, based on increasing complex algorithms (e.g., to consider multivariable, nonlinear, or predictive control processes, which could require more advanced control techniques).

In the telco industry, closed-loop control systems are also being considered an integral part of M&O systems for future 6G networks, focusing on the automation capabilities these closed-loop systems offer. Particularly, the closed-loop model is a possible framework to implement the AC model introduced in the previous Section 7.2.3.2. Although the principles are basically the same, in this specific context the vocabulary and details may differ, referring typically five basic functionalities [Ibm05], namely: monitoring, analysis, planning, execution, and (optionally) knowledge.

Monitoring refers the collection and correlation of data. *Analysis* refers the process of receiving this data and, on the basis of policy and knowledge, determine whether something should be done. The *Planning* functionality develops the procedure that will lead to the desired change. The *Execution* follows this procedure and applies it to the controlled system. Finally, the *Knowledge* represents a database that can be used from the other functionalities. This last element is only present if the closed-loop activity requires to maintain such database, typically for context-awareness. Data stored in the knowledge DB can be provided by an external source, retrieved from an external source, or produced by the components of the closed-loop themselves.

The structural view of the Hexa-X M&O architecture, in fact, includes four different sets of loops, that follow the above functional flow. As mentioned, each of these three loops leverages the corresponding interfaces to perform the monitoring and execution phases. Each of these loops has a different operation timescale (e.g., the infrastructure typically has a longer reaction time than the network), which impacts the design of the corresponding analysis and planning.

In the context of the structural view proposed in this document, the closed-loops would be implemented using functions of the Network Layer. The exact functions used may vary from closed-loops to closed-loops, and depending on the functionality the closed-loop is supposed to provide. But typically, closed-loops will use Monitoring Functions for the monitoring functionality and may rely on AI/ML Functions for analysis and planning. Execution may rely on the Management Functions. All of these generic functions may be completed with ad-hoc functions, depending on the exact purpose of the closed-loop.

Although the closed-loops have been used in many networking scenarios, several challenges remain, and it is important to address them for future 6G systems:

- With regard to monitoring, data collection itself can be difficult. In a large and heterogeneous system, it is not easy to adapt to each data source and continuously report data monitored (see Section 6.2.2.3).
- In addition, optimising NFs may lead function providers to merge functions (reduced stack) in order to improve performance, which may result in the removal of a standard interface and the associated monitoring data, therefore resulting in proprietary (vendor-specific) approaches. Conversely, where data is available, it may be too abundant, typically based on collecting everything and dispatching in a large data lake for their later processing and usage, if applies. For this reason [HM08] suggests that the monitoring itself should be an autonomous system, capable of adapting its monitoring rate, and the nature of the data being monitored, to report only meaningful information.
- The execution stage should have an efficient mechanism to call/invoke the relevant APIs (which can be dynamically discovered) to apply the procedure sent by the planning stage and receive the appropriate identification information.

- The use of AI/ML typically involve relatively long training periods. Furthermore, certain approaches are based on a continuous exploration of the configuration space (e.g., reinforcement learning – see Section 7.2.4.2), which might be inadequate for e.g., URLLC scenarios imposing very stringent delivery guarantees (the cost of exploration can be too high).
- The optional usage of AI/ML in the context of 6G by different functionalities of the closed-loops brings its own challenges, which are described in mote details in Section 7.2.4.2.

In addition to these internal challenges, a closed-loop must be able to interact with other closed-loops, which will be explored in more detail in the next Section 7.2.3.4.

7.2.3.4 Automation in multi-stakeholder scenarios

As explained in Section 7.2.3.2 closed-loops are unlikely to run in isolation. An isolated closed-loop in a complex system provides very limited benefits [Ibm05]. In a system owned by a single entity, the multiple closed-loops may communicate together via proprietary, ad-hoc means. Like the 5G system, the 6G system is expected to be divided into multiple stakeholders. As a consequence, automation means in 6G must be able to deal with multi-stakeholder scenarios. The different stakeholders envisioned to coexist in the 6G system are detailed in Section 5.1. In addition to the different roles that will exist in the system, it can also be noted that a single role, such as Public Network Operator, can be taken by different, competing stakeholders. Although each stakeholder could have its own specificities, each one could consumes resources provided by other stakeholders, and in turns provide services to other stakeholders. The key point in this situation resides in the definition of interfaces between those stakeholders.

Regarding closed-loops [HM08] points out the interoperability as an important research challenge. [AO17] precise that such interoperability would require an adapted infrastructure and standardised knowledge bases, so that closed-loops would be able to exchange information and exploit the exchanged information. Similarly, [Ibm05] proposes a common blueprint on which closed-loops could be built, in order to enhance interoperability. To generalize, the different closed-loops of the different stakeholders should be able to communicate via each other interfaces to collect information and require services. The knowledge of the existence and nature of an interface may be either known beforehand by a closed-loop, either because the connexion is standardised or via configuration, or discovered via a dedicated mechanism.

The nature of the information exchanged between closed-loops is however a challenge as well. Indeed, in a multi-stakeholder case the different closed-loops may be reluctant to share their raw data with each other, and sensitive data must remain secret. As a consequence, the inner functions of closed-loops, mainly the analysis and plan modules, should be able to work without using all the information available in other closed-loops. Examples on this topic are provided in Section 7.2.4.2.

Finally, in a multi-stakeholder scenario, the importance of isolation between stakeholders, which is detailed in Section 7.2.3.2, gains even more importance, as one stakeholder actions must not impact another stakeholder.

The Structural View proposed in this document supports multi-stakeholders' scenarios by means of the API Management Exposure block, which makes possible to connect the MNO with other stakeholders (Section 6.2.3), each one managing its own set of resources. Of course, M&O resources should be designed to run optimally in each scope, following the principles detailed above, and implementing standardised interfaces to exchange standardised data, including monitoring, analysis, and actions. The exact implementation of this block may vary depending on the case. In a classical public 6G network, potentially involving large number of functions coming from different vendors, an interesting dynamic way to expose services to API is to use an API discovery system. In this case, the functions used would include a registry, where service providers could register their APIs and service consumers could discover them. Additional

functions and mechanisms can be added to provide security services, including access control or redundancy.

7.2.3.5 Dynamic self-optimisation of network slices

The future mobile networks are expected to handle a vast number of network slices with divergent requirements in terms of hardware, compute, and connectivity resources. In order to reduce the infrastructural expenses and increase the revenue, it becomes substantial for the network operators to incorporate mechanisms enabling dynamic optimisation of network slice operations. The optimisation should include scaling the resources according to the current needs, expressed typically in form of KPIs. Moreover slices should have self-healing and self-configuration features supported by the M&O processes. The Hexa-X approach assumes optimisation in an E2E and multi domain manner. It implies integration of multiple M&O systems that in the Hexa-X approach are organized in an E2E and in a multi-layer M&O distributed architecture, with functions that can exchange the information essential for E2E slice optimisation using the API Management Exposure.

The assumed data driven M&O approach allows for proactive operations that would contribute to significant reduction of KPIs/SLA violations, and more efficient allocation of resources during slice lifetime. The operations related to dynamic optimisation of network slices includes:

- Slice-related data collection and pre-processing. The data are collected from all layers of the system (infrastructure, slice functions, configuration parameters, number of users, etc.). These collection and pre-processing operations would be executed by the Monitoring Functions block.
- Extracting from the monitoring information (also provided by the Monitoring Functions block) relevant features for slice optimisation. That includes analysis of anomalies and prediction of future values of time series, prediction of service demands, and KPIs. To this end the AI/ML Functions provided by the Hexa-X M&O framework could be used.
- Performing slice self-optimisation algorithms. Such optimisation includes efficient re-allocation of resources, orchestration of additional functions, removing of not necessary slice functions, migration of virtual functions, and scaling of resources. The AI/ML Functions provided by the Hexa-X M&O framework could be also used for the mentioned operations. The main goal would be to optimise resource consumption while keeping slice KPIs at the required level, fulfilling SLAs.
- Performing slice self-healing and self-configuration actions in a fully automated manner, enabling an efficient scaling of the solution with the increasing number of concurrent network slices, and removing the need of the human intervention to operate them.
- The overall resource allocation scheme should keep slices isolated from each other, in the sense that congestion in one slice cause minimal (if not zero) impact on other slices.

7.2.4 Data-driven processes

6G networks are expected to have cognitive capabilities and abilities in terms of autonomous operation, optimisation of operation, and prediction of future network state. In this context, undertaking an efficient data management strategy becomes substantial to facilitate all data-driven processes conducted in the network. The Hexa-X data management strategy defines:

- **Data architecture** – the formal structure of data flows management and processing.
- **Data collection strategy** – defining the scope of network components monitoring (NF metrics, user activity), monitoring data abstractions (performance and quality indicators), data pipelines, etc.
- **Data storage and provisioning** – creation of data aggregation points (warehouses) and components collecting metadata (data catalogues) for easier processing and management.
- **Data governance** – creation of data policies and procedures for data security (ownership, privacy, access), compliance and integrity maintenance.

- **Data security** – the processes to protect the collected monitoring data from corruption and limit access to privileged entities.

Considering the scope of the 6G expected capabilities, the Hexa-X M&O system will perform data-driven processes utilising the data originated at multiple network levels (slice, service, infrastructure etc.). To this extent, it is substantial to define the mechanisms for cross-layer monitoring. The data-driven processes would be used for the optimisation of the slice as it has been described in the previous subsection 7.2.1.2.

This section describes how Hexa-X sees the most relevant data-driven orchestration processes, including:

- Monitoring and handling processes (Section 7.2.4.1).
- AI-driven orchestration processes (Section 7.2.4.2).
- Security related processes (Section 7.2.4.3).

7.2.4.1 Monitoring and handling of data

The process of data collection can be considered in many dimensions:

- Time: real-time or deferred-time collection.
- Session-initiating party: source-initiated ("push" model) or destination-initiated ("pull" model) – cf. subscribe/notify or request/response mechanisms.
- Spatial: centralised or distributed collection (in case of the latter, multiple levels of data sets aggregation and pre-processing may exist).
- Triggering mechanism: spontaneous (usually source-event driven, and source initiated), on-demand (requested by the destination party) or scheduled (the source has to prepare data for collection according to a fixed time-scheme, usually for each consecutive pre-set equal period of time).
- Data characteristics: batch/unitary, structured/unstructured, formatted (i.e., structured according to a defined format)/unformatted, numeric/non-numeric (arithmetically processable or other), raw/enriched (e.g., with timestamps, source labels), time series (e.g., network events, counters values)/other (e.g., lists of object IDs mapped onto their descriptive labels), etc.

In Hexa-X, the monitoring and processing of monitoring data is implemented in the Service Layer, the Network Layer and the Infrastructure Layer. Each of the mentioned systems has implemented monitoring processes that support FCAPS (Fault, Configuration, Accounting, Performance, and Security) functions:

- Monitoring for proactive (anomaly based) or reactive (alert based) fault detection, support for RCA.
- Monitoring for discovery and self-configuration of functions, nodes and services.
- Collecting of accounting related information.
- Monitoring of the performance, calculation of KPIs.
- Monitoring of thresholds' crossing.
- Security related events detection.

The overall monitoring, in order to handle E2E optimisation, exchange also information between layers and external domains mixing application and infrastructure-based cognition. It should be noted that one of the most important factors characterising the data collection sub-system is its scalability and the principle of collection, processing, and utilisation of data as near to the source as it is possible, which leads to the distributed, layered monitoring architecture. In that context it is worth to mention the CAP theorem [MJG19] which states that only two out the three features, namely consistency, availability and partition tolerance can be provided. In the mentioned case, however, the problem is not crucial as the data sources are distributed geographically, so no overlapping partitions exist. and To increase the feasibility and efficiency of management data preparation, collection, processing, and utilisation as well as to lower the demand for resources needed for these jobs, the "filtering at source" approach is preferred (e.g., recipient-requested

creation of performance management jobs or fault related notifications subscription, cf. European Telecommunications Standards Institute (ETSI) NFV [Ifa005] [Ifa006] [Ifa007] [Ifa008] [Ifa013], 3GPP [28.545] [28.550] [28.552] [28.554]. In result of the distribution and calculation of some locally relevant indicators there is a need for the calculation of E2E indicators. In Hexa-X this is the role of Monitoring and Management Functions in the Network Layer (see Figure 6-1).

7.2.4.2 AI-driven orchestration

Mobile networks are already becoming increasingly difficult to manage due to the growing heterogeneity and complexity of the network and the multiplicity of parameters that can be configured in an attempt to achieve optimality. Such complexity, coupled with the large amount of data available, tends to shift the management and optimisation of networks from a traditional model-based approach to a data-based approach [WRS+20].

The motivation for using AI/ML in general, not only for NSs M&O, is because some tasks are not easy to program using traditional programming techniques. This may be for different reasons, e.g.: because regular non-AI/ML algorithms could not be precisely defined, because there could be hidden relationships in large amounts of data, or because of changing properties and unforeseen environmental conditions. AI/ML can help with these types of tasks by finding hidden structures and patterns in data or by creating useful approximations to the problems to be solved.

AI/ML-driven processes are expected to be very effective in managing several aspects of 6G networks due to its ability to efficiently tune a large number of configuration parameters based on the analysis of huge amounts of data. The potential of AI/ML in this area has been analysed in many recent studies, such as [BT20], [WRS+20] and [GSR+21]. Also, with the emergence of edge computing, 5G network resources are quite dispersed. This phenomenon is expected to go even beyond in 6G networks, with the introduction of the extreme-edge resources. In this context, AI/ML is also expected to address the complexity of managing the lifecycle of NFs, including their placement almost anywhere in the network based on performance requirements and resource availability, load change predictions, NF configuration, NF reliability assessment, and QoS monitoring. Also, through efficient resource management, AI/ML algorithms could provide additional support to MNOs to dimension network slices, while respecting the requested QoS.

Machine learning algorithms can be classified into three major types, namely: *supervised*, *unsupervised*, and *reinforcement* learning algorithms. There are other approaches (e.g., semi-supervised learning, self-supervised learning, multi-instance learning, stochastic learning…) but they typically are variations based on the three mains [5gaiml21]. In the field of distributed networks, a specific learning model known as *Federated Learning* [MMR+17] [YLC+19] has been also developed, which can also be supervised, unsupervised, or reinforced, but with specific application to distributed networks. Many of these algorithms are inspired by behavioural aspects of living beings or by models inspired on their nerve cells functioning (these are called *artificial neural networks* – ANNs [WS88]); other approaches are inspired by the species evolution (e.g., *genetic algorithms* [Mit96]), and others by more abstract mathematical models incorporating the concept of learning (in the sense of self-adaptation) in some way. In the following the main features of the most relevant types mentioned above are described, considering how they could be integrated in M&O processes.

Supervised Learning

Supervised learning algorithms [HTF09] are able to compute a mapping function $\boldsymbol{y} = f(\boldsymbol{x})$ from examples, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are n-dimensional vectors. For the algorithm to be functional, a *training* phase is necessary, which, depending on the nature of the data (vectors $\boldsymbol{x}$ and $\boldsymbol{y}$), can be more or less time-consuming (typically ranging from a few seconds for simple problems to even days for very complex problems). This *training* process lies on presenting to the algorithm a collection of **x**/**y** values (training data) in a repeated way. Using these sets of data, the algorithm is able to compute the mathematical function mapping the **x**/**y** values. I.e., what the algorithm learns is just

something like "to this specific input ***x*** corresponds this output value ***y***" being ***x*** and ***y*** arbitrary vectors. Although it could seem simplistic, the power of this approach lies on that:

a. These vectors could represent many different things (e.g., images, sequence of symbols, time-series, infrastructure metrics, end-user movement records, etc.).
b. It has been shown that one of the most widely used supervised algorithms (the multilayer perceptron -MLP-, a specific type of artificial neural network [RGH+86]) can work as a *universal* function approximator, i.e., the MLP can approximate even non-linear relationships between any input and output data set, so this approach can be used to address many different sorts of problems.
c. Supervised algorithms are typically able to *generalize* once trained, i.e., beyond just computing the mapping between the specific **x**/**y** values provided during the training, they can also provide adequate ***y*** values for ***x*** values that were not presented to the algorithm during the training stage. This is often referred to as an *emergent* property of such algorithms.

As a natural extension of this function-fitting (regression) feature, supervised learning can also be used for addressing pattern classification problems [KL90] [KSH12], i.e., they can be used to decide if a certain input vector belongs to a certain class. In this case, ***x*** would be the data to be assigned to a class, and ***y*** the class to be assigned to. E.g., from a human image set, deciding which of them could represent adults or children.

Supervised algorithms are quite mature in the SotA. There are many different specific implementation algorithms (e.g., logistic regression [CN06], backpropagation [RGH+86], support vector machines [TI naive bayes [TVS10], the k-nearest neighbour algorithm [SEJ15], decision trees [FB97], among others) that can be effectively used for different purposes such as patterns recognition, time-series forecasting, images processing.

Figure 7-6 shows how the overall workflow for deploying a supervised learning algorithm in the operating environment of an MNO may be represented. But, as it can be appreciated, this is not in fact very different to the steps that are typically performed with regular non-AI/ML software, in terms of deployment. For steps from 1 to 3, with the non-AI/ML software, instead selecting the ML model, the training, etc., it would be probably necessary to select the proper technologies to address the problem (e.g., a programming language, a working framework…) and issue a programmer's team to understand the problem and develop the algorithms to solve it. Steps 4 and 5 are basically the same for both AI/ML and non-AI/ML solutions: to deploy into production and to iterate based on the performance or possible bugs that could be found.

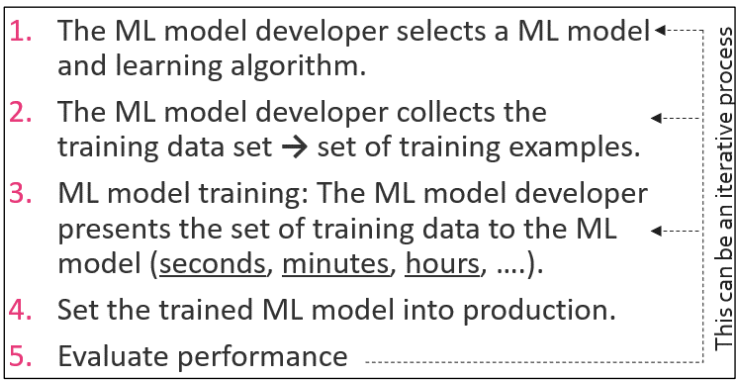

Figure 7-6. Steps to deploy a supervised learning algorithm.

This means that supervised learning workflows can be implemented in a very similar way as regular software development workflows are performed. In this case this would be implemented leveraging on the CI/CD pipelines through the Design Layer (Figure 6-4 – Structural View), that will be used to deploy this kind of NFs as part of the AI/ML Functions block.

However, it is true there is a major difference: data. For developing regular NSs in the traditional way, software developers typically do not need to access real data from the production

environment. Typically, it is enough if the MNO or the Vertical tells them what they expect from the service at the functional level, and they can start working from that. It is true that it will be necessary to define the format in which certain data should be to ease the software integration, but this is not usually a critical aspect, being typically fine-tuned at the end of the development phase, during the testing stages on the pre-production environments. But supervised learning algorithms are "built" based on specific data samples. It is true that initial approaches could be done using synthetic or simulated data, but at the end it will be necessary to provide real data to train the models and produce ready-for-production algorithms. This means that for developing some applications real data should go from the MNO (or vertical) scope towards the Design Layer[3]. To solve this dependence from data specific data pipelines should be implemented connecting the development and the operational scopes (e.g., by means of continuous monitoring pipelines). However, this should be done considering the necessary security aspects and data privacy (e.g., relying on anonymisation techniques).

Unsupervised Learning

As seen in the previous subsection, supervised algorithms work on data pairs: the ***x*** and ***y*** vectors mentioned before. In that case the "supervision" means the ML model developer informs the algorithm to which ***x*** maps each ***y*** in the training set. To explain this, it is usually said that supervised algorithms work with *labelled* data, in the sense that the AI/ML model developer must provide a *label* (the ***y*** value) for each ***x*** value in the training data set. Well, in unsupervised learning there is no such *labelling* of data, i.e., there are no pairs of ***x/y*** values for the training. The AI models developer just provide a set of data to the algorithm and expects it to recognise them properly.

A good example to understand this is the Teuvo Knhonen's phonetic typewriter [Koh88], which was one of the first attempts to address the problem of the speech recognition using artificial neural networks. In a few words, the model consisted of a densely connected artificial neural network to which input vectors representing phonetic sounds were applied. After the training, it could be seen that certain neurons (or groups of neurons) were activated only when certain phonemes were pronounced, which could be used for mechanical speech transcription by assigning the activated neurons to the corresponding written symbols. I.e., compared to the supervised learning, the mapping here is performed *a-posteriori*, i.e., once the neural network has been trained the phoneme-symbol assignation can be done.

However, this is just an example: like supervised learning algorithms, unsupervised models can also be feed with different type of data representing images, sounds, time-series, etc. Also, like supervised models, unsupervised algorithms can also generalize well (e.g., once trained, the phonetic typewriter could properly classify phonemes although they were pronounced slightly differently from how it was done during training).

Unsupervised learning is used, mainly, for two kinds of problems:

i. To find a "better" representation for data, where the term "better" here depends on the application. Possible better representations could include to reduce data dimensionality (the phonetic typewriter would match here), invariant representations, or sparse representations (or a combination of these). There are various types of algorithms for finding new representations, including simple statistical methods such as the principal component analysis [Jol02], sparse methods [HSK13] or auto-encoding neural networks [VLL+10]. Practical applications regarding this are features extraction, noise reduction, or hidden patterns discovery.

[3] If a thought is given to this topic, what has happened here is that the machine learning algorithm is somehow replacing the human programmer. However, to learn, the ML algorithm need specific data samples, which in many cases must be extracted from the environment in which it must operate.

ii. For data clustering [Jai08], where the data is grouped into *n* different clusters according to their features. Clustering methods can be extended to not only find clusters in data structures, but also to create hierarchies. This allows the dataset to be better understood and visualised in a way where similar data are assigned to the same cluster. In services M&O clustering can also be used for anomaly detection: assuming that most of the data is normal and that the anomalies are qualitatively different from the normal data, an unsupervised algorithm could be trained to inform M&O teams about potential anomalies.

Unsupervised learning algorithms are a quite mature SotA technology. There is a wide range of algorithms available, such as the self-organised memories (SOM), adaptive-resonance networks (ART), Hebbian learning networks, K-Means networks, Hopfield networks, or self-associative memories, among others (see [SA13] for more details).

Figure 7-7 shows the overall steps to deploy an unsupervised learning algorithm. As it can be seen, compared to supervised algorithms, there is an additional step on which the ML model developer must analyse the model's response before integrating it into the production environment. The problem regarding accessing data already mentioned in the previous supervised algorithm case also exists here, so the necessary DevOps workflows should also be implemented between the MNO (and/or the verticals) environments, and the software development environment.

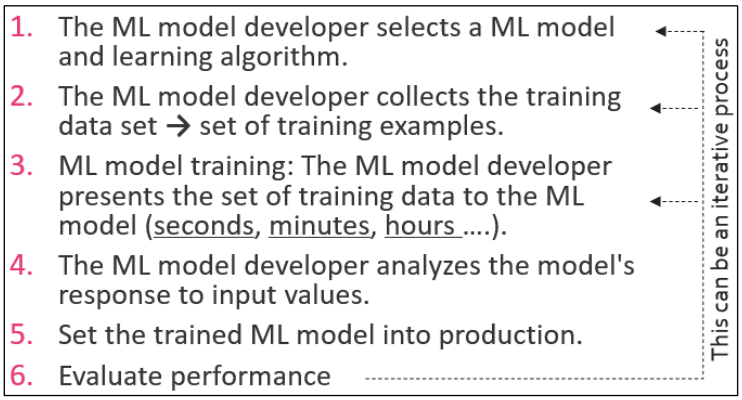

Figure 7-7. Steps to deploy an unsupervised learning algorithm.

Reinforcement Learning

Reinforcement learning (RL) is based on a general concept taken from behavioural psychology: operating conditioning [SB18]. In short, it can be defined as a learning process in which the actions of a subject can be modified by following them by the appropriate positive or negative stimuli, targeting to *reinforce* positive behaviours or inhibit unappropriated ones. By the repeated application of positive and negative stimuli, the subject learns, resulting in a change in the behaviour.

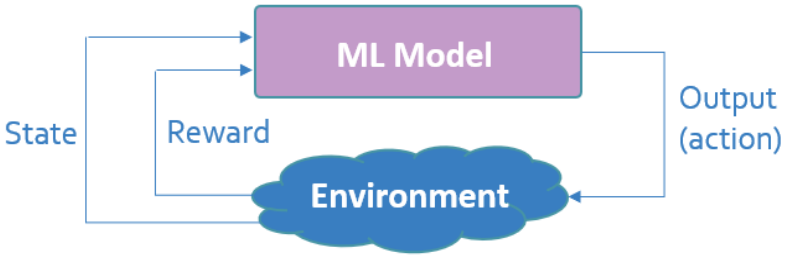

Figure 7-8. General Reinforcement Learning Framework.

In Machine Learning this concept is typically represented as a closed control loop [SB98], where the ML model acts as the subject able to: (i) receive state information from a defined environment, (ii) perform specific actions on it, and (iii) receive the corresponding rewards (reinforcement

signals) based on those actions (see Figure 7-8). By assigning specific state values and by evaluating the reward signal for each action, RL models can iteratively improve its actions. RL models are particularly useful in the absence of a clear mathematical model defining the environment.

The purpose of RL is for the ML model to learn an optimal (or near-optimal) state-action mapping (known as a policy), maximising the reward which is expressed by a *reward function*. Typically, RL agents interact with the defined environment in discrete time steps; the process can be summarised as follows: (i) the RL model receives the environment state information, which (ii) generates an action that (iii) produces a new state value and a reward signal which (iv) are used to produce a new action. This process is repeated iteratively. However, an important feature of RL is that it should consider not only immediate rewards, but also rewards received in previous iterations until certain time. This mid- or long-term planning is an essential feature for problems that require complex solutions rather than just the step-by-step execution. To address this and other implementation problems there are different algorithms for implementing RL models. Perhaps the most widespread is the Q-learning algorithm [WD92], although there are others also such as the Montecarlo, SARSA or Deep-Q network, among others [Sze10]. RL systems have demonstrated good performance in certain applications were actions to perform are well defined, such as in robotics [PDS13] [SA94] or playing computer games with even better performance than humans [MKS+15].

As it has been shown, RL is oriented to model the actions of a system on a defined environment. Hexa-X M&O services could take advantage of these techniques translating those actions into some of the basic orchestration actions defined in Section 7.1 e.g., NF instantiation actions, scaling actions or upgrading/downgrading actions, among others. A practical example might be the application of scaling actions on certain edge NFs based on measuring certain end-user behavioural patterns. These behaviour patterns would define the state of the system, which could be read through certain metrics. Reward signals may be derived from QoS metrics, while actions could be the scale-in/out orchestration actions on certain NFs.

Compared to supervised and unsupervised algorithms, RL systems do not need an external human supervisor. As described, RL models are designed to learn how to achieve well-defined goals by means of a well-defined set of actions in a continuous loop, and the learning is based on rewards obtained in continuous trial and error iterations, within a well-defined environment. I.e., once the software for the RL system has been designed (considering the metrics that need to be processed, the actions to be performed and the interfaces to communicate with the environment), it can be deployed on its working environment to start the learning process by means of repeated interactions (see Figure 7-9). Therefore, in RL there is not a clear separation between *training* and *production* stages, as it happened with supervised or unsupervised models.

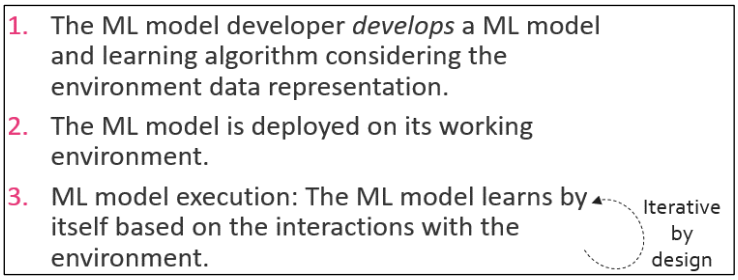

Figure 7-9. Steps to deploy a RL system.

This is probably one of the major challenges about integrating this kind of algorithms in M&O systems, because it must be assumed that the initial performance won't be good, since the system still needs to learn, and this learning happens directly on the environment on which the system should work. A typical approach used to mitigate this problem is to start the learning process in a simulated environment, and to migrate the system to a more realistic environment once a good performance level is reached. However, it has to be always considered that simulated and staging

environments are not going to be identical in terms of data and behaviour, so the overall result won't be optimal until the system starts working on the production environment.

Regarding data sharing, RL models must address the same type of challenges that have been mentioned for supervised and unsupervised models if they are first deployed on simulated environments in the Design Layer (those simulated environments should also represent data in a form as close as possible to how it appears in the production environment). Nonetheless, for some specific cases RL models may be directly deployed on the production environment without performing a previous learning stage. In those cases, RL models can evade the problem of exporting data towards the Design Layer.

Federated Learning

Federated Learning (FL) [MMR+17], also known as *Collaborative Learning*, is an AI/ML technique that uses local data samples to train ML algorithms on multiple distributed servers without extracting that local data from them. This approach contrasts with regular ML techniques (like those described before) where data sets are typically available on a specific common server. FL works on multiple local datasets without explicitly exchanging data samples; what is interchanged are just certain model parameters, such as the weights of neural networks, to generate a global model shared by all participant nodes. FL can rely on other different ML algorithms (e.g., supervised, unsupervised, or others), so FL cannot be considered a main learning paradigm by itself; the main difference is regarding the implementation, i.e., in this case ML algorithms are executed in a distributed way. The main advantage from this is that FL enables multiple distributed stakeholders to build ML models without sharing data, which allows to address key issues in telco-grade environments, such as data security, data privacy, data anonymisation, or accessing to heterogeneous data[4].

FL systems are generally categorised as centralised or fully-decentralised FL. In "centralised FL" a central Federation Manager (FM) coordinates the learning process by aggregating the parameters computed by different parties and sharing an updated model. Conversely, "fully-decentralised" FL does not require the presence of a FM and information is shared in a P2P fashion [LLZ+21]. Most FL implementations assume a centralised communication topology, the content of this section will be focused on this approach.

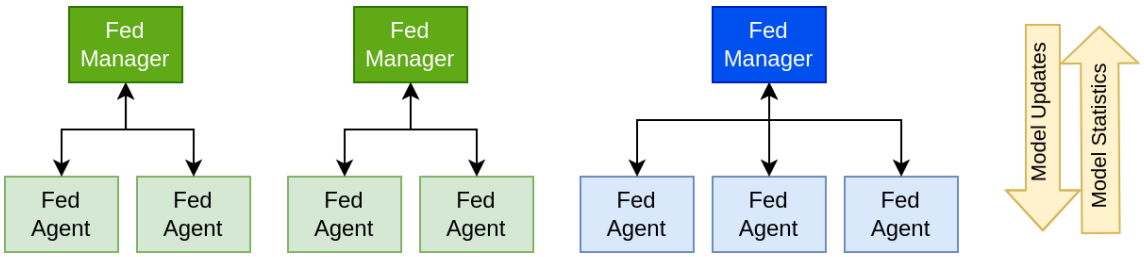

Figure 7-10. Exemplary FL scenario with multiple AI-based processes.

[4] Though FL can be seen as a form of DL, these two terms are typically used with different meanings. DL targets at training a single model using computing resources from multiple servers, being the common assumption that local datasets are independent, evenly distributed, and with roughly the same size. The aim is to parallelise the problem by relying on the distributed computing power. In contrast, FL typically assumes datasets are heterogeneous and different in size. Here the goal is addressing that heterogeneity without exchanging sensitive data, and also, assuming that participating nodes can be error-prone, as they can rely on less capable communication media (e.g., domestic networks) or battery-powered devices (such as some extreme-edge devices) [KMR15] [KMB+19]. On the contrary, DL typically relies on powerful computing nodes in regular datacentres. This document focuses on FL, since it is considered to be better aligned with the project scope (e.g., due the extreme-edge integration). However, some aspects discussed here regarding the M&O of these models could be applied to DL models as well.

Figure 7-10 shows an exemplary scenario wherein multiple Federated Agents (FAs) interact with the FM following a FL pattern. FAs can use different AI models (e.g., green and blue in the figure) among those made available in the federation platform. moreover, FAs using the same model can be associated to different federation groups [BEG+19] and, consequently, be coordinated by different FMs, to optimise model parameters such as accuracy and explainability, or use communication and computation resources more efficiently.

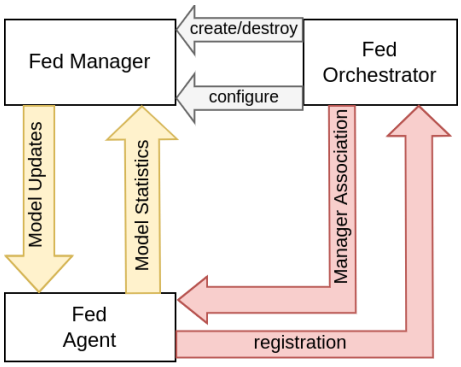

Figure 7-11. High-level view FL message sequence.

From a general perspective, each FA can be supporting the operations of either an orchestration function or a user application. In the first case the orchestration function would perform the basic orchestration actions (those in Section 7.1) on FAs and FMs. The second case refers the integration into the user services provided by the 6G system, which could be done following an AIaaS approach, e.g., by deploying FAs as services running on UEs. Examples of these services are detailed within the scope of Hexa-X WP4 D4.2 [HX22-D42].

The FL Orchestrator will be required to coordinate manager-agent interactions, and to manage the lifecycle of FMs, as shown in Figure 7-11. This FL Orchestrator would be integrated as part of the Managements Functions block in the Structural View. On the other hand, Federation Managers would be part of the AI/ML Functions block, while FAs would be those AI functions scattered across the architecture (Figure 6-1).

The instantiation of a FL-based service would work as follows:

1. A ML model developer *develops* an AI model and loads it on the federation platform.
2. An application developer *develops* a FA.
3. A FA Is deployed on its working environment.
4. A FA will register itself to the FO, specifying the requested AI model.
5. The FO searches for an available FM for the given AI model.
6. If a FM is not already available, or if the performance is suboptimal, a new FM is instantiated.
7. The FO configures the FM and performs a FM-FA association.
8. Loop:
    a. The FM transmits the (updated) model to the FA.
    b. The FA provides model statistics to the FM.
    c. The FM aggregates the statistics and updates the model.

As seen, steps a-b-c are repeated in a loop, which is executed until either the FA terminates, the FM is reconfigured/destroyed, or a new FM-FA is selected by the FO.

As it can be appreciated, FL models can approach the problem regarding the sharing of training data with external parties. However, from a development/deployment perspective, FL can be considered quite similar to Reinforcement Learning models described above, since FL systems are also designed to be trained right on the production environment. This makes it necessary to take the same consideration as for the RL models, i.e., FL models can also be error-prone (since the learning stage is still executed using real production data), although as for RL, this could be mitigated to some extent by performing initial pre-training stages (for example, avoiding the

system to trigger real orchestration actions until the learning process has not reached certain maturity degree).

In any case, although FL systems can avoid sending sensitive data to the software suppliers' scope, they still need to be in the loop: software suppliers still need to design the FL system (e.g., they will need to decide on the neural networks type and topologies, the federated/central nodes architecture, and how these elements get and process data and communicate each other). MNO operational teams should be also involved during the training process, at least to validate the models provided by the software suppliers and decide when they can go to production, or to ask software suppliers to intervene again to redesign the FL topology (or whatever else). This dependence from the software suppliers could be however minimised by relying on certain general purpose FL frameworks that could be used directly from the MNO operational teams with little support from the software supplier side.

AI-based control-loops

As explained in Section 7.2.3.3, closed-loop automation is a widespread model in control engineering when it comes to process automation. These control loops can be also enriched using AI/ML techniques. An evident example is the RL algorithms introduced above. Although closed-loop architectures do not specify any implementation, it is expected that some of their functionality will be mainly fulfilled by AI/ML algorithms [BT20].

Supervised and unsupervised learning models are also suitable for implementing closed-loop control, although the approach is not directly embodied in the standard learning algorithms for these models. In fact, the first way to include neural networks in closed-loop control was by using multilayer perceptrons, one of the earliest ML topologies [LLZ10] [MKK12] [CPL13]. However, after training, both supervised and unsupervised methods are usually deterministic, i.e., the same input always produces the same output. While they can learn highly nonlinear mappings from input to output and represent any possible function, their insensitivity to temporal structures makes them not directly well suited for time-varying problems.

However, as shown in Figure 7-6 and Figure 7-7 supervised and unsupervised algorithms can also be suitable for implementing automated control loops if the iterative ML model development process indicated by the bottom-to-up arrows is considered. As it is depicted in these figures, the initial steps involving the model training are typically performed in an offline manner, i.e., the model developer performs this work "before" the models are deployed on the production environment. Nevertheless, this process could be automated in some cases, e.g., it could be possible to re-train new instances of already deployed models with updated data sets, and automatically deploy the new re-trained instances through an automatic process able to verify that the performance of the new re-trained instances is better than the instances already in production. This process would be iterative, implementing an automatic control loop by itself.

In order to implement these mechanisms, it would be necessary to implement specific automation software which, in fact, would be a complement to the AI/ML software itself. This complementary automation software would be integrated in specific CI/CD pipelines in the Design Layer. These processes could be focused in improving the performance of already deployed AI/ML models and also in automatically deploying specific AI/ML models to meet particular time or demand requirements. E.g., it could be possible to have in the Design Layer repositories certain models already pre-trained to work on different timeslots (e.g., day/night timeslots). These models could be automatically deployed at the corresponding time or to cover specific service demands.

Another approach for automating control loops using AI/ML techniques are the so-called *adaptive* closed-loop control techniques [SB11], consisting of implementing closed control loops in which the controller can modify their own control strategies to adapt to new processes behaviours. E.g., applied to PID controllers, adaptive controllers can continuously monitor the loop's performance and update the PID tuning parameters to improve the performance on changing situations. In this case AI/ML systems can be used to learn on the changing situations and provide the loop parameters update. This way, by constantly updating the closed-loop parameters, the system can

self-adapt to unexpected or time-varying process behaviours. Using on-line data-driven learning by applying modern ML algorithms, instead of attempting to create a rigid a-priori model with all possible disturbances and system dynamic changes. This could benefit zero-touch approaches, since It allows the controllers to consistently improve their own performance without the assistance of human beings for both: controller parametrisation and operation. Using this approach ML can be used in a closed-loop setting in a very intuitive way. It can also be integrated into already in-place controllers with minimal effort and only be provided with the measurements that were already used for the in-place controllers. This approach has the advantage of being applicable with only minimal changes to the set up. It can therefore be used in almost every production system where the special capabilities of machine learning promise a benefit [Gun18].

7.2.4.2.1 Challenges regarding AI-driven orchestration

Although AI has great potential for network management and orchestration beyond 5G/6G, many challenges remain. In the following subsections it is summarised what have been considered the most relevant.

The learning stage

As explained in Section 7.2.4.2, AI/ML models need a learning stage. As described before (Sec. 7.2.4.2) the way this learning stage is introduced in the regular MNO workflows is a challenge by itself. Some approaches, such as supervised and unsupervised learning, can be treated as a regular software development process, where the AI model training stage replaces in some way the traditional software programming processes. However, this also comes with the handicap of providing data to external third parties (model trainers), with the possible associated privacy and security issues. As presented in the previous sections this can be addressed in different ways: Federated Learning (FL), or Distributed Learning (DL), represents an approach that has been specifically designed to address this problem; however, to focus only on this learning model can perhaps limit the kind of problems that could be addressed using AI/ML techniques. The same happens with the Reinforcement Learning approach. To provide a general solution including other learning paradigms the focus should also be on how to implement the necessary interfaces and workflows involving the Design Layer. These workflows would rely on both: continuous monitoring workflows and batch data interchange.

Reinforcement learning approaches represent also a challenge regarding the errors that would inevitably occur during the learning stage, since this learning stage would happen with the RL system directly integrated on the MNO production environment. This obviously could negatively impact on the service QoS/QoE, which makes this approach non suitable for all services. As previously explained, this could be mitigated by using simulated and staging environments to pre-train the models and reduce the impact of possible errors once in production.

Federated learning represents a similar challenge, since the learning in these systems also happens directly on the production environment. However, if the implementation is not based on RL (i.e., not a reinforcement signal from the production system is needed for adjusting the learning parameters) an additional mitigation measure could be to disable the FL system to perform actual M&O actions until the learning process is mature enough.

Another challenge common to all learning paradigms is the speed of training. 6G networks will be very heterogeneous and can vary enormously in a short period of time. To cope with this problem, certain models probably need to be trained continuously to avoid losing accuracy. This challenge could be addressed by using the closed-loop approaches explained above.

Computational Complexity

Artificial intelligence models (especially supervised and unsupervised models) often require long processing times for the training stage. Also, once trained, if the model is complex, it may require special hardware resources if agile real-time responses are required. This problem is usually solved using specific hardware resources (e.g., GPUs or FPGAs), which helps to reduce training and execution times (although increasing infrastructure costs). However, it is important to

consider that for supervised and unsupervised approaches the training would be typically done at the Design Layer, i.e., at the software development scope, so the main concern at the MNO side would be only regarding execution times.

Data availability

Training AI/ML systems requires data which needs to be collected, stored, and dispatched where needed, all in a timely fashion. This probably won't be a big problem for reinforcement or federated learning models, since they only work with the data that is already available in the actual environment on which they are deployed. But supervised and unsupervised approaches (or other approaches similar to them) can need huge amount of data from the different network layers, or even from external data sources (e.g., non-public networks, 3rd party infrastructure nodes). Since 5G networks are very new and 6G networks do not even exist yet, the lack of adequate data sets for addressing certain type of problems is also a challenge by itself [Sel19]. One possible method to address the lack of dataset is to use Generative Adversarial Networks (GAN) systems [GPM+14]. GAN are already successfully used in fields as audio or video, where they are used to perform image or video synthesis. Based on a small dataset (a 6G dataset in our case), a GAN can produce a larger dataset made of plausible data. Typically, a GAN is composed of two elements: a generator and a discriminator. The initial dataset is used to train the discriminator and enable it to differentiate data that could be part of the dataset from other data. The generator feeds generated data to the discriminator, which in turns indicates if the data is plausible or not. The generator learns accordingly and keeps generating more suitable data. This would allow, for example, to produce plausible 5G traffic [DMS+22] and, in the future, also 6G traffic.

Multi-stakeholder environments

As mentioned earlier, 6G networks, like 5G networks, will be very heterogeneous, and multiple parts of these networks may not belong to the same stakeholder, which has an impact on the AI algorithms used for orchestration. Firstly, some of the stakeholders may be malicious and send false data to the orchestration system, either to gain advantage or simply to disrupt the network. Therefore, algorithms are needed that are resistant to such adverse behaviour. Secondly, different stakeholders may be reluctant to provide their complete data set to the E2E orchestration, as some or all of this data may be sensitive. As E2E management still needs to be performed, AI/ML algorithms need to be adapted to provide accurate results without having access to the raw data. Regarding this topic, a federated learning system (see Section 7.2.4.2), possibly used with homomorphic encryption [AAU+18], is an interesting research direction.

Explainability

Data structures generated by AI/ML models come, typically, in the form of collections of huge rational number's matrices. Unlike traditional software programs, this makes it difficult for humans to understand the inherent logic in these models, which are commonly seen as sorts of black boxes in which the logic of its operation is not easily explainable.

This is a problem because, in case of incidence, it is important to know why the control system took certain decisions. It is important to remark that M&O actions can have serious consequences on the QoS/QoE provided to the customer, so improper M&O actions may originate serious economic and reputational cost to the operator, which obviously need to be properly explained. But beyond this, explainability is also needed as part of the regular network operations activities, although only for providing to the operational teams the necessary information to understand the operational processes and see how they could be improved. However, explainability is of paramount importance for operators to trust the AI and satisfy the customers. In fact, explainability is one of the key features regarding AI identified by the ALTAI (Assessment List on Trustworthy Artificial Intelligence) EU initiative [Alt20].

This challenge can be addressed using the so-called eXplainable AI (XAI) paradigm [DSB17], which can be used to find the underlying rules behind the already trained AI/ML algorithms. The objectives of XAI are to provide AI results that are trustworthy, transparent, informative, and confident [GSM21]. Trust and confidence are of paramount importance in general and for

explaining the AI algorithms behaviour, while transparency and informativeness are both required in order to assess the AI algorithms. The XAI model, therefore, should be explained clearly by considering related risks.

The applicability of XAI to telecommunications has been addressed by [ITM+21]. However, the XAI methods are still not in the scope of M&O systems. The main reason is generally lack of AI-driven Operation Support System (OSS)/Business Support System (BSS) solutions. Unfortunately, the M&O application of AI is complex, and the use of XAI is therefore not straightforward. The AI-driven OSS/BSS use a feedback control loop, that means the AI-assisted decisions impact the whole environment. In this case a typical feedback-based control system is used. Moreover, due to the rich functionalities of OSS/BSS there will be many OSS/BSS operations that are AI-driven, using a different set of algorithms, and not decoupled. Each mechanism may use a different AI/ML model.

In order to ease the use of XAI in this context, it is necessary to decompose the whole OSS/BSS system into smaller AI-driven subsystems to which XAI models can be reflected. It, thus, will lead to a kind of "local XAI" that will contribute to "global XAI". Such machine-to-machine explainability can be seen as a variation of XAI.

Failures

Even the most advanced state-of-the-art AI/ML systems are subject to errors. Some well-known examples are in the field of autonomous driving [CXC+19] or in the image's recognition field [EEF+17] [KP21]. The problem is that this seems to be a problem intrinsic to AI systems, so with no easy solution: patching a specific failure (e.g., an error in recognising a specific pattern) does not guarantee that the problem has been solved in a general way, or could even introduce new complications, because the wide variety of possible scenarios in real life is simply too complex for artificial intelligence to learn the best responses for all possible options. Even the best and more sophisticated neural network on earth (the human brain) can fail in classifying patterns (some optical illusions are an example of that), so it does not seem realistic to think that artificial developments based on simplified brain models could overcome this. Even more, it has been demonstrated that undecidability is in fact part of the algorithmic approach by itself [Tur36].

Regarding AI/ML techniques, the problem comes with the approach to manage complexity by delegating on algorithms able to learn by themselves, since the development of algorithms in the traditional way (i.e., relying on human programmers) overcomes human capacity when the problem cannot be explicitly described in an algorithmic way or the number or variables is too high, as it typically happens in real world complex systems.

But practical M&O systems able to deal with complexity and to make automated actions in the real world are needed, and those actions may have a relevant impact on the network performance and the deployed services. An important matter which shall be considered is that the services provider could incur in economic and reputational costs if failures may lead to degraded quality of services, which would negatively impact on the customers perception.

There could be different ways to face this problem. First of all, it is important to consider that AI/ML systems are not so different in this sense compared to regular non-AI software systems. Bug fixing is in fact a regular software maintenance practice, even with systems already in production. Of course, it is of paramount importance to minimise bugs and to have the appropriate workaround procedures always available in order to mitigate errors. The same rationale should apply also to AI/ML-based systems. However, another important recommendation is to have the human always in the loop. This is the "Human Agency and Oversight" recommendation from the ALTAI (Assessment List on Trustworthy Artificial Intelligence) EU initiative [Alt20], stating the need that final decisions should be always supervised by humans, while AI/ML systems should act just as recommenders. This should apply mainly to services where human life or security could be at risk, or when the estimated reputational or economic costs for the operator could be too high. In case of high risks human should always have the latest word [Jar18]. This measure can of

course be relaxed for those services not affecting the people safety, or those not putting at risk the reputation of the MNO.

7.2.4.3 Security-related processes

Regarding management and orchestration, security-related processes revolve around the Security M&O functional block in Figure 6-1. As detailed in Section 6.2.2.2, the core objective of this element is to apply the different steps detailed by security frameworks: Identify, Protect, Detect, Respond, and Recover. To do so, the Security M&O block must gather information of various kind from different sources and take appropriate actions toward various targets. This is summarised in Figure 7-12.

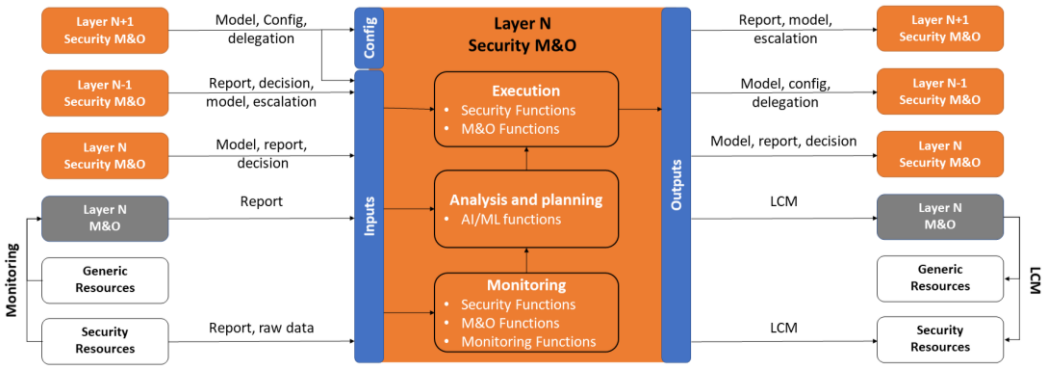

Figure 7-12. Security processes.

The security M&O would typically receive information primarily from its primary M&O block counterpart (the grey blocks in Figure 6-1). These would consist in reports summarising both the state of the resources the primary M&O is responsible of, and the LCM operation that the primary M&O will, has, or would apply to these resources. The security M&O may also receive monitoring information from its own security resources directly, gathering raw data if necessary. This monitoring may relate to the security resource itself, e.g., the statistics of a firewall, or to a generic resource that the security resource is monitoring. Additionally, the security M&O may receive inputs from its peers: higher layer (layer N+1), same layer (layer N) or lower layer (layer N-1) security M&Os. From N-1 and N layer security M&Os, the security M&O can receive reports containing the monitoring information collected by those M&Os, as well as the decisions they took through time.

Another specific type of information that can be exchanged between security M&Os, both in the same layer and intra-layer, are AI/ML models: while the exact implementation of a security M&O is open and will likely depend on the size and purpose of the M&O, it is strongly envisioned for 6G that many entities, including security M&O, will use AI/ML functions to fulfil their missions. In this context, FL can be used between M&O blocks to improve the efficiency of each individual M&O. This approach implies the exchange of models between M&O blocks. Based on all those inputs, the security M&O can decide of a set of actions to take. Additionally, the Layer N M&O may also receive from a Layer N-1 M&O a more direct request for action: an escalation. In this case, the layer N-1 security M&O identified an action to perform its own inputs, but is unable to perform it itself due to, for example, a lack of privilege. Similarly, the layer N M&O block can receive an action delegation from the layer N+1 security M&O which, thanks to its broader monitoring scope or/and its relationship with higher layers, identified an action to be taken within the layer N security M&O scope. Finally, the Layer N+1 security M&O may decide to modify the configuration of the Layer N M&O.

Based on the received inputs, the Layer N Security M&O has to decide if one or several actions have to be taken, to determine the nature of these actions, and to execute them. As evoked in the previous paragraph, the internal organisation of a security M&O can depend on its purpose, its size, the organisation that developed it, and other factors. In a single system, different security M&Os may have different internal organisation and, consequently, different internal processes.

This would typically be the case in multi-stakeholder environments explained in Section 7.2.3.4. The internal organisation in Figure 7-12 relies on the automated control-loop concept, which is described in greater detail in Section 7.2.3.3. In this specific case, all monitoring-related inputs are sent to the monitoring process. This process is built upon security, monitoring and M&O functions represented in Figure 6-1. Model-related inputs, as well as information about decisions, can be directly used at the analysis and planning processes, in order to update their internal AI/ML models. These processes rely primarily on AI/ML Functions to respectively detect potential security events and determine the actions to be taken. The execution process, just as any of the other processes, can be influenced directly by configuration modification.

The Layer N Security M&O has a set of possible types of actions to perform, and a set of targets. Just as it may receive action request from other layers, it may as well delegate/escalate actions to the Layer N-1 / N+1 security M&O block. Else, it may perform its own actions on the resources of its own layer. In this case, it may either directly perform configuration modifications over the security functions it is directly responsible of, or delegate M&O actions targeting generic functions to the primary M&O block. Finally, it may choose to share information (models, monitoring reports, reports of its own actions) to its Layer N security M&O peers.

8 Deployment View

In this section the Deployment View of the Hexa-X M&O architecture is provided. Although the section will provide some details about hardware components and how they can be grouped together to form servers clusters and datacentres, it should be understood that the intention of this section is not to provide an exhaustive description of these elements, which obviously goes beyond the scope of this deliverable, but simply to provide an overview, and above all, to explain how the structural building blocks described in Section 6 could be deployed in practice over a realistic environment, providing different considerations and possibilities to bring an architecture like the one described in the previous sections to a real deployment state.

8.1 Overview

Figure 8-1 shows the front-end/back-end division typically used in cloud-native architectures showing some of the network elements and stakeholders that are relevant for the Hexa-X M&O architectural design. As it can be appreciated, the MNO deploys the back-end network (the MNO cloud) that can be accessed from the front-end through different interfaces in the access network (If_1 to n). There are different stakeholders in the front-end: end users (using wired or radio access networks), software vendors, and other private or public clouds.

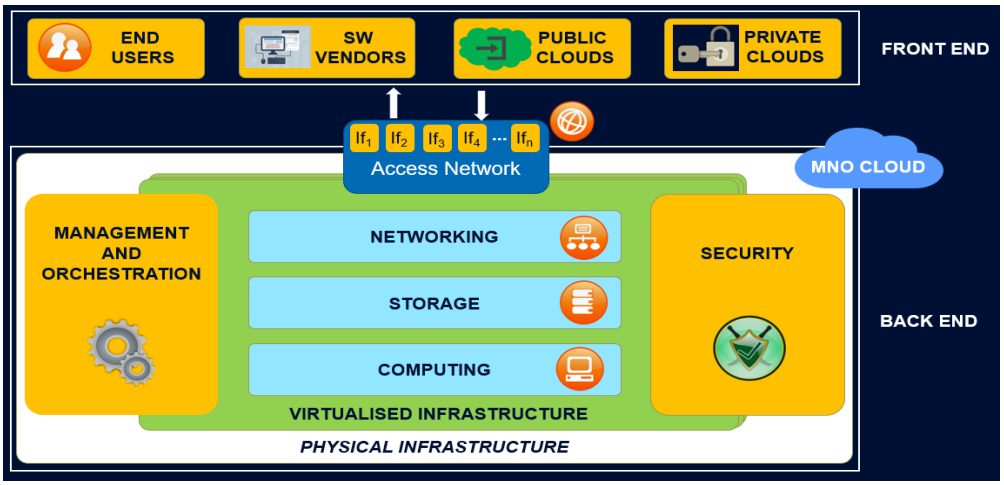

Figure 8-1. Front-end/back-end split.

As it can be appreciated, this figure follows what is the most general cloud-native approach, splitting the infrastructure into: back-and and front-end resources. In this scenario, the back-end is the infrastructure provided by the MNO. From the MNO perspective, End-Users, Software Vendors and other public/private networks are at the front end. Software Vendors "use" the MNO access network to access the MNO infrastructure (e.g., to implement CI/CD pipelines or to access monitoring data – i.e., the MNO work as "a platform" for them). Public and private clouds, on the other hand, extend the MNO infrastructure resources, offering their own infrastructure for the services deployment. End-Users, obviously, access the MNO cloud to access the offered services, but from the Hexa-X perspective, they also provide their extreme-edge resources to extend the available infrastructure beyond the edge network.

As it can be seen the backend provides the overall networking, storage, and computing resources, that are mostly deployed on a virtual infrastructure (green blocks). There is also a physical infrastructure, that is used to run VNFs or bare-metal functions, also known as PNFs, directly running on it. M&O, Security and Access Network functions also rely on that physical and virtualised infrastructure (note the partial overlapping).

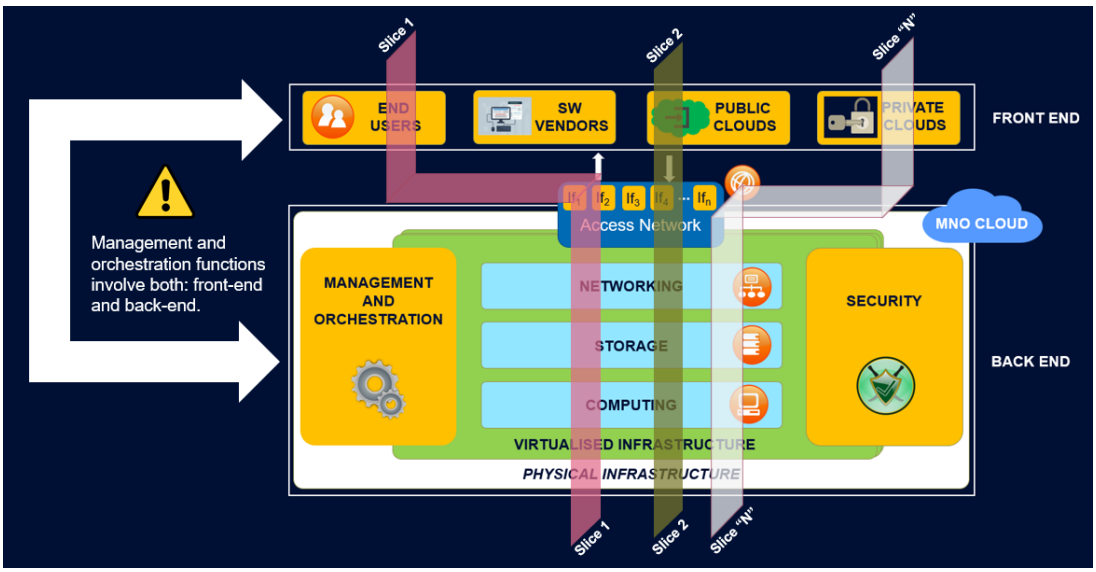

Figure 8-2. Network Slicing including front-end and back-end resources.

Network Slices would be deployed by grouping NFs on all the available infrastructure, including not only the MNO-Cloud itself, but certain available resources at the front-end, when necessary, e.g., end-user devices (the extreme-edge devices), or other public/private networks (see Figure 8-2). As described in previous sections, M&O functions would make this happen by means of the E2E seamless integration processes introduced in Section 7.2.1.

In the following subsections more details about how to deploy such architecture, starting with the main building blocks for deployment, are given.

8.2 Main building blocks for deployment

The Main Building Blocks for deployment would be mainly four:

- Computing Units (servers). These Computing Units would be used to allocate MOs, i.e., those objects outside the red-dashed line depicted in the Structural View (see Figure 6-1). These computing units typically are general-purpose servers providing the physical resources for computing (CPU and volatile memory basically) and running a virtualisation layer (e.g., a containers engine) on top of which containerised managed

objects (e.g., CNFs) are executed[5]. Sometimes they can provide hardware acceleration by means of specialised GPUs, ASICs or FPGAs.

- Management Units (or controllers). These Computing Units would be the ones used for executing M&O resources, i.e., those within the red-dashed line illustrated in the Structural View (see Figure 6-1). Management Units are specialised computing units for performing the Management Functions (e.g., instantiating NFs on different computing units and monitoring them).
- Storage units. Provide storage capacity by different means (e.g., SSD or HDD units) for the functions running on the Computing and the Management Units. They can be independent physical units (e.g., grouped in the form of high-capacity RAID units), but they can be also integrated as part of the Computing Units.
- Connectivity units (network switches). Provide network connectivity among the different units above, and also, other network elements (e.g., legacy PNFs, or third-party networks). Some of these Connectivity Units could be implemented by means of regular network switches, but they would preferably be implemented through SDN switches to ease network programmability.

Those blocks would be typically arranged in Servers Racks. Figure 8-3 represents a hypothetical deployment on just one Server Rack containing a dedicated controller (Management Unit), a switch (Connectivity Unit), and different Computing and Storage Units[6].

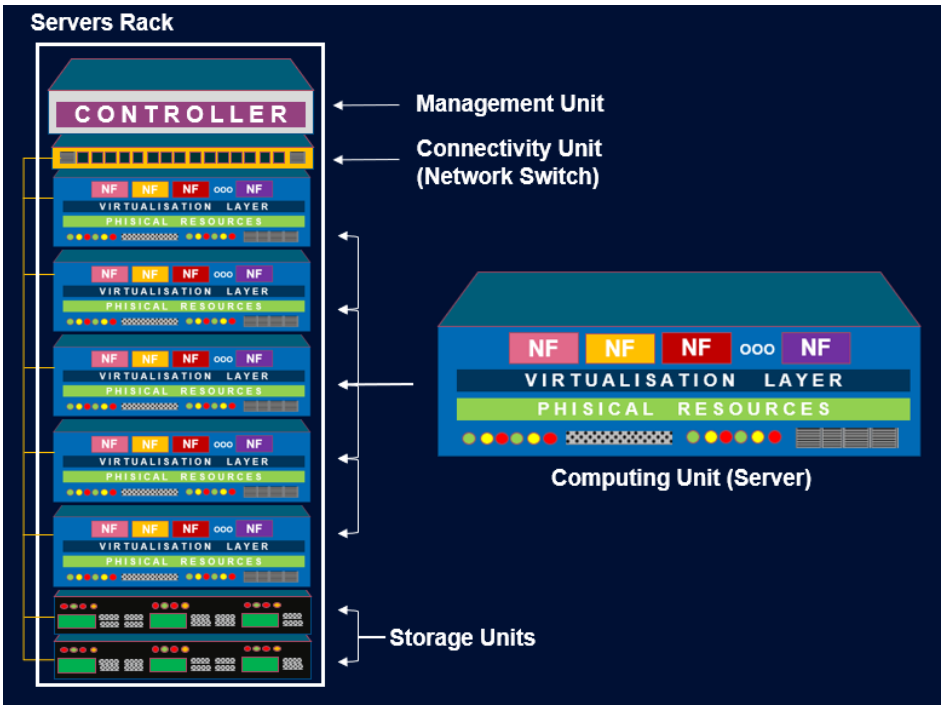

Figure 8-3. Main deployment building blocks.

[5] Although other implementation approaches could be used (e.g., VMs running on hypervisors), this section aims at lightweight microservice-based functions, as this is one of the main requirements for the M&O architectural design (see Section 5). However, this does not prevent backwards compatibility which certain services may require or even future technologies that could emerge in the coming years and that could offer similar functionality with better performance.

[6] Size and specific configuration of these racks could be different in practice. Some racks could have a different number of Computing Units (although a high degree of packaging will be sought to increase computing and storage capacity). Besides, although specific Management Units could be deployed per rack, this is not probably necessary in some cases, since the same Management Unit in one rack could be used to manage a big number or servers installed on different racks. In short, the configuration of each rack will be made according to the available resources and the specific needs that need to be covered.

Figure 8-4 shows a sequence diagram showcasing a hypothetical NS deployment on an environment like the one described in Figure 8-3 involving one controller, three different Computing Units (servers) and a NS consisting of four CNFs:

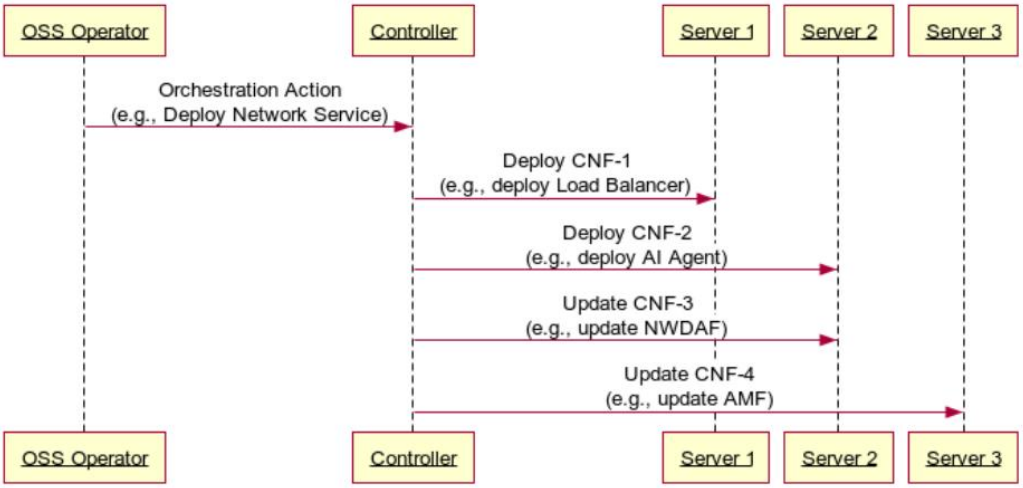

Figure 8-4. Deployment Example.

8.3 Grouping Racks

A single servers rack, like the one presented in Section 8.2 will not be enough to meet the potential demand. However, it can be used as a basic Building Block that can be multiplied as many times as necessary to build larger datacentres (see Figure 8-5).

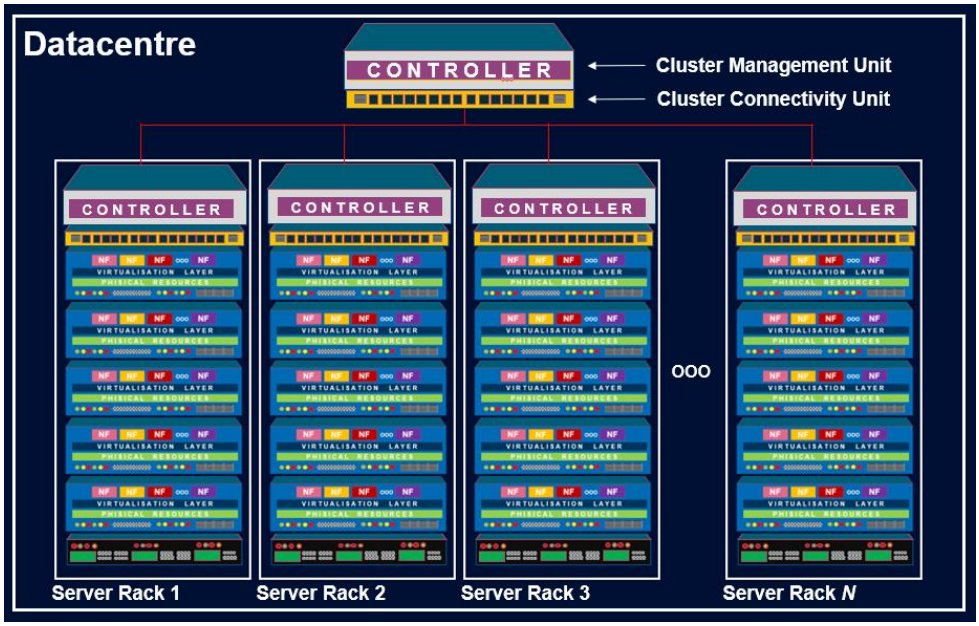

Figure 8-5. Racks Cluster.

Obviously, these datacentres would be accordingly dimensioned. As it is depicted in Figure 8-5, specific cluster Controller and Management Units have been included (top). Their function would be basically the same as those in each rack, although at datacentre-level. However, the Controller Unit allocated in the higher level could be implemented in different ways: (i) centralised, where a specialised datacentre-level controller similar to those in the racks acts on the rack controllers; (ii) federated, where the controller functionality is built-up upon the coordination among the existing controllers in the racks, i.e., it would be more a software function emerging from the coordination among the different rack controllers, and not just an additional physical building

block similar to those already in the racks [ML20][AP21]. The centralised approach can provide operational speed and simplicity, although the downside of it would be to have SPoF (Single-Points of Failure); therefore, appropriate fail-over and redundancy measures should be taken in this kind of scenario. Nonetheless, centralised scenarios still can be valid for reduced scale deployments. Federation, on the other hand, provides significant advantages, such as better availability to deploy NFs when losing servers, extensive scalability, easier migration of applications between servers, ability to spatially distribute servers on different domains (e.g., on the central cloud or the edge/extreme-edge), as well as infrastructure cost reduction through optimised balancing of on-premises and third-party servers utilisation (e.g., those from other public or private networks) to organize workloads. However, the federation approach also raises a number of security requirements, as detailed in [LBM20]: federation members' authentication, sharing of resources between them, resource discovery, resource access authorisation and integrity and privacy enforcement during all those operations. Additionally, within each single cloud the security mechanisms should have the capacity to work autonomously in case of a disconnection from the rest of the clouds [RLM18]. This implies that security mechanisms should be distributed, sharing security information between them but being able to function autonomously if needed.

8.4 Grouping Datacentres

Although datacentres help to improve computing capacity, they are probably still not enough to meet large-scale telco operators' requirements. However, this cloud-native deployment-based approach can be exploited just by repeating the same structures once and again, and by interconnecting them appropriately.

Figure 8-7 shows this approach illustrating what in Hexa-X is called a "Distribution", i.e., a grouping of datacentres. Those Distributions would group other datacentres that could be geographically distributed (to both, increase capacity and provide geographical redundancy). As it can be appreciated, Distributions also include Management and connectivity Units (controllers could be implemented in the same way as those of the individual datacentres, i.e., using centralised or federated approaches, as mentioned in Section 8.3).

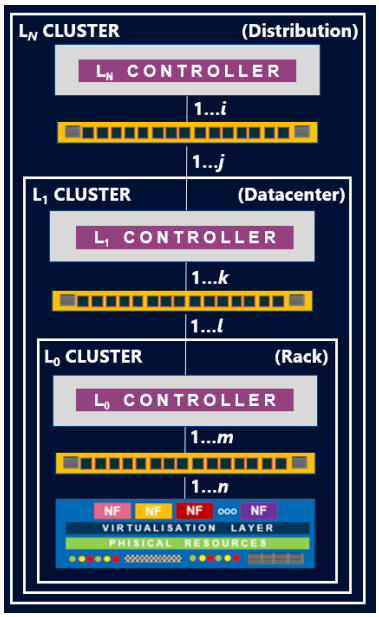

Figure 8-6. Compact Deployment View.

This structure could be repeated in a recursive way, issuing different levels of clusters. Figure 8-6 generalises this concept illustrating the possible cluster levels in a compact way: Level-0 would represent the racks level; Level-1 would represent datacentres, and Level-2 would represent

Distributions. This structure could be recursively repeated to get additional levels (i.e., Level-3, Level-4, etc.) until the required capacity and geographical coverage were reached. Each of these clusters would be accordingly dimensioned: some could be high-capacity datacentres comprising the core-network, while others could be small sparse cabinets attached to the radio access nodes conforming the edge network.

This vision implements different M&O levels. The different cluster levels (Level-1, 2, 3…) would be deployed following different criteria, e.g., they could belong to different scopes (e.g., edge or CNs) or geographical areas. By using this approach, the different domains (RAN, CN, TN, cloud, edge, extreme-edge, ...) would be reached. This approach could be escalated from very local deployments up to wide global deployments. Small Level-0 clusters would only require regular management capabilities focused on small racks, while higher-level distributions would require higher orchestration capabilities involving different network and administrative domains[7].

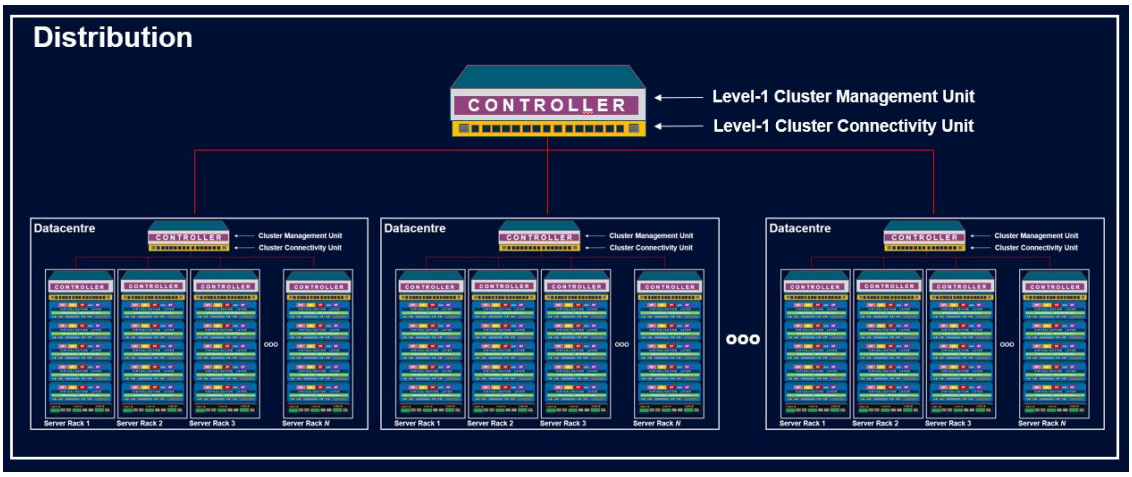

Figure 8-7. Level-1 Clusters.

In the specific case of edge/extreme-edge cloud facilities, some points of presence may have very limited resources, being constituted of small servers, single computers, or even small-form computers such as Raspberry PIs, or similar. The physical security of the machines will be lower than the ones located in central datacentres, and due to their limited resources, they may not enforce as much security control as legacy clouds [RLM18]. However, it should be noted that the increase of capacity allows for some level of autonomy even in small devices [GAD+15]. The level of security that each cloud is able to provide should be clearly advertised, so that services with high security requirements will not be placed in low security infrastructures. In addition, such advertisement could allow clouds to adapt their security measures when exchanging data with each other, to avoid being infected through an unsecure member of the cloud group.

[7] A very important aspect for implementing this vision are the Connectivity Units and the connection infrastructure associated to them. Relevant features of the Connectivity Units themselves would be number of ports, bandwidth per port, ports isolation (they should efficiently work in parallel), and power consumption. Custom ASIC designs or using specific processors for the NIC (Network Interface Card) could help to improve their overall performance and to reduce the overall network latency (e.g., by delegating on them repetitive network processing tasks instead overloading computing units with that). Besides, beyond the Connectivity Units, connectivity itself is also obviously quite important. Radio- and high-bandwidth fibre optic-based connectivity should be available. Clusters would be housed in different geographically distributed buildings to ensure redundancy and low latency connections, so high-performance intra- and inter-building routers should be available to enable connectivity. In any case, or large-scale deployments (nation-wide or higher) an appropriate deployment topology should be carefully defined, e.g., considering specific PoPs (Points of Presence) and their distribution in different regions. Each region should provide high-capacity and redundant transit gateways to enable interconnectivity with other regions or with other external public or private networks.

In order to achieve a framework that efficiently manages the deployment and LCM processes of several Network Slices across a wide variety of distributed resources, such as the Continuum M&O concept described in Section 7.2.1.1, it is important to consider the impact of the selected Management Unit's implementation approach (centralised or distributed) on the system's scalability and reliability. Centralised approaches have a big advantage regarding deployment and maintenance simplicity but, as demonstrated with SDN controllers and k8s masters [BSM18] [JHM21], this approach comes with several disadvantages related to the difficulty of adapting the dedicated control plane resources to the variability of the incoming petitions due to small or large network changes. Thereupon, centralised Managing Functions, at datacentre level (Ln-x) or at Distribution levels (Ln), will carry linked SPoF (Single-Point of Failure) issues and potential scalability and reliability concerns which will directly impact those capabilities.

On the other hand, distributed approaches where the Managing Functions or controllers are fully distributed (physically and logically independent) or partially distributed (physically distributed but logically centralised) reduce the scalability and reliability impact at the cost of increasing the complexity of these resources [BSM18]. It is important to remark, at this point, that the federation of Managing Functions will be performed per Cluster Level (see Figure 8-6) and, consequently, a clear functional division between lower-tier and higher-tier Managing Functions must be accomplished to reach the desired distributed environment. Firstly, this functional division must contemplate the edge-cloud Continuum M&O Computing Functions heterogeneity i.e., a wide range of general-purpose servers, end-user devices and other computing resources are expected in 6G networks; and even the straggling or failing of Computing Functions[8], the possibility of these computing resources losing connectivity, failing or joining some cluster level will be hard-to-deal reality in 6G networks (as explained in Section 8.1. network slices may use certain available resources from the front-end e.g., end-user devices, to expand/complete their capabilities). Secondly, Hexa-X Managing Functions of the same cluster Level should be able to establish proper communication channels between each of them in order to share their state and receive/forward operations statements to the respective Computing Functions and achieve a proper per-Cluster-level view[9]. Finally, proper inter-cluster-level communication should be accomplished in order to be able to respond and adapt to network changes and avoid overloaded Managing Functions at any cluster-level.

When addressing the above considerations, it is important to consider the consequences of the CAP theorem [MJG19] applied to the management of such a system. The CAP theorem states that in a distributed system, it is impossible to fulfil Consistency, Availability, and Partition tolerance simultaneously, i.e., only two among them (CA, CP, or AP) can be ensured simultaneously in any distributed system. This key result implies that the M&O of the disaggregated network and the data centres conforming it need to carefully consider what of the three characteristics (CA, CP or AP) needs to be fulfilled. In a telco scenario it is more probable to be able to operate in a portioned network, where the partitions are designed on purpose, for example based on RAN technology, and then design the M&O such that the system is consistent and available.

In summary, the overall scalability and reliability of the Hexa-X framework deployment will have strong dependencies on the final implementation approach and on the technologies used to stablish communication between the different Functions inherent to the Distribution.

[8] See straggler problems in wired/mobile networks [ZKJ+08] or in federated computing frameworks [SNH+21].

[9] SDN will be part of the Managing Functions and, therefore, although new technologies may arise in the upcoming years that could substitute SDN in 6G networks, it is important to remark that there is a lack of consensus/standards around East-West (SDN Controller to SDN Controller) interfaces [MB16]. This may have an impact on the implementation of Hexa-X Managing Functions.

8.5 Integration of the extreme-edge

Extreme-edge is not what is normally identified as a datacentre. However, as already explained in previous sections, it is a valuable set of resources located very close to the end-users on which M&O actions can be performed.

In general, the deployment and maintenance of the extreme-edge is not a direct responsibility of MNOs (although they could also deploy some resources in this domain to implement certain own services). The extreme-edge is potentially diverse and heterogeneous, depending on the type of end-user devices and the environment in which they are deployed. Certain environments may be envisioned, where a good control degree due the management of a social entity (e.g., factories, airports, shopping centres, sport stadiums, etc) is achieved, but other extreme-edge environments where devices belong to small entities or even individuals (e.g., SME devices, connected home appliances, or personal devices) may exist.

Another challenge has to do with the extreme-edge devices themselves: perhaps not all of them are suitable to host the necessary software artifacts to enable them as "orchestrable" resources due to their limited computing or storage resources, or because of any other technical constraint. Probably, small personal, or IoT devices will be unsuitable, but there are many other devices with higher capacities that could be used for storage and for running certain lightweight NFs e.g., on-boarded telecommunication units in vehicle fleets, smart TVs, industrial devices, drones, personal computers, gaming consoles, etc.).

From the technical perspective the integration of this diverse infrastructure into the MNO scope is a major challenge. As introduced in Section 7.2.1.1 (Device-edge-cloud continuum management) this could be initially accomplished by deploying on the extreme-edge devices microservice-based software components running on lightweight containers orchestration platforms (e.g., KubeEdge [KuE22] or lightweight Kubernetes (K3S) [K3s22]). Also, by enabling the necessary APIs to provide secure and effective communication with the devices in this domain, and by implementing enhanced infrastructure managers at the MNO scope able to manage the asynchronous nature of the extreme-edge devices and keep resource catalogues properly updated in near real-time.

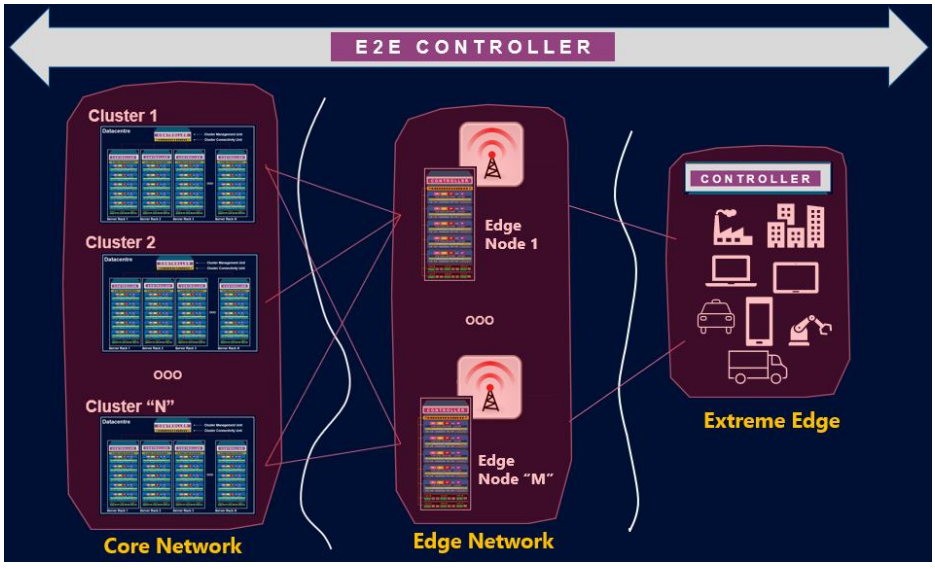

Figure 8-8. E2E Orchestration including the extreme-edge.

Figure 8-8 provides a hypothetical representation illustrating how E2E orchestration could be performed including regional cloud, edge and extreme-edge resources. As it can be seen, CN and edge networks could be implemented using different racks and clusters, as those described in the previous sections. Extreme-edge resources are treated as an additional set of infrastructure

resources that can be also associated to a controller (or a set of them) for providing E2E continuum M&O functions. Two options are envisaged to provide this functionality:

- To rely on federated lightweight controllers directly installed on certain extreme-edge devices (based on those lightweight container orchestrators previously mentioned). This approach should consider the potential high volatility of the extreme-edge resources (the node where the lightweight orchestrator is running could be unexpectedly disconnected), so that the necessary redundancy mechanisms are enabled.
- To rely on a set of "trusted" Management Units at the MNO scope, e.g., by means of specific controllers installed on some edge nodes that would be specialised in certain sets of extreme-edge resources. The benefit is that those Management Units would be in a more controlled environment (i.e., not so volatile as the extreme-edge environment).

What it has been described in this section is mainly related to technical aspects. However, beyond the technical challenge, there are also administrative challenges that should not be neglected: e.g., for personal users, it would be necessary to obtain their consent to install and execute on their devices the necessary resources to facilitate orchestration. Also, for corporations or SMEs it will be also necessary to sign specific agreements for this purpose and other purposes (e.g., related to intellectual property, personal data treatment, or confidentiality, among others).

9 Alignment with Standards

As shown in previous sections, the Hexa-X M&O architectural design does not explicitly align with a specific standard. This is intentional: the target here is to provide an agnostic design not explicitly aligned with a specific standard (or a few of them), but abstract enough to make possible implementations aligned with a variety of relevant SotA standards (or even possible future standards).

The main reasons to take this approach are: (i) the high diversity of SDOs and other standards approaching different architectural designs [HX21-D61], and (ii) the speed at which new standards can appear (or become obsolete) in the temporal scope of this project. Considering this, and since Hexa-X aims to lay the foundations for the development of the new 6G technology as a long-term objective, it is considered a risk to strongly align the M&O architectural design with a specific standard (or a few of them), i.e., if that standard were not widely adopted, the Hexa-X M&O design could become prematurely obsolete.

However, it is also considered of paramount importance that this abstract architecture has the capability to be potentially aligned with different SotA standards. Therefore, the following subsections include various alignment exercises with relevant standards from SDOs and other forums.

9.1 SDOs standards

9.1.1 ETSI

9.1.1.1 ETSI NFV MANO

The ETSI NFV MANO standard [Man001] defines a telco-cloud stack formed by the well-known abstractions NFVO (NFV Orchestrator), VNF Manager (VNFM), Virtual Infrastructure Manager (VIM) and Container Infrastructure Service Manager (CISM), with VIM allowing for IaaS (VNF hosting) and CISM allowing for Container as a Service (CaaS) CNF hosting [Ifa029]. Outside the NFV MANO stack itself the standard also defines the NFVI (NFV Infrastructure), representing both: virtualised and physical resources.

Based on the scope and responsibilities, these abstractions could be easily mapped to the Hexa-X M&O architectural design (Figure 6-1): the NFV MANO stack abstractions (i.e., NFVO, VNFM and VIM/CISM) would be part of the Management Functions block in the Network Layer, while the NFVI could be identified with the Infrastructure Layer itself, as defined in the Hexa-X Structural View. Consequently, being aligned with ETSI NFV MANO definitions, the Hexa-X architecture would also provide the possibility to make the Network Layer oblivious of any structure on a higher logical level than the NS, such as slices, which would be exclusively handled by the Service Layer M&O block.

This alignment means that the ETSI NFVI MANO abstractions can be "mapped into" the Hexa-X M&O architectural design, but it is important to remark that the Hexa-X design includes other abstractions that are not considered in the ETSI NFV MANO standard e.g., the AI/ML Functions, the Security Functions, or even more important, the relying on the SBMA model, which is out of scope for the ETSI NFV MANO standard. The term "alignment" referred in these paragraphs has a mean of "compatibility", in the sense that the Hexa-X M&O architecture could support the deployment of the ETSI NFV MANO abstractions if a potential stakeholder requires it.

However, beyond the NFV MANO stack itself, it is considered that there could be an important misalignment point between the ETSI NFV MANO standard and the Hexa-X M&O architectural design: the integration of the extreme-edge resources (see Section 7.2.1). Although the NFVI abstraction considers all infrastructure resources (virtual and physical), those resources at the extreme-edge have a particular relevant feature: they can unexpectedly change their availability state (they are error-prone, since they are not in strictly controlled datacentres; or even with no errors, they can be unexpectedly connected/disconnected by the end users). Thereupon, this would require that these resources are managed asynchronously, probably with an event-based protocol allowing to update in (near) real-time the information about the available devices and their current status. Regarding the ETSI NFV MANO standard, this would probably impact the VIM and CSIM abstractions, and also their interfaces, in order to quickly propagate the changes in the NFVI block up towards VNFM, NFVO and their associated infrastructure catalogues. These aspects are currently either out of scope, or not yet developed by ETSI.

9.1.1.2 ETSI MEC

The Hexa-X M&O architecture inherits some of the concepts of the Multi-Access Edge Computing (MEC) Framework and Reference Architecture [Mec003], in particular for the capability to deploy and orchestrate some of the service components and NFs on edge computing nodes in an NFV-enabled infrastructure. Another key characteristic provided by the MEC framework is the support for dynamic registration and discovery of MEC services provided and consumed by MEC application. A similar concept is adopted for the API Management Exposure in the Structural View (see Figure 6-1), enabling the discovery of APIs offered by various M&O functions of multiple domains towards potential API invokers. In this context, a brief discussion about the relationship between CAPIF (which directly inspires the API Management Exposure in Hexa-X) and MEC platform API-related functionalities is reported in [Mec031], which analyses the applicability of MEC to 5G networks.

Inter-MEC and MEC-to-Cloud interaction scenarios and use cases, as presented in [Mec035], are also relevant for the inter-domain orchestration procedures in Hexa-X, e.g., in terms of information exposure, federation of domains belonging to different stakeholders, etc. However, it should be noted that the ETSI MEC framework identifies specific functions and components that are not directly replicated in the Hexa-X M&O architecture; instead, the compatibility of the approaches facilitates the potential integration of MEC-enabled domains under the wider scope of the Hexa-X E2E M&O architecture, as a particular case of federated edge domain.

On the other hand, since the ETSI MEC standard relies on some of the abstractions defined for the ETSI NFV MANO standard (previous Section 9.1.1.1), the same considerations regarding that standard apply also here, especially those regarding the integration of the extreme-edge domain.

Also in relation to the fact that the term "alignment" would rather refer to "compatibility", in the same sense as mentioned in the previous section.

9.1.1.3 ETSI ZSM

This ETSI ISG defines an architectural framework [zsm-002] that allows zero-touch operation of networks and hosted services, leveraging programmability and automation capabilities. The design principles and components of this framework were reported in [HX21-D61].

According to the Structural View of the HEXA-X M&O architecture (see Section 6), one can notice that the "API management exposure" mimics the ZSM concept of cross-domain integration fabric, with the ability to make management capabilities available for external consumption, in a secure and auditable manner (controllable capability exposure). Likewise, a number of capabilities reported in HEXA-X leverages (or are aligned with) the on-going work in ETSI ISG, thereby open up opportunities for synergies and collaboration between them, in both sides. Table 9-1 reports on this relationship.

Table 9-1. ETSI ZSM work items relevant for HEXA-X work on M&O.

ZSM work item	Work item description and impact to HEXA-X
Closed-loop automation solutions (ZSM 009-2, specification)	On the one hand, the ZSM 009-2 specification describes Closed-Loop Automation (CLA) solutions of particular E2E service and network automation use cases, where one or more closed-loops are involved, following parent-child (hierarchical) or east-west (federation) relationships. On the other hand, ZSM 009-3 investigates advanced topics related to CLA such as learning and cognitive capabilities, ways to set and evaluate levels of autonomy, and operational confidence of the behavior of the closed-loops. The outcomes of both work items will inspire the WP6 work on layer specific control loops (e.g., service layer control loops, network layer control loops, infrastructure layer control loops, and DevOps control loops), clarifying the interactions across them (as per recommendations in ZSMM 009-2), and the impact of AI/ML functions in the lifecycle of the different closed-loops instances (as per conclusions in ZSM 009-3).
Closed-loop automation advances topics (ZSM 009-3, report)	
Intent-driven autonomous networks (ZSM 011, report)	This work item investigates the potential use of intents as key enabler for approaching full autonomy (zero-touch) in the management of networks and hosted services. Apart from providing guidelines on how to use intent-driven management interfaces between verticals and operators, it analyses existing domain-specific intent models defined in 3GPP SA5 and TMForum and provide solutions to define a root model out of them. The objective is to avoid hyper-fragmentation in standards, by outlining a parent a common yet extensible model based on which technology-tied models can be specified, depending on the specificities of the particular technology. The outcomes of ZSM 011 (close to be made publicly available) can be used as a basis to further enhance the intent-based service definitions and management solutions (see HEXA-X design and service layers) to be developed and reported in upcoming HEXA-X D6.3, ensuring their alignment with the work in relevant standards.
AI enablers (ZSM 012, specification)	This work item specifies AI-based capabilities providing support for the automation of management and orchestration functionalities. These capabilities are focused on areas related to data (including data collection and analytics), action, interoperation, governance and execution environment. It can easily noticed the impact and relationship of ZSM 012 with the baseline capabilities reported in the AI/ML functions box from

	HEXA-X architecture, specifically 'AI models', 'prediction' and 'data analytics. 'AI federation' is out of scope of ZSM 012.
CI/CD automation (ZSM 013, report)	This work studies use cases, potential requirements and possible solutions of automating continuous integration continuous development (CI/CD) of software related primarily to management services and/or functions and secondarily to managed services. The study looks at topics such as release models' compatibility, ensuring the compatible delivery of builds across vendors and operator teams, version control, and their handling in an operational environment [zsm-013]. The ideas captured in this work item are the core part of the HEXA-X design layer, and they can further inspire the DevOps /AIOps approaches and solutions to be worked out in this WP.
Security management (ZSM 014, specification)	This work item reports on a solution suite to support zero-touch security controls in an operator's management and orchestration stack. Looking at HEXA-X M&O architecture, it can be noticed that ZSM 014 outcomes can be used for the following purposes: i) providing intelligent security services on HEXA-X security functions, ii) defining automatic security governance for Management Functions, AI/ML functions and Monitoring Functions, within and across them; iii) enforcing adaptive trust relationship between HEXA-X architecture layers; iv) provide dynamic access control and auditability (traceability) features in the API management exposure fabric.

9.1.1.4 ETSI GANA

Generic Autonomic Network Architecture (GANA) reference model is an ETSI standard to enable the domain of autonomic (communication, networking, management) and cognition management for the self-management (networks and services) capabilities in a target architecture [GANA]. Figure 9-1 represents the GANA architecture, which contains the following main abstractions:

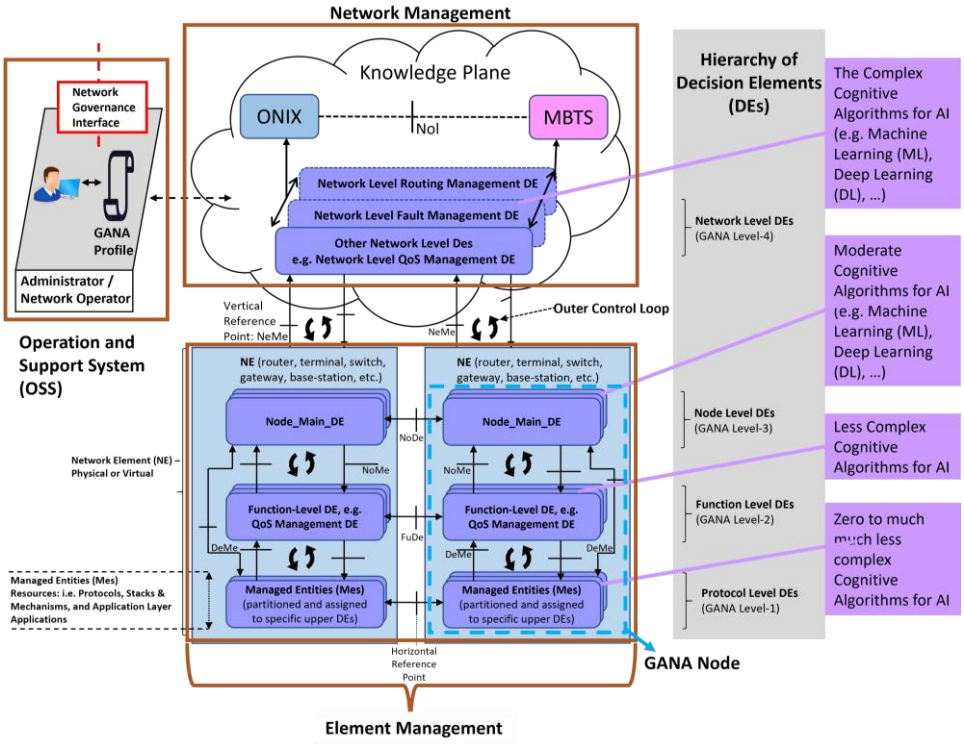

Figure 9-1. GANA reference model architecture [GANA].

1) Managed Entities (ME), which are managed resources (physical or virtualised) placed at the bottom of the management hierarchy. They represent protocols (e.g., OSPF, NetConf, TCP/IP, HTTP, OpenFlow, etc.), stacks or other types of resources.
2) Decision-making-Elements (DEs), to implement Autonomic Management and Control (AMC) by means of autonomic functions. DEs, also called Autonomic Functions (AFs), are the control-logic components responsible for autonomic management. They dynamically adapt network resources, parameters, and services. MEs can be managed by a DE and employed, orchestrated, configured and dynamically adapted to achieve some goals through parameter settings [AFI15]. As it can be seen, there is a hierarchy of DEs grouped into Levels. These DE Levels are associated to cognitive algorithms, working as hierarchical control loops. The overall idea is that Level 1 (the lowest level) reacts to events and incidents within a short timescale, while Level 4 (the highest one) operates in a network-wide scope, so the time to process all the information is typically longer. Below the description provided for each level:
   - Level 1: Also called *Protocol Level DE*. This level relates to any ME that may show intrinsic control loops (DE logic) and associated DEs. It handles the behaviour of protocol managed entities and resources (the TCP protocol has such mechanism natively).
   - Level 2: Also called *Function-Level DE*, manages entities of a Network Element (NE), including those that embed protocol level DE (Level 1) logics. It aggregates events and takes decisions based on events or policies. Relates to a DE for collective AMC of a group of protocols and mechanisms that are abstracted by a management/control or a networking function. GANA specifies six functions associated to this level: routing, forwarding, QoS, mobility, monitoring, and service/application.
   - Level 3: In this level the *Node-Main DE* orchestrates and manages *Function Level DEs*, but it can also manage MEs which are not managed by the Function Level DEs but required to be managed at node level. This Level 3 explicitly targets the following four DEs: Security management, resilience and survivability, auto configuration and auto discovery, and fault management.
   - Level 4: It covers network wide views and the management & control of lower levels e.g., node/device levels. This level is designed for DEs to operate in a logically centralised manner.
3) Network elements (NEs), which are a composition of DEs and MEs (see Figure 9-1). In practice NEs can be routers, switches or base stations, among others, either physical or virtualised. The *GANA node*, are NEs where DEs have been instantiated to automatically control and manage MEs. The *GANA Node* shall be governed by the GANA Knowledge Plane DEs.
4) The Knowledge Plane (KP), which enables advanced management and control intelligence at the Operation and Support System (OSS), and for the Element and Network Management levels, by interworking with them or enhancing and evolving the intelligence of these systems at these levels. For the KP the following abstractions are defined:
   - Network level DEs. These are DEs with a network wide scope. They are designed to operate the so-called *outer control loop*.
   - The ONIX (Overlay Network for Information eXchange) function. Used to enable auto-discovery of information or resources of an autonomic network via protocols (publish, subscribe, etc). DEs can make use of ONIX to discover information and entities (e.g., other DEs) in the network to enhance their decision-making capability.
   - The MTBS (Model-Based Translation Service) block. Which represents an intermediation layer between KP, DEs, and NEs, for translating technology and vendor specific raw data into a common data model to be used by network level DEs. This MBTS (Model-Based-Translation Service) function is basically intended to translate DE commands and NE responses to the proper data model and communication methods to ensure they can be properly decoded on each side.

An important feature of MANO is the distribution of control loops, in opposite to some centralised concepts, however interactions between DEs and their coordination is not addressed.

Alignment with the Hexa-X M&O architectural design

The main GANA architectural abstractions could be mapped to the Hexa-X M&O architectural design in the following way:

- Both, MEs and NEs, can be mapped to some of the Hexa-X MOs in the Infrastructure Layer, containing both: physical and virtualised infrastructure components.
- DEs, can be functionally mapped with the Managing Objects defined in Section 6.2, being part of the Management Functions block (see Figure 6-1), although according to the GANA principles, they would be distributed through the infrastructure, and not centralised. According to the Hexa-X approach, these DEs could be also supported by the other associated functions in the M&O scope (i.e., AI/ML, Monitoring and Security Functions). The *Node-Main DE* would be one of those functions from the Management Functions block.
- The Knowledge Plane could be also implemented with the architectural components in the Hexa-X M&O architectural design: The Network Level DEs would be also implemented by specific Management Functions, but probably in this case with highest support of the associated AI/ML Functions, and also, with an intensive usage of the API Management Exposure block, in order to integrate information from all layers in the architecture. This API Management Exposure block could also integrate the MTBS functionality regarding the data translation services. The ONIX component would be also implemented by a specific function (or set of functions) in the Management Functions block, specifically those in charge of implementing the Infrastructure Layer control-loops associated to the infrastructure discovery tasks.
- Regarding GANA Levels, the approach could be implemented relaying on the different control-loops depicted in the Hexa-X M&O architecture that would be implemented through the different functions in the Network Layer: Level-1 could be associated to the Infrastructure Layer Control Loops (for certain specific MEs) or certain Network Layer Control Loops (for certain NEs requiring low cognitive resources). Level-2 could be associated to the Network Layer Control Loops and the Service Layer Control Loops in charge of those functions associated to this level, i.e.: routing, forwarding, QoS, mobility, monitoring, and service/application. Level-3 would be implemented by the control loops associated to the *Node-Main DE* function, feeding the DEs GANA defines for this layer, i.e., security and fault management (relying on the Security Functions), auto configuration and auto discovery, and resilience and survivability. Finally, Level-4 (the highest orchestration level) would require a more holistic view and interaction with all the Hexa-X architectural layers (mainly Service, Network and Infrastructure Layers), involving control loops in all these layers to implement the Outer Control Loop concept in the GANA architecture. It should be noted that those control loops represented in the Hexa-X architectural design (see Figure 6-1) are not just four individual control loops, but an abstraction representing four sets of control loops. Hence, these sets may contain the specific control loops referred in the GANA model.

However, GANA is a highly distributed approach in which it is considered managed objects have embedded control loops, since in Hexa-X this is not considered by itself. This should be taken into account in case of possible implementations of the GANA concepts in the Hexa-X M&O architecture, where a more centralised approach for the control loops is envisaged.

9.1.1.5 ETSI ENI

The grand objective of the Experiential Networked Intelligence (ENI) industry specification group (ISG) is to use artificial intelligence (AI) techniques, specifically machine learning (ML) algorithms, to autonomously manage and orchestrate the services and operations of the assisted systems [Eni018]. According to the ENI ISG, the assisted system is "*the system that the ENI system is providing recommendations and/or management commands to*" [Eni004].

Considering the ENI principles, the Hexa-X management and orchestration framework illustrated in Figure 6-1 may be considered a collection of assisted systems on which the ENI System may make recommendations and predictions to the different layers in order to intelligently manage its operations, processes, and resources. The ENI System may autonomously collect a large collection of data related to the performance metrics and lifecycle management of network slices in order to understand their configurations and operational statuses in real-time, and then employ ML algorithms to enable intelligent network slice deployment, resource management, monitoring, maintenance, predictions, and other operations. The overarching goal of incorporating intelligence and automation as part of the Hexa-X architectural framework would be to enhance network efficiency, improve the performance of network slices, and automate complex human-dependent decisions and processes, among other tasks [Eni018].

Within the Hexa-X M&O architecture, the ENI System could be implemented by relying on the AI/ML Functions block (Figure 6-1). To accomplish this, several ENI functions may be added inside this block in order to process historical data from assisted systems, employ ML algorithms, and produce recommendations. The ENI recommendations and predictions could be employed to automate operations and gain a better understanding of probable future events in the different Hexa-X M&O architecture layers (Design Layer, Network Layer, Service Layer, and Infrastructure Layer). Some components of the ENI system could be also incorporated into the functional blocks of the Service Layer, into other various NFs in the Network Layer (such as the RAN functions, CN functions, security functions, and others from those in Figure 6-1), and also in certain components in the Infrastructure Layer. Based on the SBMA approach, the different components in the Hexa-X M&O architecture could use their own APIs to deliver input data to the ENI System through the API Management Exposure. On the other hand, the ENI System could also interact via its own customised API broker. After the input data is received by the ENI System, it would be delivered to the ENI components for processing.

9.1.1.6 ETSI SEC

The ETSI SEC group produced several reports and specifications related to security in ETSI NFV. In particular, the specification ETSI SEC 013 [Sec013] focuses on Management and Monitoring. This specification is to be deprecated and replaced by ETSI SEC 024 [Sec024], still in draft status, which shows that the security topic is still an ongoing issue for ETSI and offers room for improvement.

The focus of this section are the notions included in ETSI SEC 024 [Sec024]. In this specification, ETSI introduces new entities dedicated to security focusing on the ETSI NFV MANO standard (Section 9.1.1.1). The central actors of the security architecture are the NFV Security Managers (SMs). There may be several SMs active at the same time, to split roles and responsibilities, and typically to handle separate trust domains. They are connected to each one of the ETSI NFV MANO abstractions: NFVO, VNFM and VIM. The interaction between those entities and the SM is described in ETSI IFA 026 [Ifa026].

As the purpose of ETSI NFV MANO is to handle the LCM of VNFs and NSs, the role of the SM is to interact with the LCM events. To do so, the SM can follow three modes: Passive, Semi-Active or Fully-Active. In these modes, the SM respectively listens to LCM events without intervention, creates security policies as well as mitigation action requests, or provides a systematic approval on the LCM action proposed by the NFV MANO framework, with the ability to modify them or request additional ones, depending on internal security policy. To fulfil these roles, SMs rely on the interfaces defined for each of the three ETSI NFV MANO abstractions: NFVO, VNFM and VIM.

ETSI GS NFV IFA 033 [Ifa033] describes the interface between the SM and the NFVO and assumes that other interfaces can rely on the NFVO as a proxy. To summarize, these interfaces are used to expose telemetry metrics, VNF and NS metrics and LCM events to the SM. Depending on its mode, the SM may respond with LCM orders to enforce its security policy. In addition to the SM, two other entities are introduced by ETSI SEC: the OSS/BSS Security Managers

(OSSMs) and the Security Agents. The role of the first one is still to be defined. However, as it lies in the application layer, it can be seen as a client for the SM. Security Agents are security functions that are deployed by the SM to fulfil various security-related tasks.

The Hexa-X M&O architecture can be aligned with the SM concept described in ETSI SEC 013 [Sec013], whenever the cloud and VNF/CNF management is implemented using Management Functions according the ETSI NFV MANO standard, as explained in Section 9.1.1.1. In the current stage of interface specification, where only the SM-NFVO interface is described, the SM could be fully mapped to the Security functions block in the Network Layer (Figure 6-1), with the Network Layer M&O block representing the NFVO and VNFM blocks, as described in Section 9.1.1.1. This mapping will remain valid if the SM-VNFM interface is added. The addition of the SM-VIM interface would require including the Infrastructure Layer Security M&O block along with the Network Layer Security M&O, in order to match the capabilities of the SM, as the later one does not have direct access to the Infrastructure Layer M&O block. The OSSMs could be also associated to the Service Layer Security M&O block, i.e., the client of the Network Layer Security M&O in this case. Security agents can be mapped to the security functions that the Network Layer Security M&O block would deploy to fulfil security tasks in the network.

Based on the current information available in the ETSI NFV SEC 024 draft, it can be concluded that the Hexa-X proposed Security Functions, along with Infrastructure, Network and Service Layer M&O blocks would be capable of providing the security functionalities proposed by the ETSI SEC specification. Nonetheless, it is important to remark that the ETSI work is still in progress, and that further capabilities may eventually be added. In any case, it is clear that the definition of security interactions between the different layers is still an ongoing work, and that this field is still open to contributions.

9.1.2 3GPP

9.1.2.1 3GPP SA2

Some of the active working items of the 3GPP TSG SA WG2 (Architecture) are directly inspiring and impacting the architectural choices of the Hexa-X M&O system. Particularly relevant are the following working items:

- 5G System Enhancements for Edge Computing (phase 2): The Hexa-X M&O system is designed to orchestrate resources from edge and extreme-edge domains, handling their specific constraints (nomadicity and mobility characteristics, limited computing capabilities, power constraints, sharing of resources with user's applications) and interacting with a variety of edge platforms potentially owned and managed by different administrative entities.
- 5G system support for AI/ML-based services: Hexa-X M&O system introduces AI/ML functions as part of the SBMA (see Section 6.2.2) to optimise the M&O system decisions and increase the efficiency of the network automation strategies. The provisioning of AI/ML functions and the management of the related pipelines can be orchestrated as part of the single network slice management or with a wider scope for the overall infrastructure management. Moreover, AI/ML functions can be also deployed on-demand or automatically, in a distributed manner exploiting the edge computing resources, to support the services' and applications' logic following the AIaaS approach. Further details on the applicability of AI/ML to network data analytics are available on the subsection "Integration of the Network Data Analytics Function (NWDAF)" below.
- Enhancement of network slicing (phase 3): In 3GPP specification, these enhancements cover the topics of (i) network slice admission control in coordination with the RAN and (ii) network slice roaming. Hexa-X M&O includes Monitoring Functions and Management Functions at the Network Layer that can support the related decisions and trigger the execution of such control actions. Additionally, in Hexa-X the network slices span from the extreme-edge to the core, integrate multiple technologies at the access and transport level (e.g., NTN resources) and inherit the concept of "networks of networks".

In this sense, the Hexa-X M&O functions will need to extend the scope of the 3GPP network analytics (see the subsection below "Integration of the NWDAF") and slice control functions to cover these additional domains and technologies.

- Enablers for Network Automation for 5G (phase 3): Hexa-X M&O system adopts and extends the concept of closed-loop for network automation, introducing explicitly four sets of control loops assisted by AI/ML techniques at the service, network and infrastructure layer, with an additional one for DevOps in support of automated service design and deployment.
- Enhanced support of Non-Public Networks (phase 2): Hexa-X M&O system targets private networks as specific external infrastructures that contribute to the E2E network slicing and connectivity, with the possibility to control their resources as part of the inter-domain continuum relying on the functionalities offered for API Management Exposure.

Integration of the NWDAF

Apart from the previous items, special attention should also be paid regarding the integration of the so-called NWDAF, since data analytics techniques is one of the main points of interest in Hexa-X (see Section 7.2.4).

The NWDAF main purpose is to collect data from diverse sources in the 5GC, Cloud, and Edge networks in order to provide network analysis upon request from other NFs, facilitating data-driven automation. The NWDAF was initially defined in 3GPP Rel. 15 (but limited to a single use case); after that, more functionalities were added in Releases 16 and 17, and more extensions are expected also for the next Rel. 18 [GMB+19]. In the 3GPP Data Analytics architecture, the NWDAF works as the hub for all the agents that could produce and/or consume data analytics, being its main objective to integrate data analysis as part of the regular network operation procedures, instead of keeping data detached from network operation, as it was the common practice in pre-5G networks. In relation to the management plane, the NWDAF can act as one of the consumers of the Management Data Analytics Service (MDAS), which provides a capability for processing and analysing data related to network and service events and status, including performance measurements, QoE reports, alarms, configuration data, network analytics data, or service experience data from application functions, among others [28.104].

The NWDAF performs three basic functionalities: (i) receives historical data coming from the network, (ii) computes analytics based on the received data and a specific data model (e.g., and AI/ML model), and (iii) shares the analytics with other functions in the network (the data analytics consumers). Analytics information are either statistical information from the past events, or predictive information. The following analytics information can be provided [SMK+20]:

- Abnormal or expected behaviour information for UEs.
- Communication patterns or mobility related information for UEs.
- Load level of NSIs.
- Congestion information of user data in a specific location.
- Network load performance in an area of interest.
- NF load analytics information for a specific NF.
- Service experience for an application or for a network slice.
- QoS change statistics and potential QoS change in a certain area.

To get this information, the NWDAF can interact with different entities for different purposes:

- It collects data based on subscription to events provided by application functions in the service layer, the 5G Core (5GC) functions (e.g., AMF, SMF, PCF, or UDM) in the network layer, and the management functions in the network layer as well (this makes possible to enable data-driven closed control loops involving functions in these layers).
- It provides on demand analytics to the analytics consumers.
- It can (optionally) perform analytics and data collection using the DCCF (Data Collection Coordination Function).

- It can (optionally) store and retrieve information from ADRF (Analytics Data Repository Function), using it as a data storage resource to store and retrieve the collected data and analytics.
- It can (optionally) perform analytics and data collection using the MFAF (Messaging Framework Adaptor Function) for very heavy data transfers (based on messaging frameworks such as Kafka or Rabbit-MQ, among others).

The NWDAF itself may contain the following logical functions (one of them, or both):

- The Analytics Logical Function (AnLF), which performs inference, derives analytics information (i.e., derives statistics and/or predictions based on Analytics Consumer request) and exposes analytics service.
- The Model Training Logical Function (MTLF), which trains ML models and exposes new training services (e.g., providing trained ML model).

The NWDAF concept aligns well with the Hexa-X approach, especially regarding those requirements about using AI/ML techniques to enhance service M&O operations (Section 5.2.2), and about providing an advanced monitoring system able to collect, aggregate and dispatch data from all managed network segments, integrating infrastructure, user plane and control plane related data from different sources (Section 5.2.1). The NWDAF could also help to standardize automated control loops, as those defined in Section 7.2.3.3.

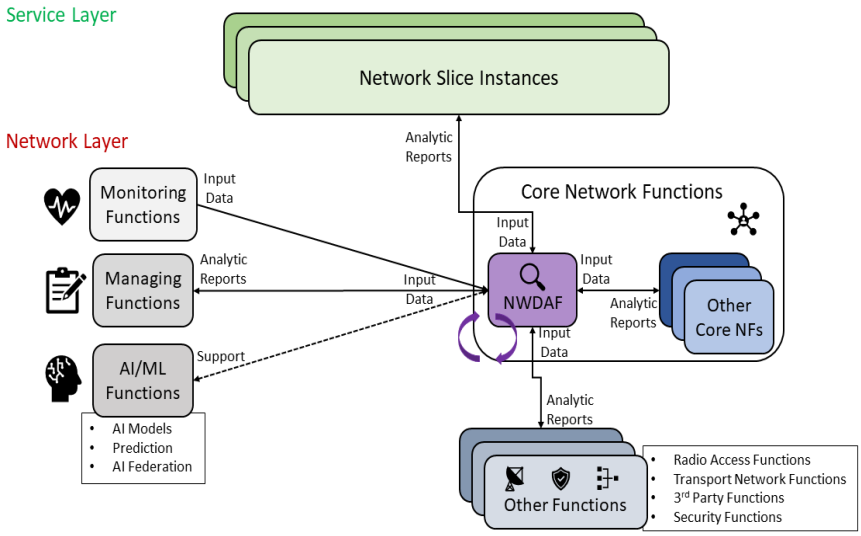

Figure 9-2. NWDAF mapping to Hexa-X M&O architecture.

As it is currently defined, the NWDAF could be mapped into the Hexa-X M&O architecture as part of the CN Functions block (i.e., outside the M&O scope delimited by the red-dashed line – see Figure 6-1). However, it should be close related to the Monitoring Functions block (to receive monitoring data), the AI/ML Functions blocks (to support the Model Training Logical Function), and certain functions in the Management Functions blocks (to perform the M&O related actions). The Management Data Analytics Function (MDAF), which is also part of the 3GPP Data Analytics architecture, could be also mapped as part of the AI/ML Functions block or the Management Functions block, as [28.104] states that the MDA Service (MDAS) may be consumed by different sources, and that the MDAF may consume analytics data offered by the NWDAF. The communication with these and other components in the architecture (other 5GC functions, application functions, etc.) would be done in a cloud-native basis by using the API Management Exposure.

9.1.2.2 3GPP SA3

The main purpose of 3GPP Technical Specification Group Service and System Aspects Working Group 3 (TSG SA3) is to provide security-related requirements and specifications for 3GPP

systems. This working group is mainly focused on the security risks and requirements for the functional application layer: intrinsic security of the functions, security of their APIs and security of the communication between them. Although those security aspects are critical for any generation of mobile networks, they are independent from the M&O system, and only depend on the functions themselves. Fulfilling 3GPP security requirements involves adding some security features, such as secured communication protocols. Those features may be implemented directly by modifying the service functions, or by adding dedicated security functions to provide the security service. Both cases are covered by the Architectural framework proposed by Hexa-X (see Figure 6-1).

3GPP SA3, however, produced a series of studies, some of which featuring requirements for the M&O system. The 3GPP system includes the possibility for a slice customer to perform LCM actions over a NSI. The actions the customer is allowed to perform, if any, are dictated by the Network Slice provider policy. The Communication Service Management Function (CSMF) receives the requests from the client, translates them and transmits them to the Network Slice Management Function (NSMF) to perform corresponding LCM operations. In the 3GPP study on security aspects of 5G network slicing management [33.811] four key issues affecting the management interface between the Communication Service Management Function (CSMF) and the NSMF are identified: (i) unauthorised access to the NSMF interface exposing management capabilities, (ii) protecting the monitoring information coming from the NSI, (iii) protecting the NSS template (both during onboarding and subsequent storage) and (iv) securing the capabilities negotiations, during which a client is offered services by a network operator. The document details the threats and security requirements for those different issues, as well as potential solutions. Those solutions typically involve using secured communication protocols, integrity checks or robust authentication methods.

Just as 3GPP, Hexa-X M&O architecture (see Figure 6-1) displays both, a Network Slice and a NS manager, and the same security issues may be faced. Hence, the solutions proposed by 3GPP can be considered during the implementation of the M&O blocs. However, they do not affect the overall M&O architectural framework. Consequently, regarding the security of the M&O framework proposed in this document, it is aligned with the requirements expressed by 3GPP SA3, as their implementation is possible.

9.1.2.3 3GPP SA5

The mission of TSG SA5 is to specify requirements, architecture, and solutions for the 3GPP management system, which takes care of the provisioning, fault supervision and performance assurance of 3GPP NFs and associated services, including network slicing. To support M&O of 5G networks, the 3GPP management system is built on two levers: 5G Network Resource Model (NRM) [28.541] and Service Based Management Architecture (SBMA) [28.533]. On the one hand, the NRM is an IM that represents the manageable aspects of 5G networks. It specifies the relationships across managed resources, each represented as a separated Information Object Class[10] (IOC), based on which Managed Object Instances (MOI, i.e., objects) [11] can be created as shown in Figure 9-3, i.e.,

[10] An IOC represents the management aspects of a 3GPP 5G network resource. It describes the information that can be passed/used in management interface. IOC has attributes that represents the various properties of the class of objects. Furthermore, IOC can support operations providing network management services invocable on demand for that class of objects. An IOC may also support notifications that report event occurrences relevant for that class of objects. For example, Network Slice IOC and NSS IOC are used to model the management aspects of a 3GPP network slice and NSS, respectively.

[11] A MOI is an instance of an IOC. Multiple MOIs (objects) can be created from an IOC (class). For example, multiple MOIs can be created from the Network Slice IOC, each associated to a different Network Slice Instance (NSI). Similarly, multiple MOIs can be created from the NSS IOC, each associated to a different NSS Instance (NSSI).

- vertically, the 5G NRM focus supports modelling 5G managed resources. These resources include Next Generation Radio Access Network, 5G Core, Network Slice as well as Generic NRM (be reused or inherited by another domain specific model).
- horizontally, the 5G NRM provides Stage 1, Stage 2 and Stage 3 definitions for 5G managed resources. Stage 1 ("requirements-level" stage) intends to provide conceptual and use case definitions for a specific network resource as well as defining subsequent requirements for this resource. Stage 2 ("information service [IS] - level" stage) provides the technology independent specification of a managed resource. Stage 3 ("solution set [SS] - level" stage) finally provides the mapping of IS definitions (UML) into one or more technology-specific Solution Sets (YAML, YANG) [Jin19].

On the other hand, the SBMA represents an architectural style based on replacing traditional telco communication patterns, based on having point-to-point interfaces (e.g., 3GPP Itf-N) between Management Functions, with cloud-native communications patterns, with the production/consumption of management services using Representational State transfer (REST) interfaces and a service mesh topology. A comprehensive view of SBMA features and components in 3GPP management system can be found in [Nok20].

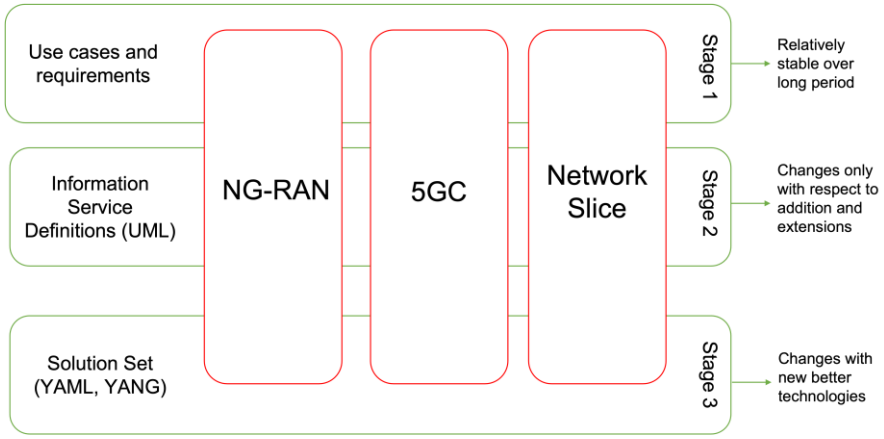

Figure 9-3. 3GPP 5G NRM [Pin19].

Figure 9-4 shows a summary of all the relevant 3GPP SA5 specifications, and their relationship with the HEXA-X work on management and orchestration. As it has been detailed, most of the specifications have served as a basis for the design of primary M&O functions, scoping HEXA-X network and service layers (HEXA-X infrastructure layer is currently out of scope of 3GPP SA5); indeed, SA5 work has been evolved/extended for primary M&O functions, so that their capabilities comply with expectations and target goals of HEXA-X project.

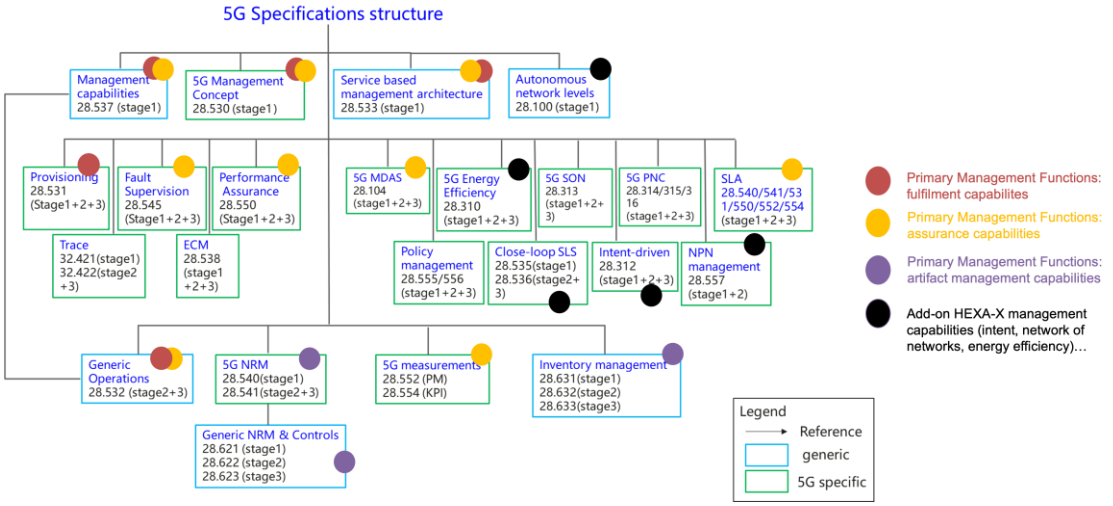

Figure 9-4. 3GPP SA5 specifications and relationship with HEXA-X work on M&O.

The specifications captured above are the result of the work done during the three first releases of 5G, namely Rel-15, 16 and 17. With the 3GPP coined "5G-Advanced era", starting on Rel-18, a new list of work items have appeared. Table 9-2 captures the SA5 Rel-18 work items that, at the time of writing, are within the scope of HEXA-X, and thus will be carefully tracked during project's lifetime.

Table 9-2. SA5 Rel-18 work items relevant for HEXA-X work on M&O

Rel-18 work item (Acronym)	Work item description and impact to HEXA-X
Study on Network and Service Operations for Energy Utilities (FS_NSOEU)	This work item has the mission to study how MNOs can provide standardised monitoring info (and performance reports) to energy utility service providers. Energy is a key vertical industry in HEXA-X, based on which a number of use cases are defined in WP1. It is important to incorporate solutions and recommendations out of this study into HEXA-X M&O system, in order to ensure extreme high reliability (quasi-zero service disruption) and quick service recovery (in case of disaster) in the mission-critical communication infrastructures used for energy transportation and delivery.
Study on AI/ML management (FS_AIML_MGMT)	This work item has the mission to study use cases, requirements and solutions for the management of AI/ML capabilities in AI/ML-enabled functions (e.g., NWDAF, Management Data Analytics Function -MDAF-, RAN intelligence, etc.) in 3GPP 5G. These capabilities include model deployment, testing & validation, and configuration and performance of AI/ML-enabled functions. The outcomes of this study will provide guidelines on how to keep improving AI/ML functionality in HEXA-X M&O system, shedding light on demarcation point between the role of AI in managed resources (NFs) and Management Functions, and coordination across them.
Enhancement of Energy Efficiency for 5G Phase 2 (EE5GPLUS_Ph2)	This work item has the mission to address the cross-SDO issues related to energy management, for the purpose of coordination, and defining new KPIs for Energy Consumption and Energy Consumption computation. Energy efficiency is a key primer of HEXA-X, and that is something that needs to be managed from the M&O stack, at both provisioning (fulfilment) and operation (assurance) time. The outcomes of this work item may influence procedures and workflows defined in HEXA-X Management Functions, with additional input or configuration options.
Study on KQIs for 5G service exposure (FS_KQI_5G)	The objective of this work item is to study the impact on Key Quality Indicators (KQIs) specification from both provider and customer views, and how to manage KQI models in relation to existing KPI and SLS constructions. HEXA-X is a user-centric system with the primer goal of enhancing customer experience, leveraging tighter app-network integration. The outcomes of this work item will be useful for HEXA-X M&O stack, specially to integrate standard system KQI specifications and evaluation method for service experience targeting HEXA-X Business-to-Business services and Business-to-Consumer applications.
Study on enhancement of management of	The objective of this work is to study enhanced management of PNI-NPN instantiations, by taking into account Rel-17 work in RAN and CN side, together with access control and multi-tenancy support. The use cases, potential solutions and recommendations resulting from

non-public networks (FS_OAM_eNPN)	this study will be incorporated to the 'API management exposure' module. This module is the HEXA-X M&O system facilitating the interaction with 3rd party consumers, which allows for the realisation of advanced PNI-NPN scenarios, and in more general, the 'network of networks'.
Enhancement of the management aspects related to NWDAF (FS_MANWDAF)	This work item has the mission to improve management of NWDAF aspects, from both modelling and performance behavior perspective. For the modelling side, the objective is to modify the NRM for NWDAF to proceed with the decomposition into MTLF and AnLF functionality. NWDAF performance behavior is described using scalability KPIs, model accuracy KPIs, and efficiency KPIs. The outcomes of this study will be used to improve the analytics capabilities in HEXA-X management and orchestration, as well as help clarify the demarcation point between MDAF and NWDAF for some scenarios.
Study on evaluation of autonomous network levels (FS_ANLEVA) Study on enhancement of autonomous network levels (FS_eANL)	These work items articulate their objectives around autonomous networks, leveraging input from other relevant industry fora such as ITU-T, GSMA and TMForum. The mission of these studies is precisely to extend SA5 management capabilities to make 3GPP management system aware of network autonomous levels, assisting in network operation accordingly. In HEXA-X, the "self-"tag in fulfilment and assurance activities is a must, and therefore the insights produced by these two work items will be carefully taken into account when developing solutions and improving system design in upcoming deliverables.
Study on enhanced intent driven management services for mobile networks (F_eIDMS_MN)	This work item will keep working on intent LCM automation (e.g., around detection of conflicting requirements and their resolution), improvements for intent model and model extensions, according to existing discussions in TMForum. HEXA-X work on IBN is a top-ranked priority, and therefore this study is quite relevant. Bidirectional interactions/contributions between 3GPP SA5 and HEXA-X is expected during project's lifetime.

9.1.2.4 3GPP SA6

The most relevant item delivered from the 3GPP TSG SA WG6 (Application Enablement & Critical Communication Applications) and applicable to the design of the Hexa-X M&O system is the Common API Framework (CAPIF) [23.222], which enables a common approach for a unified north-bound API framework across the various functions defined by 3GPP. These unified interfaces are the key to enable the secure and scalable interworking between several NFs and service-level applications, even coming from third parties. The CAPIF functionalities have inspired the design of the API Management Exposure in Hexa-X and include mechanisms for on-/off-boarding of API invokers, registration and release of APIs to be securely exposed to third parties or external domains, dynamic API discovery, authentication, authorisation, logging and charging, inter-domain interactions and federation of domains acting as CAPIF providers, with support for distributed deployments of NFs and applications.

9.1.3 TMF Zoom

TM Forum's Zero-touch Orchestration, Operations and Management (ZOOM) [TmZ22] is part of the TMF ODA framework [TmO22], incorporating concepts such as model-driven orchestration and automated onboarding. Quoting from [Mil14], ZOOM addresses the following challenges:

a. In virtual operations, the relationship between networks and services is dynamic,
b. Zero-touch, self-service operations that can respond with speed and agility,
c. Adaptive automation,
d. Customer-centric services,
e. Support technology-driven innovation.

The ZOOM approach states that the existing policy-based management concepts, such as the Policy Decision Point (PDP) and Policy Enforcement Point (PEP) are not adequate to address the new needs for zero-touch management. A reason among others is that the notion of conflict detection and resolution is completely lacking in their definitions. In addition, it is unlikely that the Event-Condition-Action (ECA) model can meet the new needs. Two important limitations are (1) it is difficult to express declarative policies in this format, and (2) it is difficult to express other types of policies (e.g., goals), since by definition this type of policy rule explicitly defines a single next state.

ZOOM has defined a policy management architecture, referred to as FOCALE (Foundation – Observe – Compare – Act – Learn – rEason), which describes its novel operation in terms of the structure of its unique control loops. The original architecture was defined in [SAL07]. FOCALE is based on the following six core principles:

a. Use of a combination of information and data models to establish a common "lingua franca" to map vendor- and technology-specific functionality to a common platform, technology, and language independent form.
b. Ontologies are defined to attach formally defined meaning and semantics to the facts defined in the models.
c. Use of the combination of models and ontologies to discover and program semantically similar functionality for heterogeneous devices independent of the data and language used by each device.
d. Use of context-aware policy management to govern the resources and services provided.
e. Use of multiple control loops to provide adaptive control to changing context
f. Use of multiple machine learning algorithms to enable FOCALE to be aware of both itself and of its environment to reduce the amount of work required by human administrators.

The ZOOM architecture covers all aspects to implement the above architecture (policy IM, IM, etc.).

The FOCALE principles, in particular for the adoption of multiple control loops in support of adaptive control and the usage of ML to reduce the human interventions and thus increase the level of network automation, have been also considered in the design of the Hexa-X M&O architecture. This is confirmed by the presence of multiple closed-loops in the M&O architecture structural view (Figure 6-1), each of them operating at different layers and with different scopes. Moreover, the Hexa-X Monitoring and AI/ML functions are used by the Hexa-X Management Functions to enhance the logic driving their actions. In this sense, there is a clear mapping between some of the FOCALE policy management phases and Hexa-X M&O functions, as follows:

- Observe: Monitoring Functions in Hexa-X M&O architecture structural view.
- Compare: AI/ML Functions in Hexa-X M&O architecture structural view.
- Act: Management Functions in Hexa-X M&O architecture structural view.
- Learn: AI/ML Functions in Hexa-X M&O architecture structural view.
- Reason: AI/ML Functions in Hexa-X M&O architecture structural view.

9.1.4 GSMA

The management of E2E network slices in the Hexa-X M&O architecture (see Figure 6-1 – Structural View) is handled by the Management Functions at the network layer. Network slices can be requested on demand by verticals, defining their service requirements or even using high-

level intent declarations as discussed in Section 7.2.2.1, and they are management automatically during their lifecycle through the fulfilment and assurance capabilities of the Management Functions. In this context, the modelling of network slices for the requests coming from the verticals can be described following the GST [Ng116] defined by GSMA. This template offers a customer-oriented description of the slice requirements, providing a good compromise between the simplified level of the technical specification (suitable for the understanding of non-specialised customers', like verticals) and the type of details required to univocally translate such requirements into a given slice specification. This translation is handled by the M&O system, e.g., through dedicated Management Functions of the M&O architecture structural view and gives as output a fully-technical network slice definition, e.g., following the NRM defined by 3GPP, which is then used for the provisioning actions.

9.1.5 IETF/IRTF

In [HX21-D61], the following IETF working groups were reported: OPSAWG, NETMOD, ANIMA and SFC. Their RFCs have driven the capabilities that are available at all HEXA-X system architecture layers: Infrastructure Layer (e.g. OPSAWG models for L2/L3 VPN service provisioning), Network Layer (e.g. ANIMA work on GRASP for autonomic communications between xNFs), Service Layer (e.g. SFC solutions for flexible and dynamic composition of xNF/Netapps into E2E communication and digital services), and Design Layer (e.g. NETMOD work on YANG language, notably used as data model in service/function/infrastructure templates).

However, the add-ons and novel capabilities come from the work in two Internet Research Task Force (IRTF) groups: COINRG and NMRG. Table 9-3 details the outcomes of these two research groups which are relevant for the HEXA-X service management capabilities reported in the present deliverable.

Table 9-3. IRTF research groups relevant for HEXA-X work on M&O.

IRTF research group: <LIST OF> in-scope documents	Impact to HEXA-X
COINRG (https://irtf.org/coinrg) 1. Use cases for in-network computing 2. Enhancing security and privacy with in-network computing 3. Transport Protocol Issues for In-network Computing Systems 4. Edge Data Discovery for COIN (draft-mcbridge-edge-data-discovery-service)	References 1, 2 and 3 provide design principles and solutions for IoT-to-edge-to-cloud continuum, with impact on the management infrastructure layer. Reference 4 provides a solution for distributed data discovery, which may require both marshalling of data at the outset of a computation and the persistence of the resulting data after the computation. The standard approach captured in this reference is relevant for the infrastructure layer discovery capabilities of Management Functions.
NMRG (https://irtf.org/nmrg) 1. Intent-Based Networking: Concepts and Definitions (draft-irtf-nmrg-ibn-concepts-definitions) 2. Intent Classification (draft-irtf-nrmg-ibn-intent-classification) 3. Interconnection Intents (draft-contreras-nmrg-interconnection-intents)	References 1 and 2 lay out the foundation for the intent work in service and design layers. Reference 3 provide advances to extend this work to multi-domain scenarios, involving service provisioning across different administrative domains. References 4 and 5 illustrates the applicability of the autonomic networking in system architecture, with the reference 6

4. Autonomic Networking: Definitions and Design Goals (RFC 7575) 5. Autonomic Networking Use Case for Distribution Detection of SLA Violations (RFC 8316) 6. SOAR (Security Orchestration Automation and Response) based Native Network Management to Optimise an adaptive B5G Network (draft-kim-nmrg-nnmb5g) 7. Opportunities of Flexible Addresing and Protocols in Digital Twin Network (drat-li-nmrg-dtn-addressing-protocols) 8. Digital Twin Network: Concepts and Reference Architecture (draft-zhou-nmrg-digitaltwin-network-concepts) 9. An Efficient Data Collection method for Digital Twin Network (draft-zhu-nmrg-digitialwin-data-collection) 10. Artificial Intelligence Framework for Network Management (draft-pedro-nmrg-ai-framework)	being the most aligned with HEXA-X scope of work. References 7, 8, 9 and 10 touch on the digital twinning concept and the impact on management domain. Of special interest is the work in 9, which has direct impact on HEXA-X Monitoring Functions across all architecture layers. Finally, reference 10 provides a framework for distributed AI in management plane. The principles, requirements and solutions captured here can help HEXA-X system operator(s) to better position AI/ML solutions for their zero-touch operation goals and associated use cases.

9.2 Other standards

9.2.1 Kubernetes

The Hexa-X M&O architectural design relies on different infrastructure resources, both physical and virtual. Regarding virtual resources, although a few years ago the by-default approach were VMs, it is envisaged that the majority of softwarised functions in 6G networks could be based on *Software Containers*, a more modern and lighter virtualisation technology [TRA15]. The CNFs (*Containerised* Network Functions) that have been repeatedly referred through this document are based in this new containers' technology (in the same way VNFs were based on VMs).

Unlike VMs, container-based software development is often based on producing very small software components know as microservices, that is, rather than having a small set of VMs, it is common to break down software applications into a large number of small containers, each running a specific service. In production environments it can be common to find applications made of tens, hundreds or even thousands of containers, which can be of course challenging for managing using legacy techniques. Here is where container's orchestration comes into play.

The *de-facto* open-source container orchestration solution is Kubernetes (typically abbreviated as K8s) [K8s22]. It is a powerful tool that makes container-based applications easier to develop, faster to deploy and more reliable to operate. It was originally designed by Google and donated to the Cloud Native Computing Foundation [Clo21], which is part of the Linux Foundation [Tlf22]. There are also other similar containers orchestration technologies such as Red Hat's Open Shift [OpS22], Docker Swarm [DoS22] or Apache Mesos [ApM22]) but as mentioned, K8s seems to be the *de-facto* standard while this document is being written [GoT22] [GoT19].

K8s is a CaaS platform that can be deployed using single-node or multi-node approaches, this group of nodes (physical servers or VMs) provide storage, computing, and networking resources for executing containerised applications. This group of nodes is typically known as a *K8s Cluster*. Kubernetes clusters consist of a *control plane* and a set of *worker nodes*. The control plane is typically referred as the *master* node, being responsible for managing the rest of the infrastructure

through its API. The basic units in Kubernetes are called pods, which add a higher level of abstraction to containers (a pod consists of one or more containers, grouped together, that are the components for processing the application workload). Pods are deployed on *worker nodes*. A detailed K8s architecture with detailed information about the different components can be found in [K8Co22].

Although K8s is known as a container orchestration system, it claims that it is not a *mere* orchestration system, since in fact it eliminates the need for orchestration. This statement is based on that the technical definition of orchestration is "the execution of a defined workflow" (e.g., first do A, then B, then C), but in contrast, K8s comprises a set of independent, composable control processes that continuously drive the current state towards the provided desired state, i.e., It shouldn't matter how to get from A to C, being centralised control also not required. Based on this, K8s claims: "to be easier to use and more powerful, robust, resilient, and extensible [K8Wh22]".

As a whole, Kubernetes provides the following capabilities[K8Wh22]:

- Cluster creation, oriented to orchestrate a group of HA nodes that are connected to function as a wholesome environment.
- Application deployment (on the created clusters) and update.
- Application scaling (to meet the required demand).
- Service Discovery and Load Balancing: K8s can expose containers using their DNS names or their own IP addresses. When container traffic is high, Kubernetes can balance and distribute network traffic, so deployments are resilient.
- Storage Orchestration: With support for different storage services (e.g., local storage, public cloud providers such as GCP, AWS or Azure, and network storage system such as NFS, iSCSI, or Cinder).
- Automatic rollout and rollback: K8s can describe the target state of a deployed container and move the current state to the target state in a controlled way. E.g., it is possible to automate operations to create/delete containers, or merge resources into new containers.
- Automatic bin packing: It is possible to provide K8s with a cluster of nodes that can be used to run containerised tasks. They tell K8s how much CPU and memory (RAM) each container needs. K8s can place containers on the nodes to fully utilize the available resources.
- Self-healing: K8s automatically restarts failed containers, replaces containers, or kills containers not responding to custom health checks.
- Secrets and configuration management: K8s allows to store and manage sensitive information such as passwords, OAuth tokens, or SSH keys on a secure manner. It is possible to deploy and update secrets and application configuration without rebuilding container images or exposing secrets in stack configuration.
- K8s also includes monitoring capabilities, based on an agent that monitors and collects resource utilisation and performance metrics, such as CPU, memory, file usage and container network on each node.
- An API server that exposes an HTTP API that lets end users, different parts of the clusters, and external components, communicate with one another.

Alignment with the Hexa-X M&O architectural design

The Kubernetes solution could align well with Hexa-X's M&O architectural design providing the orchestration resources needed to implement the Hexa-X cloud-native vision. This technology would be functionally part of the Management Functions block (Figure 6-1), being deployed on the Management Units (Section 8.2). It could also help to meet the ambitious scalability requirements for Hexa-X (the M&O system shall be able to scale to support >100 bn of devices – see Section 5.2.1), since K8s has been natively designed to cover even worldwide scale, based on the same principles that allows Google to run billions of containers [K8s22].

Besides this, K8S technology may also facilitate the implementation of the orchestration functionality in a federated way, as it was mentioned in Section 8.3. This could be done by relying

on the *K8s Federation* functionality, also known as KubeFed [KuF22], which is a tool for supporting the LCM of multiple clusters on K8s from a single control point of a group of APIs, named *Cluster API*. The Cluster API provides services such as provisioning, upgrading and operations of the federated clusters. Similar approaches are provided by Google Anthos [GoC22], which is alternative to KubeFed for managing multiple clusters on hybrid clouds composed of third-party cloud environments and on-premises K8s clusters or, Rancher Fleet [RaF22] which is a K8s cluster-fleet controller that aims at addressing the challenges of managing thousands or even millions of K8s clusters across several world geo-locations. This federation of K8s clusters has significant advantages, such as better availability to deploy a CNF when losing a cluster, easier migration of applications between clusters, more extensive scalability, ability to spatially distribute clusters deployed for MEC use, and infrastructure cost reduction through optimised balancing of on-premises and third-party clusters utilisation to deploy workloads.

Another area in which Kubernetes can offer solutions for the implementation of the Hexa-X M&O architectural design is the cloud, edge, and extreme-edge integration (Section 7.2.1.1), for implementing the device-edge-cloud continuum concept. The K8s technology may in fact provide the way to handle a variety of points of presence, from vast datacentres to remote small facilities, which is well aligned with this requirement (see Section 8.4).

Specifically, for reaching the edge and extreme-edge devices (small, hence with limited computing resources) specific lightweight K8s distributions could be used [KA20] [KuE22] [K3s22] [MKs22]. These distributions are specifically designed to work considering the challenging requirements in this environment, such as the mentioned limited set of resources and the low QoS and resilience of the underlying infrastructure, while ensuring a sufficient level of availability, even in the event of a disconnection with the central elements.

Specifically, KubeEdge [KuE22] is an extension of the K8s standard solution to support native containerised application orchestration capabilities into hosts at the edge, while supporting the native K8s APIs. KubeEdge inserts an edge controller into the K8s master node allocated in the central cloud to communicate with the edge environment. It provides specific entities (such as the *Edge Hub*, *Edge Kubelet*, *Lightweight Edge Centric Service Mesh*, and others) to hide the specificities of the management of pods, nodes, and networks in a low resource environment, providing autonomy to the edge. KubeEdge also enables the convergence of communication protocols with IoT devices using a MQTT broker.

Another alternative is K3S [K3s22], a small footprint distribution designed for a resource-constrained environment. K3S claims to work well from something as small as a Raspberry Pi [Pi22] to large 32GiB servers. It allows to create a cluster at the edge by providing two main entities: the *K3S Server* and the *K3S Agent,* which respectively play the roles of the regular K8s *master* and K8s *worker nodes* mentioned above. This implementation leads to an isolation between the cloud and the edge.

MicroK8s [MKs22] is also another lightweight K8s distribution that can run on the edge and on extreme-edge appliances to create single-node or multi-node clusters, and also to orchestrate complete central cloud resources. It can be also installed on small scale devices such as Raspberry PIs [PiM22], including features such as HA, resilience and self-healing.

These specialised variants of K8s consider the specifics of the edge and extreme-edge resources in order to provide an optimised management. However, their integration could affect the cohesion in the orchestration of resources and CNFs between the central cloud and the edge/ extreme-edge domains, which cloud make the level of services offered across the network instance inconsistent in some way. Also, the usage of diverse heterogeneous distributions could impact increasing the complexity. These aspects should be considered for future research.

Clusters Policies Management

Another aspect in which K8s can facilitate the deployment of the Hexa-X M&O architecture is the cluster policies management, which is considered essential due the multi-domain and cloudified nature of 6G NFs.

This is because, from the operator perspective, the deployment of the network in a geographically spread and heterogenous environment might imply undertaking specific measures to maintain availability, isolation and scalability of services, while ensuring compliance with possibly conflicting regulations or policies. The most common issues in multi-cloud deployment involve traffic load balancing, inter-cloud connections isolation, storage decoupling, providing HA, resource consumption optimisation (efficient CNF/VNF placement), facilitation of QoS enforcement (e.g., ensuring low latency in cross-cluster scenarios) and avoidance of vendor-lock situations. Therefore, the multi-cluster management approach must not consider solely the aspects of inter-cluster connectivity or data integrity but should also focus deeply on management of per-cluster and global policies aspects.

There are several K8s-based solutions for multi-cluster environment, such as Kubefed [KuF22], Red Hat Advanced Cluster Management [RHA21], or Google Anthos [Ant22], that provide management capabilities over a federation of K8s clusters. These solutions provide the means for establishing dedicated interconnections between isolated clusters (e.g., via VPN connection) and enforcement of cluster-level policies. In terms of K8s-based clusters the policies are enforced on a namespace level (isolated group of resources within a cluster) by instantiating the policy resources. The mentioned policy resources can be effectively used to implement, in the managed cluster fleet, the recommendations stated in the security framework created by National Institute of Standards and Technology (NIST) [nist18], using constructed templates [Cis20] as shown in [Roj21].

9.2.2 SDN

Software-defined networking (SDN) technology is an approach to network management enabling programmatic network configuration, making it better aligned with cloud computing than traditional static network management systems. It aims to solve the static architecture of traditional networks, by centralising the network intelligence into a network component (the SDN controller) and by separating the data plane from the control plane (the routing process) [CJP+07].

There is a large and evolving number of industry standards, industrial consortia, and open development initiatives involved in creating standards and guidelines for Software Defined Networks (SDN). The Internet Society, ITU-T, and ETSI are all making key contributions to the standardisation of SDN.

As mentioned in Section 8.4, the integration of the SDN controller's logic within the Management Functions (Figure 6-1) can help to leverage the scalability and reliability of the whole system. Aligned with the Hexa-X Deployment View, the implementation of the SDN technology would be done through the Connectivity Units by means of SDN switches (Section 8.2). These switches would provide network connectivity among the different network elements (e.g., legacy PNFs, other MNO networks or third-party networks).

The interconnection of these elements through SDN-enabled switches would boost the network agility and programmability [Met14]. SDN switches would play an important role for flexibly connecting elements in the Infrastructure Layer, serving as link between e.g., extreme-edge, edge and central cloud resources. This would offer a wide range of programable functionalities, enabling the future of 6G and network operators [LCZ+19]. Furthermore, these SDN controllers could also provide connectivity and flexibility between different NFs (see Figure 6-1) i.e., radio access functions, transport NFs, security functions, management functions, CN functions, third party functions, AI/ML functions and monitoring functions. In the following paragraphs it is detailed how SDN controllers can operate on such functions.

- Radio access functions: SDN switches can act providing the connection of different RAN functions. This can improve significantly the reliability and latency of the radio access channel defining the management of resources across the radio sites [SSK20].
- Transport NFs: To minimise transportation costs. A transportation network must be developed, capable of providing the required energy to transition between different nodes

of the network. This can be implemented through SDN switches minimising the overall transportation cost [SPB+18]

- Security functions: The utilisation of SDN switches in the security domain has important benefits. It protects the organisation's ability to function, it enables the safe operation of applications implemented on Hexa-X, it protects the data that Hexa-X collects and uses; and finally, it safeguards the technology that Hexa-X uses.
- Management functions: The ability of managing resources through different functions can be boost by means of SDN controllers. This implementation can enhance the QoS of the network, e.g., accommodating the number of resources needed in advance [LLC+21]
- Core NFs: The integration of SDN controllers in the core NFs of Hexa-X aims at efficiently aggregate data traffic from end devices. This new architecture will give operators the required flexibility to meet the diverse network requirements of all the different stringent 6G use cases, going well beyond high-speed fixed wireless or mobile broadband services.
- Third party functions: To ensure secure and trusted service provisioning it is necessary to implement third party functions by means of SDN controllers. Software defined networks provide this flexibility through the implementation of VNFs. VNFs can be chained across multiple domains to create network applications tailored to the requirements of specific devices, as demonstrated under previous 5G PPP phases [5GPIn19].
- AI/ML functions: The utilisation of SDN switches regarding AI/ML functions aim to minimise, for instance, the time it takes to train a neural network by switching the training phase among different high-speed functions. Furthermore, SDN promises to be an outstanding tool to implement ML functions due to the control plane and the forwarding plane of networks [PJP+16]
- Monitoring functions: SDN can implement monitoring functions in Hexa-X architecture that examine extremely fast the status of the network. The time in these cases is critical since a failure in the network operation can affect the overall performance of the 6G network.

9.2.3 O-RAN

The primary goal of the Open RAN (O-RAN) Alliance is to develop a virtualised RAN architecture based on open hardware, open software, and an open underlying cloud infrastructure in order to eliminate the lock-in of vendors and enable multi-vendor deployments for 6G communication systems [OrA21]. Aligned with the Hexa-X architectural framework shown in Figure 6-1, the O-RAN Alliance has been tackling both the physical and virtual aspects of the RAN architecture with the goal of transforming the traditional RAN (also referred to as vendor lock-in RAN) philosophy into an intelligent, slicing-aware, and multi-vendor interoperable architecture that must operate based on open and disaggregated interfaces [OrA21].

With these objectives in mind, the principles of the O-RAN Alliance could be fully aligned with the edge, extreme-edge, and transportation-related aspects of the Infrastructure Layer of the Hexa-X architectural framework shown in Figure 6-1. If the ongoing efforts of the O-RAN Alliance succeed and the aforementioned principles are applied to the Hexa-X architectural framework proposed in Section 6, the door could be opened to new small businesses and competitors to enter the wireless communication market, as well as an extraordinary level of innovations could be unleashed in the RAN architecture of the next generations of mobile networks.

More specifically, the major innovations introduced by the O-RAN Alliance could also be aligned with the extreme-edge, edge, and transportation aspects of the Hexa-X architectural framework shown in Figure 6-1. The alignment between the O-RAN Alliance and the Hexa-X architectural framework has been classified into three categories: (a) NF alignment, (b) network management alignment, and (c) cloud-site alignment. They are discussed in detail as follows.

- **Network function alignment**: One of the scopes of the O-RAN Alliance is to virtualise the components of the next-generation NodeB (gNB) [OrA21]. To that end, the Alliance

has defined the Central Unit (CU) and the Distributed Unit (DU) as VNFs, referred to as O-RAN Central Unit (O-CU) and O-RAN Distributed Unit (O-DU) in the O-RAN terminology. They are also referred to as virtual CU (vCU) and virtual DU (vDU) in the literature. The Radio Unit (RU) is considered a PNF. In line with the virtualisation of the NFs, CU and DU are also considered as VNFs and RU as a PNF in Figure 6-1. Another scope of the O-RAN Alliance is standardising an open interface between the vDU and RU. This open interface could be integrated into the gNB shown in the Network Layer of in Figure 6-1, which is claimed to lower the total cost of RAN deployment while eliminating proprietary lock-in [OrA21].

- **Network management alignment**: The RAN Intelligent Controller (RIC) is one of the critical components of the O-RAN architecture. The primary function of the RIC is to optimise the RAN architecture and radio resources. The Alliance has divided the RIC into two entities: non-real-time RIC (Non-RT RIC) and near real-time RIC (Near-RT RIC) [OrN21] [OrNe21]. Both RICs were given their names based on the timescale of their operations. The former operates in the hours, minutes, and seconds range, while the latter operates in the tens to hundreds of milliseconds range. The scope of the RIC could be aligned with the scope of the Hexa-X architectural framework shown in Figure 6-1, allowing the M&O Layer block and Network Layer to host several control and data plane functionalities of a gNB, such as mobility management, security, interference management, etc.

  The Non-RT RIC, which operates in the management plane, is an intent-based management entity that enables automation and intelligence (notably ML algorithms) at all levels of the M&O aspects of the RAN architecture [OrN21]. The Non-RT RIC could be deployed close to other Management Functions blocks in the Network Layer, aimed at managing the managed objects of the O-RAN architectural framework. It provides configuration management, facilitates non-real time network and procedure optimisation, monitors faults, manages performance, enables service and policy management, enforces network slicing policies, and makes recommendations to Near-RT RIC [OrN21] [OrS21]. These Non-RT RIC features are referred to as rApps, which are value-added services that require well-defined open APIs to communicate with other components of the non-RT RIC. The M&O Layer block of the Hexa-X architectural framework is an ideal layer where the non-RT RIC may potentially reside in order to add its associated features and functionalities in the M&O of different types of network slices and various kinds of network resources.

  The Near-RT RIC is a network optimisation microservice-based software platform that is used for controlling the components and resources of near-RT applications [OrNe21]. The Near-RT RIC is located close to the managed objects such as O-CU and O-DU in a gNB. It supports a variety of xApps, which are features that operate in near-RT. These xApps include but are not limited to mobility management, traffic steering, QoS management, radio connection management, and security [OrNe21]. They could be designed as microservices and would require the definition of specific open APIs to connect to the components of the Near-RT RIC. The xApps collect near-RT data and information from end-users and cellular sites on a regular basis and transfer it to the Near-RT RIC, which employs the embedded intelligence in order to automate the network operation and enhance network slice performance [OrS21]. The Near-RT RIC may be located in close proximity to the gNB components in the Network Layer as shown in Figure 6-1, aimed at collecting user data through xApps for network automation and intelligent service management [OrNe21].

- **Cloud-site alignment**: The Near-RT RIC, O-CU, and O-DU are hosted by the open-cloud sites, referred to as O-Cloud, on top of an open underlying compute and transportation infrastructure [OrC21]. Each O-Cloud site is composed of physical nodes that can be virtualised to create a number of virtual nodes. To meet end-user requirements for low latency and high bandwidth, the O-Cloud sites are typically connected via optical

fibre, forming an E2E O-Cloud interconnection network [OrC21]. The three logical components of the O-RAN architecture can be deployed in a flexible manner across the O-Clouds based on demographic data, operator requirements, performance objectives, and many other factors. To that end, six deployment scenarios have been introduced in [GC21] to bring substantial flexibility to the deployment of the O-RAN architecture on the O-Cloud sites for 6G communication systems. The O-RAN Alliance's scope in terms of cloud sites and resources is fully aligned with the Infrastructure Layer of the Hexa-X framework structural view, depicted in Figure 6-1, which allows for the deployment of gNB components in a variety of locations using a variety of deployment metrics.

9.2.4 TIP

The mission of the Telecom Infra Project (TIP) is to develop and deploy standardised, open, and disaggregated technical solutions for mobile networks in order to provide high-quality communication services for both, mobile phone users and vertical industries. The TIP covers E2E network solutions, including communication service design, CN, transportation network, and access network. The scope of the TIP and its proposed technical solutions are applicable to all Hexa-X architecture layers. The Service Layer and Design Layer, i.e., the communication service design related projects' outcomes of the TIP, might be used here. The Infrastructure Layer and Network Layer, i.e., the access network related projects outcomes and the CN related projects outcomes of the TIP, might be utilised here.

The Core and Services projects group is researching ways to simplify the CN architecture and the communication service design in order to increase overall network efficiency and flexibility while decreasing total network maintenance and operation costs [TiOc21]. These technical solutions could be incorporated into the Hexa-X architectural framework in order to design network slices autonomously and deploy them over the core cloud sites in a resource-, time-, and cost-efficient manner. In addition, the outcomes of this group are also useful for designing a cloudified, service-oriented, virtualised, and slicing-aware CN architecture atop the Hexa-X architecture to decrease total cost, energy consumption, network complexity, and other factors [TiOc21].

The Access Network projects group is tasked with the responsibility of identifying and developing novel solutions at infrastructure, technological, and methodological levels that are aimed at making it faster and easier for end-users to connect to a mobile data network [TiRa22]. In addition, the Access Network group is concentrating its efforts on resolving several challenges that can obstruct communication between the end users and the Access Network in cellular systems. In Hexa-X WP6 it is considered that the outcome of this group will result in providing recommendations to design and deploy open gNB components and open transportation links in the Network Layer, improving the management and orchestration capabilities, and efficient resource allocation in the Infrastructure Layer of the HEXA-X architectural framework [TiRa22].

Finally, the Transportation Network projects' group is responsible for providing technical solutions to improve backhaul, midhaul, and fronthaul links in order to keep up with the exponential growth of traffic generated by end users and vertical industries [TiT21]. The group is addressing critical issues in transportation networks such as rapid convergence, scalability, extensibility, and others. The outcome of this group may provide guidelines and recommendations for designing a high-capacity backhaul network between the CN and RAN, as well as designing fronthaul and midhaul links between the components of a RAN architecture in the Hexa-X architectural framework (see Figure 6-1).

10 KPIs, KVIs and Core Capabilities

Besides the well-know KPI concept, the previous Hexa-X Deliverable D1.2 [HX21-D12] also defines *Core Capabilities* and *KVIs*. Core Capabilities are intended to both, enhance the performance of traditional services as well as enable paradigm-shifting new services. KVIs

represent intangible, yet important, human, and societal needs. These KVIs might be, in some cases, evaluated directly, but they are usually associated with KPIs, which serve as proxies. In Hexa-X, trustworthiness, inclusiveness, and sustainability have been identified as the main KVI areas:

1) ***Trustworthiness.*** Is oriented to ensure confidentiality, integrity and availability of E2E communications in future 6G networks, and guarantee data privacy, operation resilience and security, building trust on wireless networks as well as its enabled applications among consumers and enterprises — supporting and promoting European ethical values of trust and privacy protection as well as the technological EU sovereignty goal for fostering an open, trustworthy and democratic Europe in the digital age.
2) ***Inclusiveness***. Makes reference to *connect the unconnected*. The intention is to make 6G available for everyone and everywhere (e.g., making possible access to physically disadvantaged people, or to serve people living in rural areas or in less privileged areas of cities).
3) ***Sustainability***. Refers to the transformation of 6G networks into an energy-optimised digital infrastructure that will deeply revise the full resource chains of wireless networks towards sustainability and carbon neutrality. Its digital fabric shall, beyond providing unprecedented connectivity and coverage, also create the ability to sense and understand the state of the physical world in real-time, and as such boost sustainability from the environmental, economic, and social perspectives and importantly deliver effective and sustainable digitalisation tools for global industry, society and policy makers, help achieve UN SDGs and assist the implementation/operation of the EU Green Deal, in particular after the COVID-19 pandemic, towards a circular economy and a sustainable world (see [HX21-D12] Section 6 for a more detailed view on this topic).

But, as mentioned, besides these KVIs, [HX21-D12] also introduces a set of relevant Core Capabilities. From the M&O point of view the following are considered the most relevant:

a) ***Integrated intelligence***. This capability refers to the integration of AI/ML for optimisation of network operations in a pervasive way, as well as optimisation of network operations for optimal performance of AI/ML features. AI/ML Functions are part of the M&O architecture, so this capability is directly related to the M&O function.
b) ***Usage of Embedded Devices***. This comes with the concern that if a wireless device is to be embeddable anywhere, access to an external power source cannot be taken for granted. In addition, the placement of the devices may prevent or prohibit the usage of batteries as a power source. As explained through this document, one of the main innovations regarding M&O in Hexa-X is the integration of extreme-edge resources, which may be partly made up of embedded devices.
c) ***Flexibility***. With the overall move to cloud-native realisation of 5G and beyond, new capabilities and requirements around flexibility for 6G emerge. This includes the ability of the system to be adapted and tailored to specific use cases and environments. Flexibility as a consequence of disaggregation, softwarisation and automation/orchestration is also seen as an enabler for self-healing and for smoothening the transition from 5G to 6G.

Table 10-1 shows the most relevant KPIs from the M&O perspective, as well as their connection with the KVIs and the Core Capacities mentioned at the beginning of this section.

Table 10-1. KPIs, KVIs and Core Capabilities regarding M&O.

KPI	Description	Related KVIs	Core Capabilities
Latency [s]	M&O can impact in latency by moving resources from one domain to another (e.g., moving resources from the core to the edge, or even to the extreme-	Trustworthiness (Regarding availability for uRLLC services)	Flexibility (Needed to flexible move network resources from one domain to another)

	edge). This may affect both: communication latency (or latency of the communication service as defined in [HX22-D13]) and latency of other services such as AIaaS or Localisation, if the respective functions/compute resources are redistributed.		
Storage Capacity [Bytes]	Refers to the space available to allocate MOs (e.g., NFs, NSs, etc.). The main resources to consider here are the volatile storage (RAM memory) and the non-volatile storage (hard drives, to allocate catalogues, repositories, etc.).	Sustainability (Rel. to usage of Optimised Placement techniques oriented to save energy)	Flexibility (Needed to flexible place network resources on the selected servers) --- Embedded Devices (Part of the capacity can be provided by embedded devices at the extreme-edge)
Processing Capacity [Number & Type of processing units]	Refers to the processing units (PUs) available to execute both: managing and managed objects. The number of each different type of PUs should be considered (e.g., CPUs, GPUs, FPGAs, TPUs…). More than at global level, this KPI would be more relevant granularly, even at Computing Unit level, so the M&O system could apply specific orchestration policies based on the available processing capacity in different domains (e.g., to decide executing certain NFs on the extreme-edge or on other hosts with higher processing capacity, or to execute certain AI/ML-based services on certain nodes with specific processing capabilities, such as those enabled with GPUs or TPUs).	Sustainability (Optimization techniques and dynamic provisioning based on this KPI can impact on the energy usage)	Flexibility (Needed to flexible place network resources on the selected servers or devices) --- Embedded Devices (Part of the capacity can be provided by embedded devices at the extreme-edge)
Programmability [%]	According the SBMA pattern, MOs in the M&O architecture should be programmable through an exposed API. This KPI could be measured as a percentage, indicating the number of devices in the architecture aligned with the SBMA pattern. Note programmable capabilities are	All KVIs (programmability opens the door for implementing different algorithms that can impact in all KVI areas)	Flexibility (Programmability and flexibility are obviously close related concepts)

	anyway not only "binary" (programmable or not), but also relates to the different possibilities of the exposed APIs, that will define the "grade" of programmability and flexibility for each one.		
Energy efficiency [W]	There are two clear ways where M&O can help: – Optimised placement of MOs based on: (i) heuristics or AI/ML algorithms to free-up resources by grouping together MOs in datacentres (Sec. 7.2.1.5), and (ii) by performing placement actions prioritising green energy powered datacentres (adequate profiling options should be available for performing this). – Exploiting elasticity, by automatically connecting/ disconnecting datacentres according to actual customer demand.	Sustainability	Integrated intelligence (To support the AI/ML algorithms) Flexibility (To make possible optimised placement)
Creation time [s]	This refers the time it takes to create the different MOs described in Section 6.1. It is important to remark that, depending on the specific MO which will be created, this KPI measurement may involve different LCM operations i.e., in some cases only instantiation times may be considered (golden-images or pre-configured container environments) but, in other scenarios, configuration and even deletion (of temporal services) might be considered.	-	-
Availability [%]	It is the percentage of time a system is available, i.e., it relates total uptime and total downtime in a period. In telco systems it is typically expressed as a percentage of uptime in a year. Despite there is still no HA defined for 6G, it is expected 6G should outperform the current 5G technology. M&O can contribute to this by increasing	Trustworthiness (Availability is one of the concepts associated to the trustworthiness KVI by itself)	Flexibility (Contributing to higher automation and programmability levels – e.g., automatic rollbacks could be performed in case of failures, contributing to

	programmability and automation levels, as described in Section 7.2.		increase service availability times)
Reliability [%]	Measures how long a system can perform its intended function without interruption, i.e., it measures the failure rate of a system. This impacts on the M&O system itself plus the services running on top of it. This is typically given as the total number of failures divided by the total uptime of the system. M&O can obviously impact on reliability, e.g., by enabling the agile development of bug fixes or new software features. Also, by implementing proactive orchestration behaviours (that could prevent service degradations) by relying on AI/ML approaches.	Trustworthiness (More reliable systems are clearly more trustworthy)	Integrated intelligence (Can be used to implement proactive M&O behaviours)
			Flexibility (This capability is also defined as an enabler for self-healing, which can contribute to increase reliability levels)
AI/ML models training time [s]	Training time could vary from the different AI/ML techniques (see Section 7.2.4.2). Usually the statement "the shorter the training time the better" will prevail, as this can directly impact on multiple KPIs such as availability, service creation time, or others. However, in some scenarios, a longer training-time (optimising the overall resource usage), may be preferred to avoid a negative impact on KVIs such as the Sustainability of the system.	All KVIs can be affected by this KPI, depending on the context AI/ML systems are applied.	Integrated intelligence
Security by design [Boolean]	Security should be considered from the initial stages of the system design, i.e., security strategies, tactics and patterns should be considered at the beginning of the M&O system design, being selected and incorporated as part of the architectural design.	Trustworthiness (Security is explicitly related with this KVI area)	-
Maintainability [Degree – e.g., high, medium, low]	Maintainability refers the ability of a system to be retained in, or restored to, a state in which it can perform as required under given conditions of use and maintenance [HX22-D13]. M&O can contribute to this in	Trustworthiness (A system that facilitates maintainability is also more trustworthy)	Flexibility (Needed to allow software-based maintainability)

	different ways, e.g.: (i) by enabling zero-touch self-healing approaches, (ii) by increasing MOs programmability through open APIs, or (iii) by relying on agile DevOps methodologies to ease software repair.		
Scalability [%]	6G will bring high heterogeneity of devices (home, industrial, automotive, and others) considering not only the core and edge networks, but also the extreme-edge. A quantified target for Hexa-X is to support up to 100 bn. of connected devices in the network, which is an evident challenge for the M&O system.	Inclusiveness (e.g., making possible to integrate end users and/or devices in rural areas)	Flexibility (Based on a cloud-native SBMA realisation it is possible for the system to be scaled to reach specific and disaggregated environments) Usage of Embedded Devices (Those on the extreme-edge)
Elasticity [Degree]	Refers the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible [HSR13] [HKO+16]. The degree (e.g., high-med-low) can be measured considering the amount or resources that can be elastically managed.	Sustainability (Elasticity targets to avoid over- or under-provisioning of resources, having a direct impact on costs and energy savings)	Flexibility (The definition of this Core Capability itself is perfectly aligned to this KPI) Integrated intelligence (Can be used to perform elasticity in a proactive way)
Resiliency [%]	Refers the ability of the network to continue operating correctly during and after a natural or man-made disturbance, such as the loss of mains power [M.2083]. The M&O system can help to improve resilience in different ways (e.g., by automatically migrating MOs to different servers in case of failure in the server they were originally running).	Trustworthiness (A highly resilient system is obviously more trustworthy)	Flexibility (The cloud-native approach can help to increase resilience by itself) Integrated intelligence (Can be used to proactively predict failures)

Automation [Degree]	Automation degree is a relevant KPI for 6G networks. Full network automation is driven by high-level policies and rules without minimal human intervention, with networks being capable of self-configuration, self-monitoring, self-healing, and self-optimisation [HX22-D13]. Lowest automation degree means no automation, i.e., processes should be always full executed by humans.	Trustworthiness Sustainability (A highly automated network is obviously more trustworthy; also more sustainable, since it requires fewer human interactions)	Flexibility (Automation is seen as one of the enablers for this Core Capability, as it is defined) --- Integrated intelligence (AI/ML techniques can support closed-loop automation)
Intent expressiveness [Degree]	This KPI refers to the M&O system's ability to support orchestration actions expressed by means of high-level intent declarations. Highest intent-based configurability degree means intents could be declared using natural language, and that they could affect all the configurable managed objects in the network. Lowest degree would mean the imposition of specific formal languages for declaring intents, that could be applied to a very restricted set of managed objects.	Inclusiveness (Using natural language to perform M&O actions enables people with little technical knowledge to perform configuration actions on deployed NSs. This can help to break technological barriers to integrate new verticals).	Integrated intelligence (AI/ML techniques would be used to perform natural language processing in order to translate high-level intent declarations into low level configuration actions)

11Alignment with the Hexa-X use cases

Deliverable D1.2 [HX21-D12] provides an initial set of use cases connected to the Hexa-X 6G vision. In that deliverable use cases are grouped in different families, namely: sustainable development, massive twinning, tele-presence, robots to cobots, and local trust zones use cases. Naturally, the services M&O architectural design described in this deliverable impacts on all those use cases (and other use cases that could be defined in the future), since the main purpose of M&O processes is to manage and orchestrate any service that could be deployed on the network.

All the use cases described in D1.2 rely on two of the main Hexa-X KVIs: Trustworthiness and Sustainability (see Fig. 4-1 in D1.2). Thereupon, as it can be seen in the previous Section 10 (Table 10-1), most of the M&O-related KPIs are in fact impacting on those two KVIs.

Other values impacting use cases introduced in D1.2 are: Global Coverage, Extreme Experiences, Connecting Intelligence and Network of Networks. The first one could be achieved following the cloud-native deployment model (on which the M&O architecture is based), and the approaches described in the Deployment View (Section 8), which would allow scaling the M&O system to reach even the global scale. For the Extreme Experiences use case (which includes Massive Twining, Telepresence and Robots-to-cobots use cases), the Device-Edge-Cloud continuum M&O functionality (Sec. 7.2.1.1) may have a clear impact, since these use cases require the deployment of services over the distributed network resources, including the devices at the

extreme-edge domain. Besides, in Connecting Intelligence related use cases, the data-driven approach (relying on the AI/ML functions) included in the M&O architectural design will definitely have a direct impact (Sec. 7.2.4), allowing the managing of the complexity associated to these use cases (e.g., regarding to time series processing, proactive management, processing of large amount of heterogeneous data). Finally, regarding the Network of Networks related use cases, the M&O system would address them by relying on the mechanisms that have been envisaged for integrating different networks: the API Management Exposure (Sec. 6.2.3) and the SBMA model on which the M&O architecture is based (see Section 6 introduction).

12 Conclusion

This document has presented the design of service management and orchestration functionalities for 6G networks, addressing the main objectives that were defined for this document (Section 3.1) as stated below for each single objective (in *italic*):

- To serve as the mean for verifying one of the main milestones in the project (MS5), which requires to provide the architectural design of the novel orchestration and management mechanisms for Hexa-X.

  The architectural design has been described by means of three architectural views: the Structural View (describing the main architectural blocks), the Functional View (describing the main functional processes) and the Deployment View (describing how the architecture could be deployed in practice). A specific section has been also included describing how the architecture can align with relevant standards. It is considered this architectural design can be used as a reference for implementations that should meet the requirements and provide the novel capabilities described in Section 5.

- To fulfil the previous considering disruptive trends and technologies, based on the gap analysis performed in the previous Deliverable D6.1 [HX21-D61].

  Regarding this, the main features and enablers identified in the previous Hexa-X WP6 deliverable, D6.1, were used to define the Table of Contents and methodology for elaborating this one. Also, the specific SotA standards analysed in D6.1 have been taken in account to consider possible alignments with the architectural design provided here.

- To describe how the architectural design also includes the necessary means for automation and network programmability of 6G infrastructures.

  The Functional View includes a description of both: programable processes (see Section 7.2.2) and automation processes (see Section 7.2.3).

- To describe also how the M&O architecture can provide also intent-based mechanisms for elaborating on requirements, diagnosing the performance of networks and services, modelling/abstracting services/networks, or implementing corrective actions through CI/CD.

  Intent-based mechanisms are described in Section 7.2.2.1. What the document describes regarding this goes beyond the objective, because the description not only focus on the Design Layer aspects (i.e., elaborating on service requirements, diagnosing performance of networks and services, modelling/abstracting services/networks, or implementing corrective actions through CI/CD mechanisms). The document also explains how to apply intent-based mechanisms to support the verticals at the Service Layer, giving them the possibility to manage their services by using high-level intents.

- To describe how the architectural design can support orchestration of a wide variety of service definitions and decompositions, including (traditional) virtual appliances, microservices and containers, and serverless functions in all domains.

  This objective has been addressed by adopting the cloud-native principles for the architectural design, which in fact offers a design based on the SBMA model.

- To describe how cognitive-based service management and orchestration mechanisms based on optimised placement, resource optimisation, and dynamic allocation would be performed.

  Cognitive-based service management and orchestration mechanisms are provided through a set of specific AI/ML Functions, which are described in Section 6.2.2.1. The functionalities that can be provided by these functions are explained in Section 7.2.4 (data-driven processes). Optimised placement (together with resource optimisation and dynamic allocation) is addressed in Section 7.2.1.4.

- To describe also how data-driven device-edge-cloud continuum management mechanisms would be implemented.

  Data-driven mechanisms are explained in Section 7.2.4, including the monitoring and handling of data, AI-driven orchestration mechanisms (together with the challenges associated to them), and the security-related processes. Device-edge-cloud continuum orchestration mechanisms are specifically addressed in Section 7.2.1.1, considering the M&O on the extreme*-edge infrastructure resources and other external public and private networks.*

- To serve as reference for other work packages in the project, and for the next planned Deliverable D6.3 in this WP6 regarding the final evaluation of the service management and orchestration mechanisms.

  During the editing process of this deliverable, some of the concepts introduced here have already influenced the work of other WPs in the project, namely: in WP1 regarding the E2E architectural design described in Deliverable D1.3 [HX22-D13]; in WP5, regarding the architecture transformation, described in Deliverable D5.1 [HX21-D51]; in WP4, regarding the methods and algorithms for implementing a sustainable and secure distributed AI, described in Deliverable D4.1 [HX21-D41].

  Also, the alignment with certain relevant standards provided in Section 9 and the definition of the main KPIs/KVIs in Section 10 in this report paves the road towards the final evaluation of service management and orchestration mechanisms presented through this document, and that is planned to be reported in the next WP6 Deliverable D6.3.

13 References

[1253a] TM Forum, IG1253A, "Intent Common Model v1.1.0", January 2022.

[23.222] 3GPP TS 23.222, "Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs; Stage 2 (Release 17)", June 2021.

[28.104] 3GPP TS 28.104, "Management Data Analytics (MDA) (Release 17)", March 2022.

[28.533] 3GPP TS 28.533, "Management and Orchestration; Architecture Framework (Release 17)", December 2021.

[28.540] 3GPP TS 28.540, "Management and orchestration; 5G Network Resource Model (NRM); Stage 1", December 2021.

[28.541] 3GPP TS 28.541: "Management and Orchestration; 5G Network Resource Model (NRM); Stage 2 and Stage 3 (Release 17.5.0)", December 2021.

[28.545] 3GPP TS 28.545 5G, "Management and orchestration; Fault Supervision (FS), (Release 16)", November 2020.

[28.550] 3GPP TS 28.550 5G, "Management and orchestration; Performance assurance, (Release 16)", November 2020.

[28.552] 3GPP TS 28.552 5G, "Management and orchestration; 5G performance measurements, (Release 15)", October 2019.

[28.554] 3GPP TS 28.554 5G, "Management and orchestration; 5G end to end Key Performance Indicators (KPI), (Release 15)", October 2019.

[28.622] 3GPP TS 28.622, "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)", August 2020.

[3010] International Telecommunication Union (ITU), Principles for a telecommunications management network, ITU-T Recommendation M.3010. February 2000.

[33.811] 3GPP TR 33.811, "Technical Specification Group Services and System Aspects; Study on security aspects of 5G network slicing management (Release 15)", June 2018.

[5gaiml21] 5G PPP Technology Board, "AI and ML – Enablers for Beyond 5G Networks", Version 1.0, 2021-05-11, DOI 10.5281/zenodo.4299895, online available at: https://5g-ppp.eu/wp-content/uploads/2021/05/AI-MLforNetworks-v1-0.pdf, [Accessed: 2022-04-03].

[5GEVE-D41] 5G EVE D4.1: Experimentation Tools and VNF Repository. [Online] Available at: https://www.5g-eve.eu/wp-content/uploads/2019/11/5g-eve-d4.1-experimentation-tools-and-vnf-repository.pdf [Accessed 11 March 2022]. November 2019.

[5gp21] 5GPPP, "Architecture Working Group. View on 5G Architecture", Version 4.0, August 2021. [Online] Available at: https://5g-ppp.eu/wp-content/uploads/2021/08/Architecture-WP-v4.0_forPublicConsultation.pdf [Accessed 12 April 2022].

[5GPIn19] 5GPPP, 2019. 5G PPP – 5G innovations for verticals with third party services. [Online] CORDIS | European Commission Available at:

https://cordis.europa.eu/programme/id/H2020_ICT-41-2020 [Accessed 12 April 2022].

[5GTRA-D31] 5G Transformer D3.1: “Definition of vertical service descriptors and SO NBI. [Online] Available at: http://5g-transformer.eu/wp-content/uploads/2019/11/D3.1_Definition_of_vertical_service_descriptors_and_SO_NBI-1.pdf [Accessed 10 March 2022]. November 2019.

[5GVIN-D31] 5G VINNI D3.1: Specification of services delivered by each of the 5G-VINNI facilities. [Online] Available at: https://zenodo.org/record/3345612 [Accessed 28 March 2022]. June 2019.

[9595:98] ISO/IEC 9595:1998 "Information technology - Open Systems Interconnection - Common management information service", October 1998.

[9596-1:98] ISO/IEC 9596-1:1998 "Information technology - Open Systems Interconnection — Common management information protocol - Part 1: Specification", October 1998.

[AAU+18] A. Acar, H. Aksu, A. S. Uluagac, M. conti, "A Survey on Homomorphic Encryption Schemes: Theory and Implementation". Association for Computing Machinery, New York, NY, USA, vol. 51, n. 4, 2018.

[ACL05] K. H. Ang, G. Chong, and Y. Li, “PID control system analysis, design, and technology,” IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559–576, 2005.

[AFI15] ETSI TS 103 195-2, “Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 2: An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management”, May 2015.

[Alt20] European Comission. Assessment List for Trustworthy Artificial Intelligence (ALTAI) for self-assessment. 2020. [Online] Available at: https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=68342 [Last Accessed: 16 March 2022].

[AMB20] N. Aydın, İ. Muter, and Ş. İ. Birbil, “Multi-objective temporal bin packing problem: An application in cloud computing,” Computers & Operations Research, vol. 121, 2020.

[Anssi21] French Network and Security Agency (ANSSI/Agence nationale de la sécurité des systèmes d'information): Managing Cybersecurity for Industrial Control Systems, June 2021.

[Ant22] Google, Google Anthos documentations. 2022. [Online] Available at: https://cloud.google.com/anthos [Accessed 14 March 2022].

[AO17] D. B. Abeywickrama and E. Ovaska, ‘A survey of autonomic computing methods in digital service ecosystems’, SOCA, vol. 11, no. 1, pp. 1–31, 2017

[AP21] G. Arbezzano, A. Palesandro, “Simplifying multi-clusters in Kubernetes”. 2021. [Online] Available at: https://www.cncf.io/blog/2021/04/12/simplifying-multi-clusters-in-kubernetes [Last Accessed: 16 March 2022].

[ApM22] Apache Mesos. 2022. Apache Mesos. [Online] Available at: https://mesos.apache.org [Accessed 28 March 2022].

[ATS+18] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, “Network Slicing and Softwarisation: A Survey on Principles, Enabling Technologies, and Solutions,” IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429–2453, 2018.

[BBR+16] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, ‘Survey on Network Virtualization Hypervisors for Software Defined Networking’, IEEE Commun. Surv. Tutorials, vol. 18, no. 1, pp. 655–685, 2016.

[BEG+19] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. ingerman, V. Ivanov, "Towards federated learning at scale: System design." Proceedings of Machine Learning and Systems 1, pp. 374-388. 2019.

[BMR+20] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al., "Language Models are Few-Shot Learners", July 22, 2020, arXiv:2005.14165.

[BPP+19] M. Beshley, A. Pryslupskyi, O. Panchenko and H. Beshley, "SDN/Cloud Solutions for Intent-Based Networking," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), 2019.

[BSM18] F. Bannour, S. Souihi, A. Mellouk, "Distributed SDN Control: Survey, Taxonomy, and Challenges". IEEE Communications Surveys & Tutorials, 20(1), pp. 333–354, 2018.

[BT20] C. Benzaid and T. Taleb, ‘AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions’, IEEE Network, vol. 34, no. 2, pp. 186–194, 2020.

[CAA09] T.B. Callo-Arias, P. America and P. Avgeriou, “Defining execution viewpoints for a large and complex software-intensive system”, Proceedings of WICSA/ECSA, 2009.

[CB21] E. Calvanese Strinati, S. Barbarossa, “6G networks: Beyond Shannon towards semantic and goal-oriented communications”, Computer Networks, Vol. 190, 2021.

[CBB+10] P. Clements, F. Bachmann, L. Bass, D. Garlan et al., “Documenting Software Architectures: Views and Beyond”, Addison-Wesley Professional; Ed: 2 (15 oct. 2010), ISBN-10: 0321552687, ISBN-13: 978-0321552686.

[CCG+21] A. Clemm, L. Ciavaglia, L. Granville, J. Tantsura , “Intent-Based Networking - Concepts and Definitions”, IETF Draft, December 2021

[CEC+21] Coffman Jr., Edward G.; Csirik, János; Galambos, Gábor; Martello, Silvano; Vigo, Daniele (2013), Pardalos, Panos M.; Du, Ding-Zhu; Graham, Ronald L. (eds.), "Bin Packing Approximation Algorithms: Survey and Classification", Handbook of Combinatorial Optimization, New York, NY: Springer, pp. 455–531, 2021.

[Cis20] Centre of Internet Security (CIS). 2020. Policy Template Guide v1.1. [Online] Available at: https://www.cisecurity.org/wp-content/uploads/2020/07/NIST-CSF-Policy-Template-Guide-2020-0720-1.pdf [Accessed 14 March 2022].

[CJP+07] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown, Scott Shenker, “Ethane: taking control of the enterprise”, SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pp.1-12, 2007.

[Clo21] Cloud Native Computing Foundation. 2021. Cloud Native Computing Foundation. [online] Available at: https://www.cncf.io [Accessed 21 October 2021].

[CN06] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in Proceedings of the International Conference on Machine Learning, pp. 161–168, 2006.

[CPL13] S.-G. Cui, H.-L. Pan, and J.-G. Li, “Application of self-tuning of PID control based on BP neural networks in the mobile robot target tracking,” in Proceedings of the International Conference on Instrumentation, Measurement, Computer, Communication and Control, pp. 1574–1577, 2013.

[CXC+19] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, Z. M. Mao, "Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving." In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. Association for Computing Machinery, New York, NY, USA, pp. 2267–2281, 2019.

[DK15] Dhavare, U., Kulkarni, U. (2015). Natural language processing using artificial intelligence. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 4(2), 203-205.

[DLH19] Y. Dang, Q. Lin, and P. Huang, “Aiops: real-world challenges and research innovations,” 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), IEEE, pp. 4-5, 2019.

[DMS+22] D. Kim, M. Ko, S. Kim, S. Moon, K. Cheon, S. Park, "Design and Implementation of Traffic Generation Model and Spectrum Requirement Calculator for Private 5G Network". IEEE Access, vol. 10, pp. 15978-15993, 2022.

[DoS22] Docker Documentation. 2022. Swarm mode overview. [Online] Available at: https://docs.docker.com/engine/swarm [Accessed 28 March 2022].

[DSB17] D. Doran, S. Shulz, TR. Besold, "What does explainable AI really mean? A new conceptualization of perspectives". arXiv preprint arXiv:1710.00794, 2017.

[EEF+17] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prokash, A. Rahmati, D. Song, "Robust Physical-World Attacks on Machine Learning Models". CoRR, vol. abs/1707.08945, 2017.

[EGH+16] C. Ebert, G. Gallardo, J. Hernantes and N. Serrano, "DevOps", IEEE Software, vol. 33, no. 3, pp. 94-100, 2016.

[Eni003] ETSI GS ENI 003, “Experiential Networked Intelligence (ENI); ENI Use Cases”, December 2020.

[Eni004] ETSI GR ENI 004, “Experiential Networked Intelligence (ENI); Terminology for Main Concepts in ENI”, December 2021.

[Eni005] ETSI GS ENI 005, “Experiential Networked Intelligence (ENI); System Architecture”, August 2021.

[Eni018] ETSI GR ENI 018, “Experiential Networked Intelligence (ENI); Introduction to Artificial Intelligence Mechanisms for Modular Systems”, December 2021.

[FB97] M. A. Friedl and C. E. Brodley, “Decision tree classification of land cover from remotely sensed data,” Remote Sensing of Environment, vol. 61, no. 3, pp. 399–409, 1997.

[Fib17] Wikipedia Commons. Fibonacci, Kanizsa triangle. 2017. [Online] Available at: https://en.wikipedia.org/wiki/File:Kanizsa_triangle.svg/ [Last Accessed: 04 April 2022].

[FPE+17] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network Slicing in 5G: Survey and Challenges,” IEEE Communications Magazine, vol. 55, no. 5, pp. 94–100, 2017.

[GPM+14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., Generative Adversarial Nets, Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.

[GAD+15] P. G. Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere 'Edge-centric Computing: Vision and Challenges', SIGCOMM Comput. Commun. Rev., vol. 45, no. 5, pp. 37–42, 2015.

[GANA] ETSI Whitepaper No. 16, "GANA - Generic Autonomic Networking Architecture Reference. Model for Autonomic Networking, Cognitive Networking and Self-Management of Networks and Services". October 2016. [Online] Available at: https://www.etsi.org/images/files/etsiwhitepapers/etsi_wp16_gana_ed1_20161011.pdf [Accessed 03 March 2022].

[GB921] TM Forum GB921, "Business Process Framework (eTOM) R17.0.1", June 2017.

[GB999] TM Forum "GB999 ODA Production Implementation Guidelines", 2020.

[GC21] A. Garcia-Saavedra and X. Costa-Pérez, "O-RAN: Disrupting the Virtualized RAN Ecosystem," in IEEE Communications Standards Magazine, vol. 5, no. 4, pp. 96-103, 2021.

[GKM18] A. Grange, I. Kacem, and S. Martin, "Algorithms for the bin packing problem with overlapping items," Computers & Industrial Engineering, vol. 115, pp. 331-341, 2018.

[GMB+19] A. Ghosh, A. Mäder, M. Baker, C. Devaki, "5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15.", IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2939938, 2019.

[GoC22] Google Cloud. 2022. Hybrid Cloud Management with Anthos, Google Cloud. [Online] Available at: https://cloud.google.com/anthos [Accessed 28 March 2022].

[GoT19] Google Books Ngram Viewer. 2019. Google Books Ngram Viewer: Container Orchestrators Comparison. [online] Available at: https://books.google.com/ngrams/graph?content=Kubernetes%2CApache+Mesos%2CDocker+Swarm%2COpenShift&year_start=2012&year_end=2019&corpus=26&smoothing=0 [Accessed 28 March 2022].

[GoT22] Google Trends. 2022. Google Trends: Container Orchestrators Comparison. [online] Available at: https://trends.google.com/trends/explore?date=2014-01-01%202022-03-26&q=kubernetes,docker%20swarm,apache%20mesos,%2Fm%2F0j9pqc0 [Accessed 28 March 2022].

[GPM+20] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan and M. Zorzi, "Toward 6G Networks: Use Cases and Technologies," in IEEE Communications Magazine, vol. 58, no. 3, pp. 55-61, March 2020, doi: 10.1109/MCOM.001.1900411.

[GSM21] P. Gohel, P. Singh, M. Mohanty, "Explainable AI: current status and future directions", arXiv:2107.07045v1, 2021.

[GSR+21] J. Gallego-Madrid, R. Sanchez-Iborra, P. M. Ruiz, and A. F. Skarmeta, 'Machine learning-based zero-touch network and service management: A survey', Digital Communications and Networks, 2021.

[Gun18] J. Günter, "Machine intelligence for adaptable closed loop and open loop production engineering systems", 2018.

[Hel22] Helm Charts Docs. 2022. Helm Charts. [online] Available at: https://helm.sh/docs/topics/charts [Accessed 09 March 2022].

[HKO+16] N. Herbst, R. Krebs, G. Oikonomou, G. Kousiouris, A. Evangelinou, A. Iosup, S. Kounev, "Ready for Rain? A View from SPEC Research on the Future of Cloud Metrics". Technical Report SPEC-RG-2016-01, SPEC Research Group - Cloud Working Group, Standard Performance Evaluation Corporation (SPEC), 2016.

[HM08] M. C. Huebscher and J. A. McCann, 'A survey of autonomic computing—degrees, models, and applications', ACM Comput. Surv., vol. 40, no. 3, pp. 1–28, 2008.

[HSK13] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. 438–445, 2013.

[HSR13] N. Herbst, K. Samuel, R. Ralf, "Elasticity in Cloud Computing: What It Is, and What It Is Not". Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24–28, 2013.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning". Springer, 2009.

[HX21-D12] Hexa-X Deliverable D1.2: Expanded 6G vision, use cases and societal values – including aspects of sustainability, security and spectrum. April 2021.

[HX21-D41] Hexa-X Deliverable D4.1: AI-driven communication & computation co-design: Gap analysis and blueprint. August 2021.

[HX21-D51] Hexa-X Deliverable D5.1: Deliverable D5.1Initial 6G Architectural Components and Enablers. December 2021.

[HX21-D61] Hexa-X Deliverable D6.1: Report on identified gaps, features and enablers for service management and orchestration. June 2021.

[HX22-D13] Hexa-X Deliverable D1.3: Targets and requirements for 6G -initial E2E architecture. February 2022.

[HX22-D42] Hexa-X Deliverable D4.2: AI-driven communication & computation co-design: initial solutions. Ongoing.

[Ibm01] Horn, Petr Jan. "Autonomic Computing: IBM's Perspective on the State of Information Technology.", 2001.

[Ibm05] IBM whitepaper, "An architectural blueprint for autonomic computing", 2005.

[Ifa005] ETSI GS NFV-IFA 005, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification", August 2019.

[Ifa006] ETSI GS NFV-IFA 006, "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification", August 2017.

[Ifa007] ETSI GS NFV-IFA 007, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification", September 2019.

[Ifa008] ETSI GS NFV-IFA 008, "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification", May 2021.

[Ifa013] ETSI GS NFV-IFA 013, “Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification”, April 2019.

[Ifa026] ETSI GS NFV-IFA 029, “NFV Management and Orchestration; Architecture enhancement for Security Management Specification”, November 2019.

[Ifa029] ETSI GR NFV-IFA 029, “NFV Architecture; Report on the Enhancement of the NFV architecture towards ‘Cloud-native’ and ‘PaaS’”, November 2019.

[Ifa033] ETSI GS NFV-IFA 033, “NFV Management and Orchestration; Sc-Or, Sc-Vnfm, Sc-Vi reference points - Interface and Information Model Specification”, Aug 2020.

[ITM+21] R. Inam, A. Terra, A. Mujumdar, E. Fersman, “Explainable AI – how humans can trust AI,” Ericsson White Paper GFTL-21:000529Uen, 2021.

[Jai08] A. K. Jain, “Data Clustering: 50 Years Beyond K-means”, Springer Berlin Heidelberg in Machine Learning and Knowledge Discovery in Databases pp. 3-4, 2008.

[Jar18] M.H Jarrahi, "Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making". Business Horizons, Vol. 61, Issue 4, pp. 577-586, 2018.

[JAS02] G. Joya, M. A. Atencia, and F. Sandoval, “Hopfield neural networks for optimization: study of the different dynamics,” Neurocomputing vol. 43, no. 1-4, pp. 219-237, 2002.

[JHM21] A. Jeffery, H. Howard, R. Mortier, "Rearchitecting Kubernetes for the Edge". Association for Computing Machinery, New York, NY, USA, pp. 7-12, 2021.

[Jin19] Jing Ping, "Network Resource Model for 5G Network and Network Slice", Journal of ICT Standardization, Vol: 7, Issue: 2, May 2019, Article No: 4, Page: 127-140, doi: https://doi.org/10.13052/jicts2245-800X.724

[Jol02] I. Jolliffe, “Principal component analysis”. Wiley Online Library, 2002.

[K3s22] K3s: Lightweight Kubernetes. 2022. [Online] Available at: https://k3s.io/ [Accessed 17 March 2022].

[K8Co22] Kubernetes. 2022. K8s Docs: Components. [Online] Available at: https://kubernetes.io/docs/concepts/overview/components [Accessed: 28 March 2022].

[K8s22] Kubernetes. 2022. K8s: Production-Grade Container Orchestration. [Online] Available at: https://kubernetes.io [Accessed: 28 March 2022].

[K8Wh22] Kubernetes. 2022. K8s Docs: What is Kubernetes. [Online] Available at: https://kubernetes.io/docs/concepts/overview/what-is-kubernetes [Accessed: 28 March 2022].

[KA20] K. Manaouil, A. Lebre, “Kubernetes and the Edge?”. RR-9370, Inria Rennes - Bretagne Atlantique, pp.19, 2020.

[KAR19] Kalyanathaya, K. P., Akila, D., & Rajesh, P. (2019). Advances in natural language processing–a survey of current research trends, development tools and industry applications. International Journal of Recent Technology and Engineering, 7(5C), 199-202.

[KL90] A. Khotanzad and J.-H. Lu, “Classification of invariant image representations using a neural network,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 6, pp. 1028–1038, 1990.

[KMB+19] Kairouz, P., McMahan H.B., Brendan A., Aurélien B.; Mehdi B. et al. "Advances and Open Problems in Federated Learning". arXiv:1912.04977. 10 December 2019.

[KMR15] Konečný J., McMahan B., Ramage D., "Federated Optimization: Distributed Optimization Beyond the Datacenter". (2015) arXiv:1511.03575.

[Koh88] T. Kohonen., "The 'Neural' Phonetic Typewriter". Computer 21, 3 (March 1988), 11–22, 1988.

[KP21] S. Kmkov, A. Petiushko, "AdvHat: Real-World Adversarial Attack on ArcFace Face ID System". 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021.

[Kru95] P. Kruchten. "Architectural Blueprints—The 4+1 View Model of Software Architecture". Paper published in IEEE Software 12 (6) November 1995, pp. 42-50.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", 2012.

[KuE22] KubeEdge: Kubernetes Native Edge Computing Framework. 2022. [Online] Available at: https://kubeedge.io/en [Accessed 17 March 2022].

[KuF22] KubeFed: Kubernetes Cluster Federation. 2022. [Online] Available at: https://github.com/kubernetes-sigs/kubefed [Accessed 14 March 2022].

[LBM20] C. A. Lee, R. B. Bohn, M. Michel, 'The NIST Cloud Federation Reference Architecture', National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 500-332, 2020.

[LCZ+19] Q. Long, Y. Chen, H. Zhang, X. Lei, "Software Defined 5G and 6G Networks: a Survey", Mobile Netw Appl, 2019.

[LIM21] B. Lim, S. Zohren, "Time-series forecasting with deep learning: a survey," Philosophical Transactions of the Royal Society A, 379, no. 2194, 2021. Available online at: https://www.oxford-man.ox.ac.uk/wp-content/uploads/2020/11/Time-Series-Forecasting-With-Deep-Learning_-A-Survey.pdf

[LLC+21] A. Llorens-Carrodeguas, I. Leyva-Pupo, C. Cervelló-Pastor, L. Piñeiro, S. Siddiqui, WAn SDN-Based Solution for Horizontal Auto-Scaling and Load Balancing of Transparent VNF Clusters". Sensors, 202.

[LLF+21] Z. Lv, R. Lou, H. Feng, et al., "Novel machine learning for big data analytics in intelligent support information management systems," ACM Transactions on Management Information System (TMIS), vol. 13, no. 1, pp. 1-21, 2021.

[LLZ+21] S. K. Lo, Q. Lu, L. Zhu, H. Y. Paik, X. Xu, c. Wang, "Architectural patterns for the design of federated learning systems." arXiv preprint arXiv:2101.02373. 2021.

[LLZ10] F. Lin, S. Lin, and S. Zeng, "Co-simulation of neural networks PID control for ship steering hydraulic system", Proceedings of the International Conference on Information and Automation, pp. 2097–2100, 2010.

[M.2083] ITU-R M.2083-0, "IMT Vision - ITU. IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond", September 2015.

[Man001] ETSI GS NFV-MAN 001, "Network Functions Virtualization (NFV); Management and Orchestration", December 2014.

[Max68] J.C. Maxwell, "On Governors". Proceedings of the Royal Society of London. 16: 270–283, 1868.

[MB16] M. B. Mamoun, R. Benaini, "An Overview on SDN Architectures with Multiple Controllers". Journal of Computer Networks and Communications, Hindawi Publishing Corporation, 2016.

[Mec003] ETSI GS MEC 003, “Multi-Access Edge Computing (MEC); Framework and Reference Architecture.”, December 2020.

[Mec031] ETSI GS MEC 031, “Multi-Access Edge Computing (MEC); MEC 5G Integration”, October 2020.

[Mec035] ETSI GR MEC 035, “Multi-Access Edge Computing (MEC); Study on Inter-MEC systems and MEC-Cloud systems coordination”, June 2021.

[Met14] MetaSwitch Networks White paper, “A Guide to NFV and SDN”. [Online] Available at: https://info.metaswitch.com/hs-fs/hub/415294/file-2505370972.pdf [Accessed: 17 March 2022].

[MGZ+19] B. Mutaz, S. Garg, A. Y. Zomaya, L. Wang, A. Moorsel, R. Ranjan, “Orchestrating Big Data Analysis Workflows in the Cloud: Research Challenges, Survey, and Future Directions.” ACM Computing Surveys, 1–37, 2019.

[MH19] Masood, A., Hashmi, A., (2019), "AIOps: Predictive Analytics & Machine Learning in Operations", Cognitive Computing Recipes: Artificial Intelligence Solutions Using Microsoft Cognitive Services and TensorFlow, Apress, pp. 359–382, doi:10.1007/978-1-4842-4106-6_7, ISBN 978-1-4842-4106-6

[Mil14] D. Milham, “ZOOM User Stories”, TR229, v14.4.0, October 23, 2014.

[Mit96] M. Mitchell, “An Introduction to Genetic Algorithms”, 1996.

[MJG19] F. D. Muñoz-Escoí, R. de Juan-Marín, J. García-Escrivá, J. R. González de Mendívil and J. M. Bernabéu-Aubán, "CAP Theorem: Revision of Its Related Consistency Models," The Computer Journal, vol. 62, no. 6, June 2019.

[MKK12] B. V. Murthy, Y. P. Kumar, and U. R. Kumari, “Application of neural networks in process control: automatic/online tuning of pid controller gains for±10% disturbance rejection,” in Proceedings of the International Conference on Advanced Communication Control and Computing Technologies, pp. 348–352, IEEE, 2012.

[MKS+15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.

[MKs22] MicroK8s: Low-ops, minimal production Kubernetes, for devs, cloud, clusters, workstations, Edge and IoT. 2022. [Online] Available at: https://microk8s.io [Accessed 14 March 2022].

[ML20] K. Manaouil, A. Lebre, “Kubernetes and the Edge?” RR-9370, Inria Rennes - Bretagne Atlantique., pp.19, 2020.

[MMR+17] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data”, arXiv:1602.05629, 2017.

[MRP11] M. Sindelar, R. Sitaraman, P. Shenoy, "Sharing-Aware Algorithms for Virtual Machine Colocation". Proceedings of 23rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), p.p. 367–378, 2011.

[Ng116] GSM Association, Official Document NG.116, “Generic Network Slice Template”, June 2021.

[NGPaaS-D32] Next Generation Platform as a Service (NGPaaS) Deliverable D3.2: Final Dev-For-Operations Model specification. [Online] Available at: http://ngpaas.eu/projectoutcomes/#ngpaasdeliverables [Accessed 10 March 2022 - Status "Under validation by EC"]. May 2019.

[Nist18] National Institute of standards and Technology (NIST), "Framework for Improving Critical Infrastructure Cybersecurity", April 2018.

[Nok20] Nokia Bell Labs, "The 3GPP-defined Service Based Management Architecture", Technical brief, Document code: SR2007045991EN (July) CID207723, (2020), [Online] Available at: https://nokianews.net/files/Nokia_Bell_Labs_The_3GPP-defined_Service_Based_Management_Architecture_White_Paper_EN.pdf

[ONAP] ONAP. [online] Available at: https://www.onap.org [Accessed 12 April 2022].

[OnDev22] The Linux foundation Open Network Automation Platform, ONAP. 2022. ONAP: DevOps - integration with CI/CD pipelines. [online] Available at: https://wiki.lfnetworking.org/download/attachments/50528946/ONAP%20And%20DevOps%20-%20integration%20with%20CI-CD%20pipelines.pdf?version=1&modificationDate=1612389561036&api=v2 [Accessed 10 March 2022].

[OpS22] Hybrid Cloud, Red Hat. 2022. What is OpenShift? - Red Hat OpenShift. [Online] Cloud.redhat.com. Available at: https://cloud.redhat.com/learn/what-is-openshift [Accessed 28 March 2022].

[OrA21] O-RAN, “O-RAN Architecture Description; WG1: Use Cases and Overall Architecture Workgroup”, July 2021.

[OrC21] O-RAN, “O-RAN Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN; WG6: The Cloudification and Orchestration Workgroup”, July 2021.

[OrN21] O-RAN, “O-RAN Non-RT RIC Architecture; WG2: The Non-Real-Time RAN Intelligent Controller and A1 Interface Workgroup”, July 2021.

[OrNe21] O-RAN, “O-RAN Near-Real-time RAN Intelligent Controller Architecture & E2 General Aspects and Principles; WG3: The Near-real-time RIC and E2 Interface Workgroup”, July 2021.

[OrS21] O-RAN, “O-RAN Slicing Architecture; WG1: Use Cases and Overall Architecture Workgroup”, July 2021.

[Osm22] ETSI Open source MANO. 2022. OSM. [online] Available at: https://osm.etsi.org [Accessed 09 March 2022].

[PDS13] P. Pilarski, T. Dick, and R. Sutton, “Real-time prediction learning for the simultaneous actuation of multiple prosthetic joints,” in Proceedings of the International Conference on Rehabilitation Robotics, pp. 1–8, 2013.

[Pi22] Raspberry Pi Foundation. 2022. Teach, Learn, and Make with Raspberry Pi. [Online] Available at: https://www.raspberrypi.org [Accessed 28 March 2022].

[PiM22] Ubuntu. 2022. How to build a Raspberry Pi Kubernetes cluster using MicroK8s. [online] Available at: https://ubuntu.com/tutorials/how-to-kubernetes-cluster-on-raspberry-pi?&_ga=2.256596491.1495567704.1648324976-491072784.1648324976#1-overview [Accessed 28 March 2022].

[Pin19] 3GPP SA5, J. Ping, "Network Resource Model for 5G Network and Network Slice", February 2019.

[PJP+16] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares and H. S. Mamede, "Machine Learning in Software Defined Networks: Data collection and traffic classification," 2016 IEEE 24th International Conference on Network Protocols (ICNP), 2016.

[RaF22] Rancher Fleet. 2022. [Online] Available at: https://fleet.rancher.io [Accessed 29 March 2022].

[RGH+86] D. Rumelhart, E. Geoffrey, E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation", 1986.

[RHA21] Red Hat, Red Hat Advanced Cluster Management for Kubernetes. [Online]. Available at: <https://www.redhat.com/en/technologies/management/advanced-cluster-management> [Accessed 14 March 2022].

[RKL13] R. Robin, W. ken, T. Lana, "New Zealand Security Incident Management Guide for Computer Security Incident Response Teams (CSIRTs)", New Zealand National Cyber Security Centre, 2013.

[RLM18] R. Roman, J. Lopez, M. Mambo, 'Mobile Edge Computing, Fog et al.: A Survey and Analysis of Security Threats and Challenges', Future Generation Computer Systems, vol. 78, pp. 680–698, 2018.

[Roj21] Rojas, G. "NIST Cybersecurity Framework and Kubernetes". [Online] Available at: https://eng.lifion.com/nist-cybersecurity-framework-and-kubernetes-770d3df84d6c [Accessed 14 March 2022].

[RRB+22] D. Roeland, K. Raizer, V. Berggren, P. Öhlén, N. Linder, "Cognitive networks – towards an end-to-end 6G architecture", Ericsson blog. [online] Available at: https://www.ericsson.com/en/blog/2022/1/cognitive-networks-6g-architecture [Accessed 24 March 2022].

[SA13] S. Ramadass, A. Abraham, "Comparison of supervised and unsupervised learning algorithms for pattern classification." International Journal of Advanced Research in Artificial Intelligence 2.2, pp. 34-38. 2013.

[SA94] S. Schaal and C. G. Atkeson, "Robot juggling: implementation of memory-based learning," IEEE Control Systems, vol. 14, no. 1, pp. 57–71, 1994.

[SAL07] J. Strassner, N. Agoulmine, E. Lehtihet, "FOCALE – A Novel Autonomic Networking Architecture", International Transactions on Systems, Science, and Applications (ITSSA) Journal, Vol. 3, No 1, pp 64-79, 2007.

[SB11] S. Sastry and M. Bodson, "Adaptive control: stability, convergence and robustness". Courier Corporation, 2011.

[SB18] R. S. Sutton, A. G. Barto, "Reinforcement learning: An introduction". MIT press, 2018.

[SB98] R. S. Sutton and A. G. Barto, "Introduction to Reinforcement Learning". Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[Sec013] ETSI GS NFV-SEC 013, "NFV Security; Security Management and Monitoring specification", February 2017.

[Sec024] ETSI GS NFV-SEC 024, "Network Functions Virtualization (NFV); Security; Security Management (Release 4)", April 2021.

[SEJ15] B. K. Samanthula, Y. Elmehdwi, and W. Jiang, "K-nearest neighbor classification over semantically secure encrypted relational data," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 5, pp. 1261–1273, 2015.

[Sel19] Select committee on artificial intelligence of the national science and security council, “The National Artificial Intelligence Research and Development Strategic Plan: 2019 Update”, 2019.

[SMK+20] Salih Sevgican, Meriç Turan, Kerim Gökarslan, H. Birkan Yilmaz, and Tuna Tugcu, "Intelligent Network Data Analytics Function in 5G Cellular Networks using Machine Learning", Journal of Communications and Networks, ISSN: 1976-5541, Vol. 22, No. 3 (June 2020), pp. 269 - 280, doi: 10.1109/JCN.2020.000019

[SNH+21] Y. M. Saputra, D. N. Nguyen, D. T. Hoang, Q. Pham, E. Dutkiewicz, W. Hwang, "Federated Learning Framework with Straggling Mitigation and Privacy-Awareness for AI-based Mobile Application Services". FOS: Computer and information sciences, arXiv, 2021.

[SPB+18] J. C. Sercel, C. E. Peterson, D. T. Britt, C. Dreyer, R. Jedicke, S. G. Love, O. Walton, “Chapter 9 - Practical Applications of Asteroidal ISRU in Support of Human Exploration”, Elsevier, 2018.

[SSK20] S. K. Singh, R. Singh, B. Kumbhani, "The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities". 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 1-6, 2020.

[Sze10] C. Szepesvári, "Algorithms for reinforcement learning. Synthesis lectures on artificial intelligence and machine learning", vol. 4, no 1, pp. 1-103. 2010.

[SZF+18] T. Szigeti, D. Zacks, M. Falkner, et al., “Cisco Digital Network Architecture: Intent-based Networking for the Enterprise,” Cisco Press, 2018.

[SZI21] P. Szilágyi, “I2bn: Intelligent intent-based networks,” Journal of ICT Standardization, pp. 159-200, 2021.

[SZV+19] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, “Resource Allocation for Network Slicing in 5G Telecommunication Networks: A Survey of Principles and Models,” IEEE Network, vol. 33, no. 6, pp. 172–179, 2019.

[TCP91] G. A. Tagliarini, J. F., Christ, and E. W. Page, “Optimization using neural networks,” IEEE transactions on computers, vol. 40, no. 12, pp. 1347-1358, 1991.

[TiOc21] TIP Open Core, “Private 5G Scenarios; Open Core Network Project Group; Applications and Services Subgroup,” v1.0.0, November 2021.

[TiRa22] TIP OpenRAN, “Automation and Optimisation for OpenRAN”, January 2022.

[TiT21] TIP Open Transportation, “Use Cases and Technical Requirements for Wireless Backhaul SDN Domain Controller & Network Equipment,” v1.0.0, November 2021.

[TK01] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” Journal of machine learning research, vol. 2, no. Nov, pp. 45–66, 2001.

[Tlf22] Linux Foundation. 2022. Linux Foundation - Decentralized innovation, built with trust. [Online] Available at: https://www.linuxfoundation.org [Accessed: 28 March 2022].

[TmO22] TM Forum ODA, 2022. Open Digital Architecture. [online] TM Forum. Available at: <https://www.tmforum.org/resources/whitepapers/open-digital-architecture/> [Accessed 6 April 2022].

[TmZ22] TM Forum ZOOM, 2022. Zero-Touch Orchestration, Operations & Management (ZOOM). [online] Tmforum.org. Available at: <https://www.tmforum.org/wp-content/uploads/2016/04/ZOOM_Poster.pdf> [Accessed 6 April 2022].

[TRA15] A. Tosatto, P. Ruiu, A. Attanasio, "Container-Based Orchestration in Cloud: State of the Art and Challenges," 2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 70-75, 2015.

[Tur36] A. Turing, "On computable numbers, with an application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42 (1936), pp 230 - 265. [Online] Available at: https://turingarchive.kings.cam.ac.uk/publications-lectures-and-talks-amtb/amt-b-12 [Last Accessed: 06 March 2022].

[TVS10] J-A. Ting, S. Vijayakumar, S. Schaal, “Encyclopedia of Machine Learning”, Sammut C, Webb GI. Boston, MA: Springer US. p. 613-624, 2010.

[VARYS] Next Generation Platform as a Service (NGPaaS). 2019. VARYS: Multi-tier Technology-agnostic Monitoring as a Service solution for Cloud systems. [Online] Available at: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/horizon-results-platform/13777;isExactMatch=false;advancedFilters=1;projectId=761557 [Accessed 30 March 2022].

[VIT5G-D21] VITAL 5G D2.1: Initial NetApps blueprints and Open Repository design. [online] Available at: https://www.vital5g.eu/wp-content/uploads/2022/01/VITAL5G_D2.1_Initial_NetApps_blueprints_and_Open_Repository_design_Final.pdf [Accessed 10 March 2022]. January 2022.

[VLL+10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” Journal of Machine Learning Research, vol. 11, pp. 3371–3408, 2010.

[WCW+17] M. Wang, Y. Cui, X. Wang, S. Xiao, J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities", 2017.

[WD92] C. Watkins, P. Datan, "Q-learning. Machine learning", vol. 8, no 3, pp. 279-292. 1992.

[WRS+20] C. X. Wang, M. Di Renzo, S. Stańczak, S. Wang, and E. G. Larsson, “Artificial Intelligence Enabled Wireless Networking for 5G and Beyond: Recent Advances and Future Challenges”, 2020

[WS88] P.D. Wasserman, T.Schwartz, “Neural networks. II. What are they and why is everybody so interested in them now?”; 1988.

[YLC+19] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, H. Yu, “Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning”, 13(3), 1-207, 2019.

[YYY+19] Q. Yang, Y. Liu, Y. Cheng, et al., “Federated learning. Synthesis Lectures on Artificial Intelligence and Machine Learning,” vol. 13, no. 3, pp. 1-207, 2019.

[Zha19] S. Zhang, “An Overview of Network Slicing for 5G,” IEEE Wireless Communications, vol. 26, no. 3, pp. 111–117, 2019.

[ZKJ+08] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, I. Stoica, “Improving MapReduce performance in heterogeneous environments,” Proc. OSDI, vol. 8, no. 4, p. 7, Dec. 2008.

[zsm-002] ETSI GS ZSM 002, “Zero-touch network and Service Management; Reference Architecture”, August 2019.

[zsm-013] ETSI ZSM Work Programme - Work Item Detailed Report. (2022). Retrieved 26 April 2022, from https://portal.etsi.org/webapp/WorkProgram/Report_ WorkItem.asp? WKI_ID =61979

[ZT20] E. Zeydan and Y. Turk, "Recent Advances in Intent-Based Networking: A Survey," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020.

END OF DOCUMENT