Integrating Network Digital Twinning into Future AI-based 6G Systems

D1.3

Frameworks for zero-touch service, network management, and the orchestration of its AI-based NF and NS (initial)

Document Information	
Grant Agreement N°	101136314
Authors and institutions	Paola Soto (IMEC, main editor); Julian Jimenez (IMEC), Miguel Camelo (IMEC), Luis Santos (UBI), Ayat Zaki Hindi (LIST), Sebastien Faye (LIST), German Castellanos (ACC), Ayse Sayin (EBY), Gaetano Volpe (POLIBA), Wasim Ali (POLIBA), Pritom Touhid (TUD), Sidi-Mohammed Senouci (UBOU), Shajjad Hossain (UBOU)
Date	30 June 2025
Related WP	WP1 | 6G Architecture Design
Dissemination level	PU | Public, fully open

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Document change history

Version	Date	Author	Description
V0.1	02/05/2025	Paola Soto (IMEC)	Initial table of content and timeline proposed.
V0.3	15/05/2025	All authors	Contributions in bullet points form.
V0.4	30/05/205	All authors	Finalized Section 2 content
V0.5	15/06/2025	IMEC	Finalized Section 4
V0.6	20/06/2025	UBI, IMEC, ACC, LIST	Initial draft Section 3 and Section 5.
V0.7	24/06/2025	LIST and TUD	First peer review
V0.8	27/06/2025	All authors	Pre-final version after peer review
V0.9	30/06/2025	Paola Soto (IMEC)	Document clean up
V1	30/06/2025	Sebastien Faye (LIST)	Final version submitted

Quality control

	Name	Organization	Date
Editor:	Paola Soto	IMEC	30/06/2025
Peer review 1:	Sebastien Faye	LIST	24/06/2025
Peer review 2:	Burkhard Hensel	TUD	24/06/2025
Authorized and submitted by (Project Coordinator):	Sebastien Faye	LIST	30/06/2025

Legal disclaimer

Executive Summary

Deliverable D1.3 marks the final contribution of Work Package 1 (WP1) in defining the functional architecture for the 6G-TWIN project, with a specific focus on the Management and Orchestration (MANO) layer. The overarching goal of 6G-TWIN is to establish an AI-native architectural framework for 6G networks, in which Network Digital Twins (NDTs) serve as central mechanisms for real-time, closed-loop optimization, management, and control. This deliverable advances previous work (D1.1 and D1.2) by refining the integration of NDTs within the network control stack and elaborating on their Lifecycle Management (LCM), orchestration, and federated operation.

The document introduces the 6G-TWIN MANO layer, a vertical coordination layer that supervises NDT instances' deployment, operation, and automation and subcomponents across diverse domains. It details the internal composition of NDT instances, distinguishing between basic and functional models, and describes workflows for automated model updates in response to network dynamics. Furthermore, the report aligns the architecture with Zero-Touch Service Management (ZSM) principles, supporting AI-native orchestration, intent-based control, and secure, scalable operations across federated environments.

This deliverable also presents early practical demonstrations validating some key architecture concepts, including telemetry-based User Equipment (UE) localization, radio coverage prediction using Machine Learning (ML) models, and automated functional model retraining. These contributions provide a robust reference for integrating NDTs into AI-driven, real-world 6G network deployments, bridging theoretical design and implementation.

This document is presented in its initial version, reflecting the work carried out during the first half of 6G-TWIN (i.e., up to Month 18). A final, updated version will be produced at the end of the project and published as Deliverable D1.6.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

6GSNS

Abbreviations and acronyms

Abbreviation	Definitiion
3GPP	Third Generation Partnership Project
5G	Fifth Generation
6G	Sixth Generation
AI	Artificial Intelligence
ANOVA	Analysis of Variance
API	Application Programming Interface
BER	Bit Error Rate
CLA	Closed-Loop Automation
CN	Core Network
CNF	Cloud-native Network Function
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CQI	Channel Quality Indicator
CSMF	Communication Service Management Function
CU	Central Unit
DL	Deep Learning
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DT	Digital Twin
DTR	Decision Tree Regressor
DU	Distributed Unit
E2E	End-to-End
ETSI	European Telecommunications Standards Institute
EUCNC	European Conference on Networks and Communications
FDD	Frequency Division Duplex
FL	Federated Learning
FR	Functional Requirement
GIS	Geographic Information System
gNB	Next Generation Node B
gNMI	gRPC Network Management Interface
GPR	Gaussian Process Regression
GPU	Graphical Processing Unit
gRPC	google Remote Procedure Call
ID	Identification
IETF	Internet Engineering Task Force
ISG	Industry Specification Group
ITU	International Telecommunication Union

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

KPI	Key Performance Indicator
KPM	Key Performance Metrics
LCM	Lifecycle Management
LLM	Large Language Models
LR	Lineal Regressor
MANO	Management and Orchestration
MAPE-K	Monitor-Analyze-Plan-Execute over a shared Knowledge
MDP	Markov Decision Process
ML	Machine Learning
MLOps	Machine Learning Operations
MPTCP	Multipath TCP
MQTT	Message Queuing Telemetry Transport
NATS	Neural Autonomic Transport System
NDT	Network Digital Twin
NETCONF	Network Configuration Protocol
NF	Network Function
NFR	Non-Functional Requirement
NFV	Network Function Virtualization
NS	Network Service
NSMF	Network Slice Management Function
NSSMF	Network Slice Subnet Management Function
NWDAF	Network data analytics function
OODA	Observe - Orient – Decide – Act
PM	Probabilistic Model
PT	Physical Twin
RAN	Radio Access Network
REST	Representational State Transfer
RL	Reinforcement Learning
RSG	RAN Scenario Generator
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SBA	Service-Based Architecture
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service Level Agreement
SO	Specific Objective
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TGW	Telemetry Gateway
TOSCA	Topology and Orchestration Specification for Cloud Applications

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

TRA	Threat and Risk Assessment
UDM	Unified Data Model
UDP	User Datagram Protocol
UDR	Unified Data Repository
UE	User Equipment
VNF	Virtual Network Functions
WP	Work Package
YANG	Yet Another Generation
ZSM	Zero-touch Service and Network Management

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Table of contents

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Table of figures

Table of tables

1. Introduction

6G promises to implement Artificial Intelligence (AI) and Machine Learning (ML) suitable for networking, an unfulfilled promise from 5G networks to deal with the unprecedented complexity and dynamism across the entire communications ecosystem. Addressing the stringent demands of next-generation applications, such as immersive extended reality, teleoperated mobility, and large-scale digital automation, requires a fundamental shift in how networks are modelled, managed, and optimized. To that end, the Network Digital Twin (NDT) is introduced as a foundational concept to bridge the gap between real-world network infrastructures and their intelligent, not yet implemented counterparts.

An NDT is defined in this context as a real-time replication of network behaviour and status, underpinned by mathematical and AI-enhanced modelling methods that enable understanding, prediction, and pre-validation (analytical NDT) or active management, optimization, and direct influence over the physical network (controlling NDT). Analytical NDT instances can be used to generate hypothetical ("what-if") scenarios to evaluate how the physical network would respond to different setups, allowing proactive refinement of network behaviour. They also are crucial for anticipating possible issues according to historical data and current network behaviour, optimizing network performance before they manifest in the real world. Finally, analytical NDTs serve as a crucial solution for overcoming modelling challenges by allowing for the reliable and safe training/testing of AI-based Network Functions and Services (NFs and NSs) before deployment in the physical network. On the other hand, the ultimate stage in NDT evolution is the controlling NDT, which not only achieves a highly accurate representation of the physical network but also earns sufficient trust to enable fully autonomous control, removing the need for human intervention in the decision-making loop.

These capabilities are made possible thanks to several main technological enablers. First, a robust data collection framework, as detailed in Deliverable 1.2, that efficiently ingests, harmonizes, and processes real-time telemetry data from various sources and across multiple network domains (radio access, core, transport, edge, cloud). This ensures a unified data repository for accurate, up-to-date representation of the physical network, enabling efficient data management and analysis. Second, a Network MANO/Zero-touch Service and Network Management (ZSM) layer that creates the principles to automate processes for network operations and oversee interactions between the physical network and the digital network, with reduced or even without human intervention. Third, the tight integration of AI-based workflows, simulation environments, and orchestration mechanisms across distributed domains, providing the ability to learn from data and enable the network to predict changes, generate synthetic data, and perform "what-if" analyses on the network, and continuously improve network performance supporting the validation and refinement of AI-based NFs.

Building on the architectural principles introduced in earlier deliverables (D1.1 and D1.2), this document (D1.3) expands the functional architecture of 6G-TWIN to present an enhanced view on the Management and Orchestration (MANO) layer, that spans not only the physical world but also the digital one by encompass the lifecycle of NDT instances, as well as their use within the closed-loop operation in the physical network. It formalizes how Digital Twins (DTs) should be instantiated, maintained, and evolved in real-time to support intelligent network behaviour, and details how data, models, and simulators interoperate to close the loop between AI insights and network operations.

This deliverable provides three key contributions:

- Definition and Management of NDT Instances: We introduce the concept of an NDT instance as a context-aware, scenario-specific configuration composed of basic and functional models. These instances are managed dynamically by the NDT MANO layer to adapt to changing network requirements.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

- AI-Driven Functional Modelling and Simulation Integration: We describe how predictive and learning-based models are trained and integrated using telemetry from the physical network and simulation-based what-if scenarios. This is achieved through a harmonized data ingestion pipeline, supported by a modular AI training infrastructure.
- Validation Through Practical Implementations: Building on the 6G-TWIN architecture, we showcase several proof-of-concept workflows, such as energy-efficient Radio Access Network (RAN) optimization, User Equipment (UE) localization, and radio coverage prediction, demonstrated at several international venues. These highlight the end-to-end interaction between telemetry ingestion, model deployment, simulation orchestration, and automated NDT updates.

This deliverable marks the final piece defining the functional architecture and its associated processes. With the theoretical groundwork laid out across Deliverables D1.1, D1.2, and D1.3, the remaining phase of the project will focus on implementation, translating these concepts into practical applications. The insights gained from this hands-on work are expected to refine and validate the architectural elements introduced so far, effectively bridging the gap between theoretical design and real-world deployment of NDTs in 6G networks. This progression will support the realization of scalable, AI-native network MANO.

1.1. Aims and objectives

1.1.1. 6G-TWIN objectives

In response to the accelerating digitization across industries, the 6G-TWIN project emerges with a singular mission: to pioneer an AI-native reference architecture for the forthcoming 6G systems. At its core lies an ambitious vision to seamlessly integrate NDTs into the fabric of future networks, revolutionizing their optimization, management, and control in real-time.

To achieve its ambition, the 6G-TWIN has been built around several specific objectives:

- Specific Objective 1 (SO1) is central to the project's ambition, promising to design an open, federated and AI-native network architecture for the imminent 6G landscape. This architectural blueprint is designed to leverage NDTs, empowering intelligent data analytics and real-time decision-making, thereby laying the groundwork for unprecedented network efficiency and performance.
- Moreover, Specific Objective 2 (SO2) underscores the project's commitment to constructing a federated, graph-based NDT capable of accurately representing the intricate dynamics of highly dynamic and complex network scenarios. By establishing this digital sandbox for network planning, management, and control, 6G-TWIN paves the way for enhanced operational agility and adaptability.
- Simultaneously, Specific Objective 3 (SO3) drives the project's efforts towards implementing a robust modelling and simulation framework. This framework serves as a cornerstone for accurately portraying networked environments and rigorously testing the functionalities of the envisioned 6G architecture.
- Ultimately, as the culmination of its efforts, 6G-TWIN aims to materialize Specific Objective 4 (SO4) by testing, validating, and demonstrating the transferability of its solutions. Through the development of dynamic demonstrators catering to tele-driving and energy efficiency use cases, the project aims to showcase the practical impact of its architectural foundation on real-world network scenarios, heralding a new era of connectivity and innovation.

Embedded within the core of the 6G-TWIN project lies a foundational framework driven by specific objectives aimed at revolutionizing the architecture of future 6G systems.

1.1.2. Deliverable objectives

Deliverable D1.3's primary aim is to refine and extend the 6G-TWIN functional architecture by introducing a more comprehensive MANO layer, called the 6G-TWIN MANO. This vertical layer consolidates the managers and orchestrators of all subcomponents critical to operating an AI-native architecture that integrates NDTs. Specifically, this deliverable seeks to:

- Introduce a vertical MANO layer within the 6G-TWIN architecture, which coordinates the orchestration and Lifecycle Management (LCM) of the Physical Twin (PT) and DT-related components.
- Provide an enhanced decomposition of NDTs, distinguishing between a running NDT instance and the broader superset of available basic and functional models, along with internal architectural elements and lifecycle workflows.
- Define the role of Network MANO/ZSM as a key enabler of fully automated, AI-driven service and network operations in 6G, leveraging intent-based control and Closed-Loop Automation (CLA) across physical and digital domains.
- Present ZSM-aligned design principles that support the integration of NDTs and ML, facilitating a seamless bridge between simulated and real-world environments for robust, adaptive, and automated 6G network management.

The present document is an initial version, considering the work done so far in the first half of 6G-TWIN. A final version, Deliverable D1.6, will be provided at the end of the project.

1.2. Relation to other activities in the project

The overarching goal of the 6G-TWIN project is to lay the groundwork for the design, implementation, and validation of an AI-native architecture for 6G networks. At its core, this architecture integrates NDTs as a central mechanism for enabling End-to-End (E2E), real-time optimization, management, and control of increasingly dynamic and complex network environments.

To support this objective, Work Package 1 (WP1) is dedicated to defining the architectural framework. Deliverable D1.1 introduced the initial architecture, while D1.2 expanded upon it by detailing data collection methodologies. Building on these foundations, this deliverable (D1.3) further refines the architecture by introducing a more comprehensive MANO layer. Referred to as the 6G-TWIN MANO, this vertical layer consolidates the MANO functions of the various subcomponents required to operate an AI-native, NDT-integrated architecture.

This deliverable consolidates the outcomes of three key tasks within the project. First, Task 1.3: Zero-touch network planning, management, and control aims to evolve the traditional control layer into an AI- and NDT-driven closed-loop system capable of automating the management of network and infrastructure resources across multiple time scales and domains. The goal is to reduce human intervention while enhancing efficiency, programmability, and adaptability. Building on the foundations laid by Task 1.3, Task 1.4: Federated MANO of AI-based NF/NS focuses on designing a federated MANO framework to support the orchestration and LCM of such NFs and NSs in distributed, multi-domain environments. Lastly, Task 2.4: NDT models for network planning, management and control combine the models developed in Tasks 2.1, 2.2, and 2.3 to construct and validate NDT instances, enabling practical implementations of AI-native management and control in 6G networks.

1.3. Report structure

This deliverable is structured in six sections, with four presenting technical content related to the three tasks as outlined previously. The report structure is organized as follows:

- Section 1 introduces the deliverable and outlines the motivation and overarching goals of the within the 6G-TWIN project. It contextualizes the evolution from earlier deliverables (D1.1 and D1.2) and introduces the focus on MANO for NDT-based architectures.
- Section 2 presents an evolved version of the 6G-TWIN architecture, highlighting the introduction of a vertical MANO layer that orchestrates and supervises NDT instances and their subcomponents, enabling zero-touch automation and dynamic coordination.
- Section 3 explores the mechanisms required for the 6G-TWIN Network MANO and ZSM, and how to apply ZSM principles to automate network and service management for NDT-enabled systems. The section also outlines interfaces and interactions needed for AI-driven, intent-based control across physical and digital domains, as well as some NDT operations in the context of 6G-TWIN.
- Section 4 describes the decomposition of NDTs into basic and functional models and provides workflows for managing their lifecycle—from instantiation and monitoring to updates and termination.
- Section 5 details real-world demonstrations that validate the architectural principles presented, including use cases for UE localization, radio coverage prediction, and automatic functional model updates in response to changing network conditions. This section is intended to be more implementation-oriented than theoretical and is therefore aimed at an expert audience rather than a general one.
- Section 6 concludes the document.

1.4. Contribution of partners

The following table presents the partners' contributions to the deliverable, where bold numbers indicate the section's leader.

Table 1. Partner's contribution to D1.3.

Partner	Section / Subsection (s)	Contribution
IMEC	**1, 2.1, 2.2.2, 2.3, 4, 5.3, 6**	IMEC is the main editor of the deliverable, responsible for the outlining the table of contents, coordinating the inputs and editorial revision from the document, writing the introduction and conclusions. Additionally, IMEC is the main responsible for the technical content of Section 2.2.2 (Federated AI-based Network Functions and Services MANO) and 2.3 (NDT Instance), Section 4 (NDT internal operations), and Section 5.3 (NDT Update: Functional Model Based on Deep Learning).
UBI	2.2.1, **3.1, 3.2, 3.4**	UBI has contributed to Section 2.2.1 (Network MANO with ZSM capabilities) and coordinated the contributions on Section 3 (ZSM-enabled MANO for 6G-TWIN), providing an introduction, with Section 3.1 (Evolution of MANO: from NFV to AI-native ZSM architectures), Section 3.2 (Network MANO/ZSM automation mechanisms) and Section 3.4 (Functional and non-functional requirements)
LIST	2, **3.3.1**, 4, **5.2**	LIST supported sections 2 and 4. Responsible for the content in sections 3.3.1 (Beyond traditional methods: unlocking the power of generative AI for NDT LCM automation) and 5.2 (NDT creation: predicting radio coverage). Document review.
ACC	**2.2**, 2.4, **2.7, 5.1**	ACC has supported the definition of the physical functional architecture based on open interfaces while taking into consideration the data and telemetry framework for the creation of the basic and functional models.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

EBY	**2.4**, **3.3**	EBY supported Section 2.4.1 with the strategies for Data Management with a special focus on Security and Privacy. Additionally supported Section 3.3 with NDT uses for federated learning.
POLIBA	**2.5**	POLIBA has contributed to section 2.5 (AI training and workflow management)
TUD	**2.6**	TUD coontributed by writing Section 2.6 (Simulation Framework)
UBOU	**3.3**	UBOU contributed with two use cases for NDTs in Section 3.

1.5. Deviations from the Grant Agreement

Section 5 of this document presents early implementations and examples demonstrating the feasibility and effectiveness of the proposed NDT framework within the context of the 6G-TWIN project. These efforts also draw contributions from other work packages, particularly WP4, which has not yet officially started, and are more technical than theoretical in nature. Nevertheless, the implementations effectively illustrate and bring to life the concepts explored in D1.1, D1.2, and D1.3.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

2. 6G-TWIN functional architecture with a focus on management and orchestration aspects

Modelling modern communication networks presents significant challenges due to their inherent complexity, heterogeneity, and dynamic nature. Contemporary networks span multiple operational domains, such as radio access, core, edge, and cloud, and involve various interacting components across different layers. These systems integrate physical and Virtual Network Functions (VNFs), built upon diverse technologies requiring advanced modelling paradigms encompassing mathematical frameworks, simulators, and emulators.

Despite the progress made through traditional modelling approaches, several key limitations remain. These methods often have high computational overhead, lack scalability, and struggle to deliver real-time analysis or consistent performance evaluation in large-scale, heterogeneous network environments. This limits their utility in scenarios that require dynamic, context-aware, and continuous network insight.

NDTs are a crucial solution for overcoming these challenges. They offer an effective abstraction of the network, i.e., the PT, allowing for real-time replication of network behaviour and status, underpinned by mathematical and AI-enhanced methods. NDT provides extensive monitoring and automated management capabilities while enabling predictive analysis, optimization, and decision-making throughout the entire lifecycle of network services.

In D1.1 [1], 6G-TWIN proposed a functional architecture for an AI-native 6G network that integrates NDTs in their closed-loop control operations. The architecture is divided into three layers: the physical and digital networks and the human world. The physical network includes the UE, the surrounding environment that can influence the propagation patterns, the RAN, and the Core Network (CN). The digital world is highly dominated by the NDT, which was designed in line with the preliminary work on the topic. Besides the NDT, this layer comprises the basic and functional models required to form the NDT and a simulation framework, which supports some operations of the NDT, such as what-if analyses and the creation of functional models. Finally, the human world offers a way to interact with the NDT in the form of network applications, following a completely closed-loop control, or an open-loop control where human operators are involved.

Besides the updates presented in D1.2 [2] related to the data collection framework, this deliverable, and the work presented at the European Conference on Networks and Communications (EuCNC) 2025 [3] update the architecture presented in D1.1, to consider a more detailed view of the components needed to operate AI-native 6G network that integrates NDTs as a core mechanism for network optimization and control. This section formally presents such updates and focuses more on the MANO layer, to oversee lifecycle and interactions between persistent and on-demand layers, ensuring seamless transitions between data-driven and simulation-driven workflows. Section 2.1 summarizes the key updates and presents the resulting 6G-TWIN functional architecture. The subsequent sections provide detailed descriptions of each architectural block and explain how the MANO layer oversees and coordinates their processes as needed

2.1. Updates on the 6G-TWIN architecture

An NDT architecture must integrate intelligent mechanisms for seamless orchestration, enabling real-time simulation, analysis, and optimization of network operations. To ensure effective implementation, the NDT's functional architecture should include a data collection framework that enables seamless data integration, access policies, and scalable processing.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Moreover, all the components in the architecture should be designed with ZSM principles in mind to support AI-driven automation, real-time monitoring, and resource optimization.

Furthermore, the MANO layer should support federated operation and ensure AI-based LCM, standardized APIs, and Machine Learning Operations (MLOps) support. Finally, the Simulation Framework shall enable tailored simulations and platform interoperability. Non-Functional Requirements (NFRs) emphasize performance and scalability, ensuring optimized resource usage, minimized latency, and platform-independent operation, while maintaining privacy and compliance across all components.

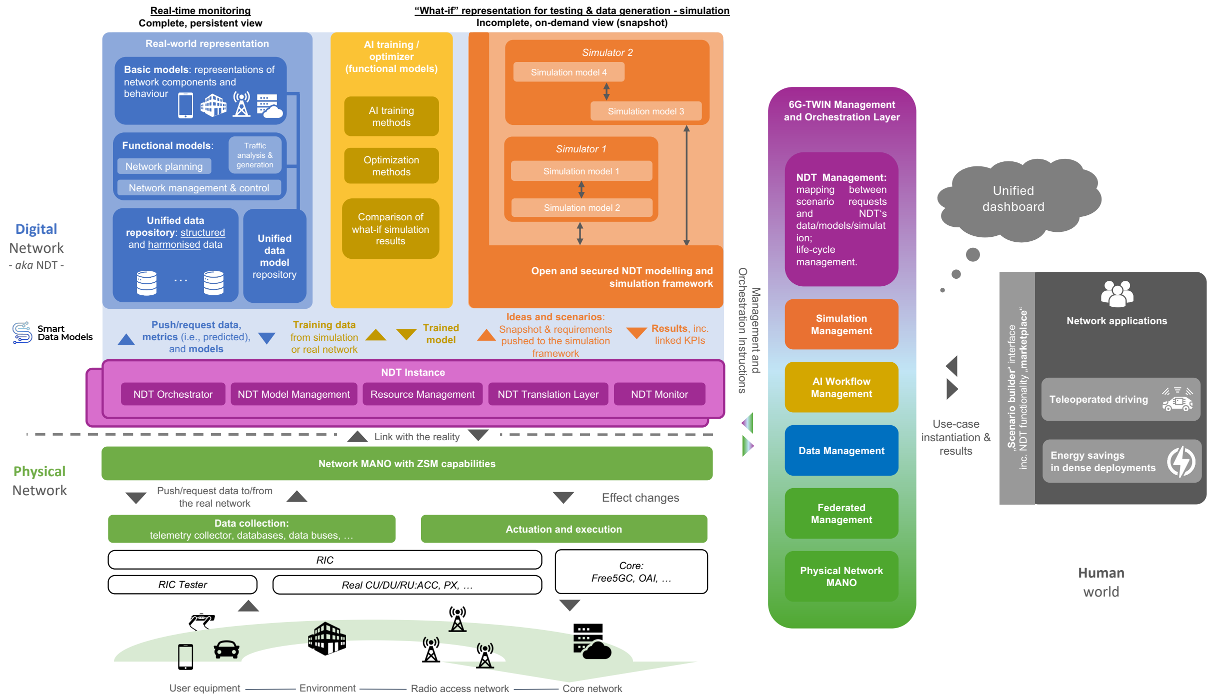

Figure 1. Updated 6G-TWIN functional architecture.

Therefore, the paper [3] presented at EuCNC 2025 updates the initial architecture presented in D1.1. Our proposal includes four key pillars: (i) a data collection framework for dynamic data acquisition, (ii) ZSM for AI-driven automation, (iii) Federated MANO (F-MANO) for decentralized network control, and (iv) the Simulation Framework for predictive modelling.

This deliverable presents further refinements to the architecture introduced in the EuCNC paper, focusing on the MANO layer, as illustrated in Figure 1. Managing NDTs requires coordination across diverse components, including the PT, network simulators, and various models. The updated 6G-TWIN MANO layer is proposed as a unifying vertical layer that integrates these domains. **This layer encompasses the MANO functions of both the physical and digital network components, including the Federated MANO (F-MANO), the simulation framework, and the AI workflow.**

At its core, the **NDT Management** block orchestrates the instantiation and lifecycle of NDT instances. At the same time, the **Data Management** component ensures that real-time telemetry continuously feeds the basic and functional models, supporting their dynamic adaptation. Through integrated **AI workflow and simulation management**, this layer enables dual functionality for NDT instances: (a) serving as a sandbox for training and validating AI algorithms before deployment in the live network, and (b) generating hypothetical ("what-if") scenarios to assess how different configurations would impact network behaviour. The remainder of this section will describe the main elements of the updated 6G-TWIN architecture.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

2.2. Physical network

The physical network or the PT in green colour (cf. Figure 1) serves as the infrastructure backbone, encompassing the RAN, CN, Transport Network (TN), and edge and cloud computing resources. Guided by **network programmability** [4] and **ZSM** principles [5], it enables AI-driven automation and dynamic optimization. This layer includes the data collection framework, the network MANO layer, and mechanisms to implement optimization outcomes. In this context, NDTs work as analytical services, supporting data-driven decision-making. They enable safe training and testing of algorithms or directly generate decisions for controllers and orchestrators to apply to the physical network. This enables efficient cross-domain coordination with continuous feedback loops for self-optimization and adaptive management.

2.2.1. Network MANO with ZSM capabilities

The DT concept has been explored for mission-critical applications ever since the 1960s [6], but only recently, with the **development of 6G** technologies, has this paradigm been able to evolve towards covering the **whole physical network infrastructure** [7], [8].

One of the pillars of the NDT is the data collection performed on the physical network: **analysing**, **diagnosing**, and **simulating** the network requires the real-time collection of massive amounts of data across different network objects. Moreover, enacting the necessary changes for the major role of network optimisation of the NDT involves some level of **control** across these different network objects; this is the main challenge MANO across the NDT, to maintain the connection between network objects from various network domains (RAN, Core, Edge, Cloud), **manage the immense volumes of network traffic and data generated**, and orchestrating the control across multiple devices with different communication protocols.

The International Telecommunication Union (ITU), a United Nations agency specialised in digital technologies and responsible for coordinating shared global use of telecommunication networks, developed a recommendation [9] for the development of an NDT (referred to as Digital Twin Network); in this document, the NDT needs to target four main requirements to support the DT paradigm: a framework that **stores the massive amounts of data** collected from the physical network in a **unified data repository**, a **real-time mapping** between the physical network and its digital representation, network device and application modelling, and a **standardised** southbound and northbound interfaces between the components of the NDT.

Another recommendation [10] outlines the key role of NDTs in managing **and orchestrating 5G networks and beyond**, extending the potential benefits to network slicing through **optimal resource allocation, efficient data collection, and network simulation.** Some general requirements for NDT in this case are the collection of different types and frequencies of data, through different tools and methods; the support for the registration and authentication of multiple DT entities, as well as monitoring the status and performance of these very same entities; and finally, the capability to receive and analyse different types of simulation results, as well as enacting the optimal outcomes of the DTs across the network topology, preventing any potential conflicts.

Management of the NDT lifecycle also needs to address these requirements. Using data collected from the network state and devices enables **real-time decision-making**. Still, the integration orchestration of the data collection process from multiple devices, in various network domains, is a challenge that needs to be addressed by the NDT. **ZSM** is a concept born from this challenge, referring to the capability of networks and services to be deployed, configured, integrated, maintained, and optimised with minimal or no human (zero-touch) intervention. The European Telecommunications Standards Institute (ETSI), a European Standards Developing Organization founded by the European Commission, in the domain of telecommunications, explores this field with its Industry Specification Group (ISG) ZSM [11].

Their role is to explore and develop mechanisms enabling the full E2E **network and service management automation**.

To remove the need for human intervention, a management layer between the **physical network and its digital representation** was originally proposed in D1.1 [1], illustrated in Figure 2 by both green boxes (Zero-touch network management and control, and Federated MANO of AI-based NFs), and now referred to as **Network MANO with ZSM capabilities (or Network MANO/ZSM)**. With this proposal, the focus on developing an intelligent orchestration and automation layer, implementing a **closed-loop control system** (a system model that continuously monitors, assesses, and adjusts itself based on real-time data and feedback) and **AI/ML mechanisms** for the enhancement of network management and control, alongside traditional LCM capabilities; self-monitoring, self-analysis and self-adjustment capabilities introduced by the ISG ZSM are also critical considerations for this **Network MANO/ZSM** layer, in order to maintain optimal performance and rapid response to network and device changes.

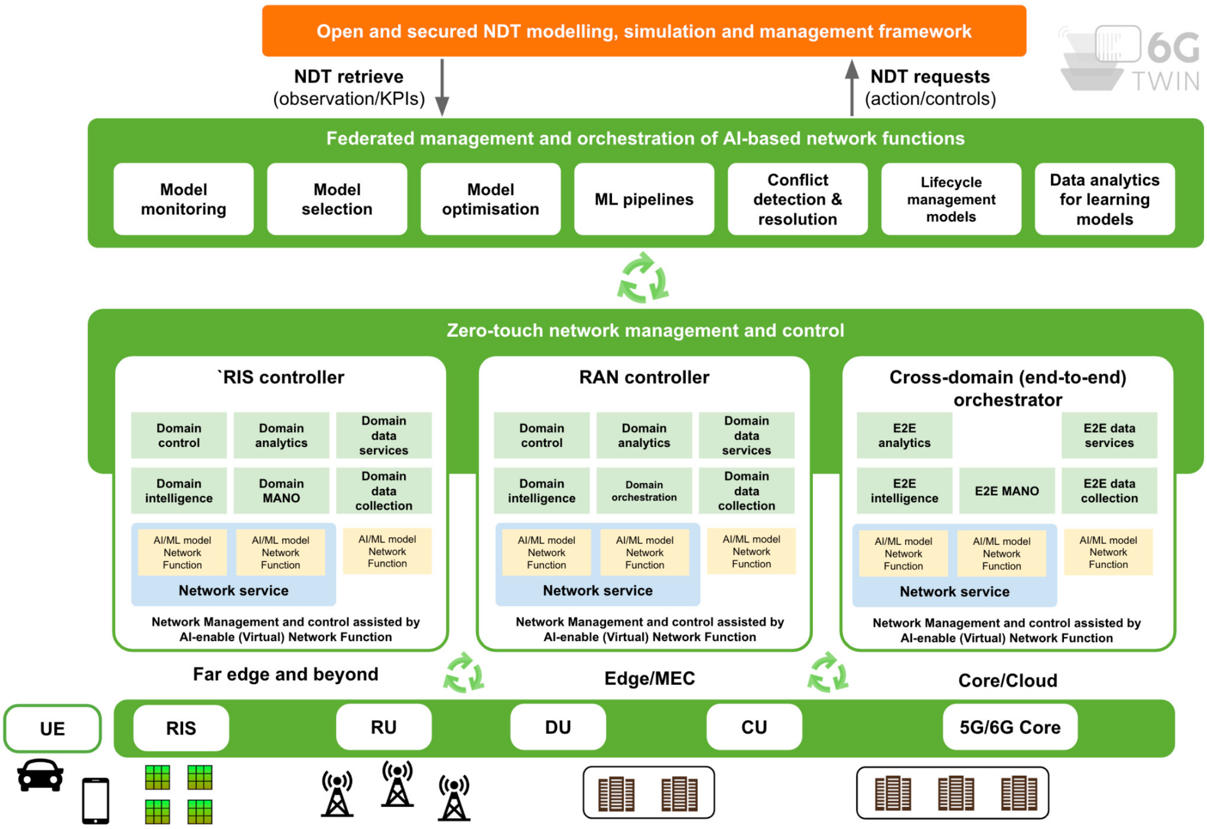

Figure 2. High-level architecture of 6G-TWIN Physical Network MANO/ZSM.

This Network MANO/ZSM layer also focuses on two main considerations: novel mechanisms that increase **network programmability** and support closed-loop management strategies and **efficient data harmonisation pipelines**. There are two important closed-loops outlined in this deliverable: the **internal closed-loop**, tied to the mechanisms that support the creation of the NDT and NDT Components; and the **external closed-loop**, encompassing the MANO of the operations of the NDT lifecycle.

The internal closed-loop is implemented in a modular way, enabling a simpler deployment of the NDT main building blocks; this instantiation focuses on setting up the **necessary interfaces** (southbound, northbound and intra/inter-NDT) that will enable the communication between the NDT with the physical network components, the digital layer components (such

Co-funded by the European Union
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

as the **AI Training/Optimiser** and **Simulator Framework**, represented by the orange block), and the network application providers.

The external closed-loop is focused on the main tasks performed by the NDT during its operation, **defining automated support mechanisms** such as: enabling AI-based learning and real-time maintenance, through the analysis of basic and functional models by the AI training/optimiser and Simulation Framework, diagnosing and determining an optimal configuration; the process of **applying this configuration** across the network topology and network objects; and support to the data pipelines that handle diverse data formats, protocols, and traffic patterns, as well as the **harmonisation of this data** in a standardised model format like Smart Data Models, as outlined in [2].

Besides these internal network management capabilities, the **Network MANO/ZSM** layer should also be capable of auto-discovery, broadcasting device capabilities and securely **onboarding physical network objects** into the NDT topology. The physical network state of devices should also be translated so that the Network MANO/ZSM is capable of providing the best possible configuration from the optimisation and simulation processes; this can be done by attaching a specific **high-level intent** to network applications, as an optimal goal to achieve; the Network MANO/ZSM then translates this high-level intent into operational tasks, to optimise these services, simplify the necessary communication between different types devices.

2.2.2. Federated MANO

Modern and future communication networks are inherently composed of multiple **autonomous systems,** each managed by distinct administrative domains. These domains independently define and enforce their own policies to retain control over their resources, leading to a highly decentralized and heterogeneous environment. Moreover, the methodologies employed to formulate and implement these policies vary significantly across domains, reflecting diverse operational goals, technical capabilities, and governance models [12]. This decentralized nature poses significant challenges for the **federated MANO (F-MANO)** (cf. Figure 1 or Figure 2) of NFs and NSs.

To ensure E2E service stability and reliability across federated environments, services must be decomposed into resource slices that span two or more administrative domains. Recent 3GPP releases support this vision by introducing a hierarchical architectural model that facilitates such cross-domain decomposition. For example, 3GPP Release 16 [13] defines three core NFs essential for 5G network slicing: the Communication Service Management Function (CSMF), the Network Slice Management Function (NSMF), and the Network Slice Subnet Management Function (NSSMF). The CSMF translates high-level service requirements into network slice descriptors. The NSMF then decomposes these slice requirements into domain-specific objectives, while each NSSMF, operating within its respective domain, orchestrates and manages the local resources allocated to the slice. This hierarchical structure enables scalable and modular orchestration, echoed in other emerging standards that aim to support federated, multi-domain network service management [14], [15].

While this hierarchical model provides a solid foundation for structured service orchestration, the integration of AI-based NFs and NSs introduces new layers of complexity and potential disruption. In such a federated environment, orchestrating AI-based NFs demands mechanisms that can operate across heterogeneous domains without violating local policies or compromising administrative autonomy. Ensuring consistent behaviour, resource optimization, and service quality becomes increasingly complex when orchestration decisions must be negotiated across varying trust levels, policy frameworks, and infrastructural constraints. Crucially, these systems must maintain **network stability and reliability**, even as services are dynamically instantiated, migrated, or scaled across administrative boundaries.

Unlike traditional NFs, AI-driven functions exhibit dynamic behaviour, context sensitivity, and learning-based adaptability, which may not align neatly with the static decomposition principles currently defined by standards like 3GPP. For instance, AI functions often require continuous model updates, real-time data access, and cross-layer feedback loops, characteristics that challenge the predictability and determinism expected in hierarchical orchestration frameworks.

Moreover, the autonomy of AI functions can blur the boundaries between architectural layers. An AI-based optimizer embedded at the NSSMF level might influence upstream decisions typically handled by the NSMF, creating vertical coordination challenges. Similarly, deploying distributed AI agents across domains raises questions about horizontal synchronization, trust, and consistency, especially when decisions in one domain affect service quality in others.

As such, incorporating AI-based services into federated architectures necessitates a rethinking of the hierarchical model to support more flexible, adaptive, and intelligence-aware orchestration mechanisms, capable of maintaining E2E stability while leveraging the full potential of AI. Achieving this necessitates deploying appropriate interfaces for exchanging information, knowledge, intentions, and optimization strategies.

To support the seamless integration of AI-based NFs within federated and multi-domain environments, an alternative could be based on a Cross-layer AI Orchestration Plane that spans vertically across the existing CSMF–NSMF–NSSMF hierarchy [16]. This plane coordinates the lifecycle, placement, and monitoring of AI components, such as inference engines, training agents, or optimization modules, by dynamically interfacing with each layer:

- **CSMF Level:** At the highest abstraction layer, AI models assist in translating ambiguous or high-level service intents into structured slice descriptors using techniques such as natural language understanding [17], [18], [19] and policy graph modelling [20].
- **NSMF Level:** AI is used for multi-objective optimization tasks, such as predicting slice performance under various resource allocations, leveraging historical service-level data, and performing risk-aware decomposition [21]. Reinforcement learning (RL) agents are particularly useful for dynamically adjusting slice boundaries based on observed behaviour and demand.
- **NSSMF Level:** At the domain level, real-time AI models are embedded into the orchestration pipeline for localized decision-making, such as traffic prediction [22], anomaly detection 6/30/25 11:03:00 PM, and resource scaling [23], [24]. These models operate under domain-specific constraints and report their insights upward for global coordination.

Given the autonomous and often mutually distrustful nature of administrative domains, AI-based orchestration requires a dedicated mechanism for **inter-domain coordination** that respects sovereignty while promoting collaborative intelligence. In such cases, a federated intelligence coordination layer might help by operating horizontally across the NSSMFs of different domains and providing the following functions:

- **Federated Learning (FL) Coordination:** Supports training of AI models using FL protocols [25], allowing local data to remain within its origin domain while contributing to global model convergence.

- **Model Exchange and Verification:** Facilitates the sharing of AI inference engines or policy models using secure attestation and sandboxing techniques [26].
- **Trust and Compliance Management**: Incorporates decentralized identity and trust mechanisms (e.g., blockchain-based attestations [27]) to ensure only verified and compliant AI models influence cross-domain orchestration.

However, the FL approach assumes that all the federation participants share the same AI model structure and possess similar data distributions. Alternatively, heterogeneous FL [28] approaches could be used. Another alternative is to employ a federated NDT approach while preserving privacy and sovereignty. For example, data may be gathered locally, then homogenized and anonymized before being shared with other NDTs. In this way, the differences between domains are abstracted, allowing the federated solution to be prototyped in the digital world and later deployed in the physical network.

2.3. NDT instance

In purple (cf. Figure 1), a running instance of an NDT is depicted. This instance consists of a combination of basic and functional models. The NDT instance relies on several key components to function correctly, as illustrated in the figure. These include: an **NDT Orchestrator**, which coordinates all internal processes related to the creation and execution of the NDT; a **Model Manager**, which governs the interaction between basic and functional models and ensures their performance, triggering corrective actions when necessary; a **Resource Manager**, responsible for overseeing the computing resources used by the NDT instance; and an **NDT Monitor**, which continuously tracks the status and behaviour of the NDT, issuing alerts in case of anomalies. Additionally, an **NDT Translation Layer** is required to interpret human inputs for the NDT and convert its outputs into human-readable information. A more detailed description of these components is provided in Section 4.

2.3.1. NDT MANO

The **NDT Management** block is a fundamental component responsible for the operation, coordination, and LCM of NDT instances and their subcomponents [29], [30]. This block serves as an intelligence layer, synthesizing consumer requirements, network context, and available resources. Among its key responsibilities are (i) overseeing the instantiation, configuration, and orchestration of NDT components based on application and operational requirements; (ii) monitoring the resource utilization and ensuring optimal performance of twin instance; (iii) managing the lifecycle of NDTs, including creation, adaptation, and termination, aligned with the evolution of network services and topologies. Furthermore, the NDT MANO layer governs the integration and operation of all components within the NDT architecture, ensuring synchronized interactions between the physical network and the NDT elements, including data, models, simulation framework, and AI training mechanisms.

NDTs may operate at varying levels of abstraction, tailored to specific management needs. In Node-level twins, the focus is on individual network elements or functions, whereas Domain-level twins encapsulate broader segments, such as the RAN, core, edge, or cloud. Similarly, Service-level twins span multiple domains to provide an E2E representation of service delivery chains.

Independent of the level of abstraction, the creation and operation of a DT involve several stages [31], [32], [33], [34]; thus, the NDT management block is seen as the mastermind behind the previously mentioned blocks within the 6G-TWIN management layer. It begins with requirements gathering and planning, where the scope is defined and the relevant assets and systems to be modelled are identified. This is followed by data acquisition, which involves collecting and integrating information from diverse sources to build a reliable foundation for the

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

twin. Once the data is in place, a virtual representation is created using appropriate modelling techniques tailored to the system's complexity.

Once the virtual entities composing the NDT are created, the next step involves federation, integration, and ensuring interoperability by connecting the DT with existing systems and adhering to established industry standards and communication protocols. In large-scale or federated environments, such as multi-operator or cross-domain scenarios, multiple NDT instances may operate concurrently. This necessitates state synchronization, coordination, and policy-driven orchestration mechanisms to maintain coherence and performance across the DT federation. Similarly, depending on the use case, a federation might imply the integration of an NDT instance with other simulators, platforms, or frameworks [35]. If different simulators must be coupled, the simulation framework plays an important role in synchronizing the state among multiple simulators (as explained in Section 2.6).

The deployment phase then places the DT in the desired environment—whether in the cloud, on-premises, or at the edge—while establishing seamless data exchange with its physical counterpart. The twin is then used for simulation and analysis, enabling the prediction of potential issues, performance optimization, and decision support. Ongoing monitoring ensures that the twin stays accurate by continuously updating it with real-time data from the physical system. This continuous refinement is key to maintaining model fidelity. In parallel, collaborative and visualization tools allow users to interact with the DT, share insights, and make informed decisions. Furthermore, the DT facilitates process optimization by enabling scenario exploration and hypothesis testing, improving operational efficiency, and reducing costs. Finally, maintenance and LCM ensure the DT remains relevant over time through regular updates, enhancements, and alignment with evolving system conditions. Section 4 provides a more detailed examination of the steps outlined above.

2.4. Real-world representation

In blue (cf. Figure 1), this layer allows the creation of the necessary models to represent the physical network accurately. At its core, the **Unified Data Repository (UDR)** is a centralized storage system aggregating historical and real-time data from network infrastructure, sensors, and external contextual sources. It enables efficient data harmonization and retrieval to support decision-making within the NDT framework.

The **Unified Data Model (UDM)** repository defines the structure of network representations, consisting of basic and functional models. Basic models capture the real-time state of physical and virtual network elements, including configurations, topology, and environmental conditions, ensuring an accurate emulation of network dynamics. They serve as a foundation for validation and control mechanisms. In contrast, functional models build upon these insights to optimize network operations, predict behaviour, and improve decision-making, often incorporating AI-driven techniques. The interaction between the model and data management components ensures the LCM of NDT instances, aligning them with application-layer needs. This layer enables a continuously evolving and self-adaptive NDT environment by facilitating seamless integration between data, basic, functional, and simulation model repositories.

2.4.1. Data management

The evolution of 6G networks introduces significant complexity and diversity, making efficient and secure data management crucial. In the NDT-enabled 6G architectures, effective data management supports real-time decision-making, optimization, and orchestration, addressing scalability, reliability, and security challenges within the 6G-TWIN MANO layer.

Data management ensures data integrity, availability, and usability across systems, especially in dispersed cloud-based environments. The data management provides the real-time network data feeds to the basic and functional models, enabling a continuous refinement of models, simulations, and AI-driven functionalities. Protecting sensitive data involves controlling access through role-based access control and multi-factor authentication, while preserving data integrity. Additionally, privacy regulations must be considered, which require data masking or anonymization operations. The vulnerabilities that cause malicious exploitation are increasing as the system expands. Therefore, it is crucial to have key requirements, including maintaining data consistency across nodes, preventing the sharing of sensitive data, and providing secure data management throughout the system.

As explained in D1.2 [2], 6G-TWIN proposes an integrated data management system that extends a data exposure and collection architecture, where data is moved from the physical world to the digital and human world via diverse interfaces. Based on the Internet Engineering Task Force IETF RFC9232 [36], the 6G-TWIN architecture includes the **Generation, Collection Processing, and Consumption Data Modules**, mapped in the 6G-TWIN architecture via the telemetry data layer composed of the **Data ingestion, Data processor, Data storage, and Data Stream** output. These modules support data MANO services, such as security, scalability, and federated distributed parallelization.

Additionally, the **Data Harmonization** layer acts as a bridge between the Telemetry layer and the NDT instance. It ensures that the raw telemetry data is transformed into standardized formats compatible with structured Smart Data Models. These models serve as the foundation for defining and operating the entire architecture, as illustrated in Figure 3.

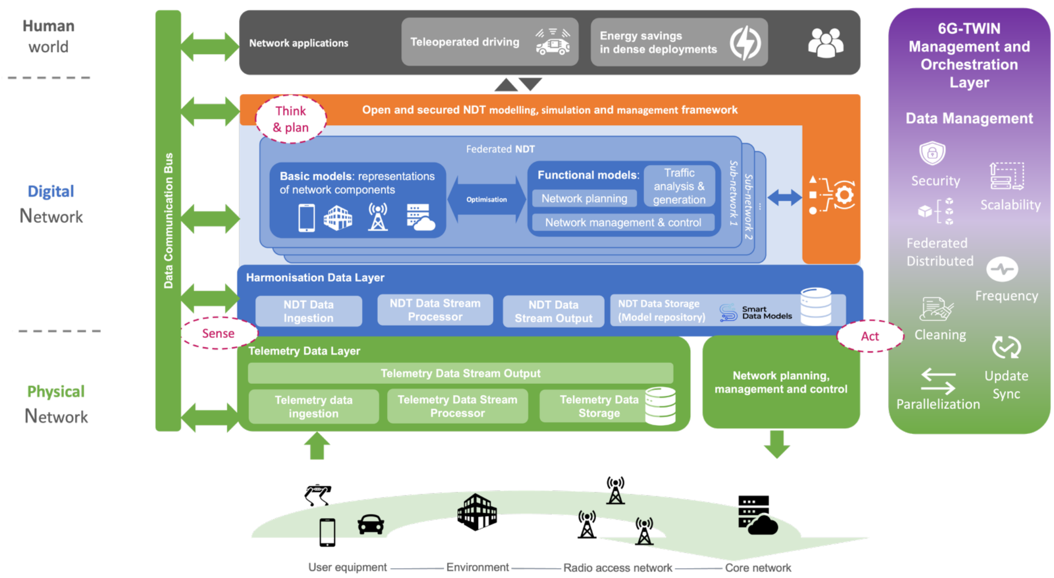

Figure 3. 6G-TWIN data collection and exposure framework.

2.4.2. Security and Privacy Aspects of Data Management

A data management framework within the NDT-enabled 6G architecture is examined in this subsection, focusing on securing data while preserving privacy throughout data-related processes. Besides, this effort includes the security and privacy aspects of the NDT MANO layer, emphasizing secure service LCM and access control mechanisms. Key aspects include:

- Identify vulnerabilities, threats, and potential attacks specific to data management processes.
- Analysis of countermeasures tailored to the identified threats.
- Examine the practicality and feasibility of existing countermeasures to ensure security and privacy within the NDT data management system.

While NDT integration with 6G networks offers advantages and new opportunities, it also expands the threat and attack surface by introducing new vulnerabilities. Therefore, potential attack and threat surfaces are categorized to address security and privacy challenges. This categorization assumes all virtual or physical elements through the NDT and 6G network as assets.

Security Aspects

The related threat surfaces are identified and described specifically to the 6G network with NDT capability to secure the data management processes, including the data collection and exposure points. The threats identified considering the data collection and management framework from Figure 3 are briefly described, with evaluations of their requirements regarding communication channels, associated protocols, components, and the corresponding NDT layer [37]:

- **Physical Infrastructure of the Network:** This section highlights the vulnerability of various 6G components, such as actuators, sensors, IoT devices, and data collection elements, to security breaches. Protecting these components is crucial because they play a vital role in supporting NDT functionalities within 6G networks. Given the necessity of real-time data collection for accurate simulations and real-world operations, securing this infrastructure is essential to maintaining the integrity of NDT-enabled 6G systems. Several potential attacks threaten this infrastructure, including: physical damage to hardware components, tampering with devices or data, information disclosure, leading to data leaks, unauthorized access to sensitive systems, or single point of failure vulnerabilities that can disrupt operations [38].
- **Communication Interfaces:** This description outlines the vulnerability of connections and exposure points involved in data exchange between the physical network, the NDT, and APIs. The communication interface plays a critical role in managing the flow of data from the physical network to the NDT, ensuring that the NDT accurately reflects the network's real-time state [39]. Additionally, it facilitates the transfer of data back from the NDT to the physical network, allowing for actions to be executed based on simulations and predictions generated by the NDT. Potential attacks that threaten this surface include: replay attacks that can disrupt data integrity, eavesdropping that leads to unauthorized data access, Denial of Service (DoS) attacks that can overwhelm network resources, Distributed Denial of Service (DDoS) attacks that amplify disruption, spoofing attempts that impersonate legitimate entities, man-in-the-middle attacks that intercept and alter communication. Addressing these threats is essential to safeguarding the communication interface and ensuring the secure and reliable operation of NDT-enabled 6G networks.
- **Application and Access Layer:** The user interface layer for the NDT, which connects to NDTs via application layer APIs to support various applications, offers stakeholders access to interfaces and simulation visualizations, rendering it one of the network's most exposed areas due to its direct interaction with users and external systems. It typically comprises user interface modules, visualization tools, API gateways, and application-specific components that access NDT functions. Communication within this layer commonly utilizes web-based interfaces and protocols accessible through internal or external networks [40]. Potential attacks targeting this layer include: backdoor installations that can provide unauthorized access, ransomware threats that

can encrypt and hold data hostage, API exploitation that can manipulate or disrupt NDT functionalities.

- **Synchronization Process:** This process is essential for maintaining accurate real-time synchronization between the virtual and physical components of the network, encompassing the timing, volume, and frequency of data flow between the physical network layer and the NDT layer. The synchronization process comprises two key elements: the synchronization between the physical network and its corresponding NDT, and dual synchronization across multiple NDTs. Potential threats to this synchronization process include: replay attacks that can inject outdated data, disrupting synchronization. Also, time delay issues that can hinder real-time accuracy and responsiveness.
- **Prediction Process:** AI/ML models and algorithms are integral to simulating, predicting, and optimizing network operations within the NDT. They process substantial volumes of data to forecast network activity, identify potential issues, and propose solutions. For these models to be reliable, the data used must be comprehensive, accurate, and reflective of real-world scenarios. Consequently, the datasets for training and validation, the computational infrastructure for processing, and the AI/ML algorithms themselves form critical components of this threat surface. Potential attacks targeting this area include: privacy attacks that compromise sensitive data used or generated by the models, model and data poisoning that manipulates algorithms or corrupts data to skew outcomes, white-box and black-box attacks that exploit model transparency or opacity to gain unauthorized insights or control.
- **Big Data Management Life Cycle:** The life cycle mentioned here could be categorized as four main stages: data collection, data storage, data processing and analysis, and knowledge extraction and creation [41]. These stages are susceptible to several security threats, including: impersonation attack for unauthorized users posing as legitimate entities to gain access to data, data breaching unauthorized access or exposure of sensitive information, malware injection introducing malicious software to corrupt or steal data, and identity theft stealing personal information to impersonate users and access data.

Privacy Aspects

Balancing security and privacy enhancements with overall system utility remains a key challenge. While providing security of the data management process it is also important to preserve the privacy. There are some countermeasures to protect the privacy throughout the system such as blockchain-integrated solutions, decentralized ML algorithms, blockchain integrated FL mechanisms, customizable encryption and security protocols and privacy enhancing solutions. For instance, in a multi-operator NDT setting, preserving privacy while sharing KPIs for collaborative optimization remains a critical concern. Techniques like federated learning may still leak information through model updates unless differential privacy is applied. Additionally, the use of distributed ledger technologies must account for the immutability of sensitive metadata, which may conflict with data minimization principles.

These solutions are proposed to address issues like data tampering and synchronization vulnerabilities, having an immutable data storage while enhancing transparency and security. Also, to avoid vulnerabilities in centralized data collection and model training, decentralized approaches like FL allow collaborative analysis without central server reliance. Blockchain integration to decentralized ML processes offer secure data transfers in decentralized learning environments. On the other side, protecting data privacy involves techniques like anonymization, encryption, differential privacy, and homomorphic encryption, which safeguard data while enabling real-time AI model use for network optimization. These measures aim to protect privacy while maintaining the utility of NDT-enabled 6G systems.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

2.5. AI training and workflow management

The yellow box in the above architecture (cf. Figure 1) represents the AI training block, which provides alternative pipelines for most functional models. This part of the system allows the models to learn from data, predict changes, make adjustments, and continuously improve their performance.

Notice that no explicit pipeline is needed for analytical and deterministic models. These models are represented by equations and algorithms with specific inputs and parameters and do not need a training stage before being saved in the model repository. As a result, they are stored directly in the **UDM repository** with suitable values for eventual static parameters so that they're ready to be invoked by the NDT management layer.

On the contrary, AI-based and Probabilistic Models (PMs) need a training stage to compute the weights of the underlying neural networks or the probability distribution so that they are ready to be used later at runtime.

Figure 4 below shows the training pipeline necessary to create functional models for the NDT. In this proposed pipeline, Supervised Deep Learning (DL), PMs, and RL methods are subject to this pipeline.

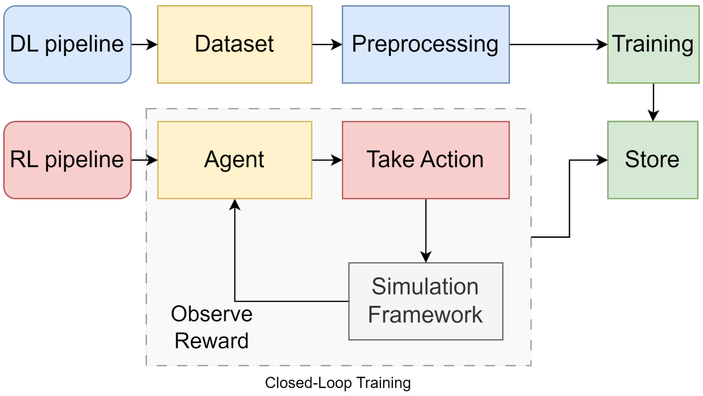

Figure 4. Training Pipeline.

The training process for DL and PM utilizes an input/output dataset, which can be created as a synthetic dataset by a simulation campaign within the simulation framework or gathered from the underlying real network. In contrast, in RL, the training process is trial-and-error based, and the environment's output is collected "on-the-fly" throughout the training, often from the simulation framework.

2.5.1. Deep learning pipeline

Preprocessing: In the DL pipeline, some preprocessing activities must be performed to ensure that the training converges as fast and accurately as possible with no deviations or overfitting. Data cleaning is the most crucial step, which could dramatically impact the training outcome. These data cleaning procedures mainly handle missing and outlier data points by replacing, deleting, or transforming them.

Feature selection is another important preprocessing step that removes unnecessary and redundant features and retains only the most crucial ones using a variety of approaches, such as wrapper-based approaches (forward, backward, and stepwise selection), filter-based

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

approaches (ANOVA, Pearson correlation, and variance thresholding), and embedded approaches (Lasso, Ridge, Decision Tree). In this stage, the features are not replaced by new features but reduced by selecting a subset of the features to reduce the dimensionality of the dataset. For example, the principal component analysis algorithm makes new synthetic data by linearly combining the original features and discarding the less important ones.

Training: Following preparation, the dataset passes to the training stage, where data for each model is divided into input and output features and prepared following the structure of the chosen neural network. Furthermore, a loss function to minimize, such as mean square error, mean absolute error for regression, or an entropy loss-based function for classification [42], will be selected to check the difference between the predicted outputs of a ML algorithm and the actual target values at each iteration and compute the new direction to adjust the weights using its gradient.

Store: Once the model has been trained, the final step is to preserve this work for future use. In the context of the NDT, the trained models' weights and important metadata, such as training conditions, model version, and performance metrics, are handed over to the NDT Management layer to support its operations concerning the models. All the data is then securely stored in in the **UDM repository**, ensuring it will be easily accessible and well-organized. Functional models may be updated when a new dataset is available or upon request.

2.5.2. Probabilistic models pipeline

Regarding PMs, a pipeline is needed to learn probability distributions from data according to the specific technique (Bayesian Networks, Markov Decision Process (MDP), Monte Carlo, etc.). In this case, the workflow is similar to the DL pipeline, since data needs to be pre-processed for cleaning and dimensionality reduction and then passed to the training stage. Finally, the resulting probability distributions along with the model metadata are stored the model in the Unified Data Model Repository.

2.5.3. Reinforcement learning pipeline

In contrast to DL models, RL does not use a pre-processed dataset for training purposes. The fundamental objective of RL is to enable an agent to learn optimal long-term decision-making strategies within a specific environment through an exploration/exploitation methodology. More specifically, the RL framework is formalized as an MDP comprising: (i) a state space representing observable environmental conditions, (ii) an action space defining possible agent interventions, (iii) a reward function that provides feedback to the agent after executing an action at a given time step, transitioning the environment from one state to another.

For environments with relatively low-dimensional state and action spaces, tabular Q-learning algorithms can effectively learn and store the values of the actions for each state. However, as dimensionality increases, employing neural networks to approximate the value function becomes computationally advantageous. Integrating neural network function approximators with RL algorithms constitutes Deep Reinforcement Learning (DRL).

Every RL or DRL training procedure necessitates an environment for agent interaction. When available, this environment can be an NDT instance; otherwise, it can be constructed upon the simulation framework by selecting appropriate simulators. During each training iteration, the environment processes the agents' actions according to an exploration or exploitation policy,

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

and the value function or Q-table is iteratively optimized based on the observed environmental states and rewards.

The training process systematically balances exploration (selecting new actions) and exploitation (leveraging existing knowledge). Initially, exploration is the primary approach to gain an extensive understanding of dynamic environments. As training progresses, the policy gradually shifts towards exploitation, prioritizing the actions with the highest expected returns. This shift can often be controlled through techniques such as Boltzmann exploration, ε-greedy policies, or more complex approaches, including curiosity-driven exploration and intrinsic motivation [43].

When DRL training concludes, the resultant Q-table or neural network weights representing the approximation value function are stored in the **UDM repository**, allowing for runtime deployment in which the trained policy can monitor current environmental conditions and conduct optimal actions based on learnt strategies.

The architectural distinction between DL and RL pipelines lies in their data acquisition mechanisms: DL leverages pre-existing datasets, while RL generates training data through environmental interaction. Section 4.3.2 provides a more detailed view of the DL and RL training pipelines, where we illustrate how the AI training component interacts with other NDT components in the functional architecture shown in Figure 1.

2.6. Simulation framework

In orange (cf. Figure 1), the DT simulation framework enables decision-making by coupling simulation models and simulators within the NDT architecture. Simulators such as OMNeT++, ns-3, VIAVI TeraVM RAN Scenario Generator (RSG), and MATLAB are software tools capable of running various simulation models. Simulation models abstract key network elements, including UEs, Next Generation Node Bs (gNBs), and CNs (that is, models in the traditional sense), as well as parameters like trajectories, transmission power, and protocols, all forming a complete scenario to be simulated. Simulation models are typically implemented through configuration files specific to each simulator.

To ensure flexibility, the simulation framework allows the integration of multiple simulators rather than restricting decision-making to a single tool. It enables the interoperability of simulation models across different platforms by facilitating data exchange between simulators at runtime, similar to Functional Mock-up Interface [44] and the High-Level Architecture [45]. This openness is reinforced by an open-source implementation, allowing developers to adapt the framework as needed.

Rather than supporting real-time simulation, which is often infeasible due to computational constraints, the framework focuses on scenario reproduction and what-if analysis, fitting the purposes of the NDT. Since AI models require extensive datasets often unavailable in real networks, synthetic data must be generated through simulation.

The **simulation framework** interaction works closely with the **NDT MANO** (cf. Section 2.3.1), which retrieves necessary models, configurations, and parameters from the **UDM repository and AI training module**. The simulation framework instantiates the simulators, manages their execution, collects results, and feeds them back into AI training and model optimization via the **NDT MANO**. The NDT then leverages the extracted insights to adjust real network parameters, ensuring continuous system improvement.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

6GSNS

2.6.1. Simulation management

The simulation management, lifecycle, and orchestration of simulators or federates will be done through a co-simulation framework or middleware. A federate is one of the autonomous components in a federated simulation, which is joined together forming a federation. Therefore, any simulator that is directly connected with the co-simulation framework is a federate; however, not all federates are necessarily a simulator, but a standalone component connected with the co-sim framework, forming a federation by being connected to a larger group of components (federated simulation system).

The **simulation framework** can be used for various purposes, including training RL models when data is unavailable from the **UDR**, generating synthetic data, and performing what-if analyses on the network. The co-Sim framework will act as an interface between the NDT (through the **NDT MANO**) and the federates. A gRPC-based interface (called *protocol-1*) will be developed to communicate between the **NDT MANO** and the **simulation framework**. The **NDT MANO** is responsible for telling the simulation framework what to simulate, and the co-Sim framework performs the simulation by orchestrating federates and managing their lifecycle. **Orchestration of the federates is the primary job of the simulation management, while all the simulation models are encapsulated within the federates**. Therefore, explicit instructions or simulation configurations must be passed from the **NDT MANO** to the simulation framework to perform the simulation. Hence, the **NDT MANO** can run the simulation and re-run it using a different configuration. The results from the co-Sim framework can be collected at the end of the simulation.

An important part of this project is to simulate a mobile communication network. The co-sim framework should orchestrate heterogeneous federates and a network federate/simulator to simulate a realistic heterogeneous scenario. Since there could be more than one alternative and people might choose one over the other depending on the use cases, we have decided to make the co-sim framework very generic such that any of the federates can be replaced by their potential alternatives using the generic *protocol-2* federate interface. For this project's reference implementation, we use OMNeT++ as the network simulator and SUMO as the traffic simulator. Of course, these can be switched with another simulator, e.g., ns-3, instead of OMNeT++.

Since network simulation will probably be performed using the simulation software OMNeT++ as simulator, on a high level, the **NDT MANO** will pass three pieces of information to the co-Sim framework: (i) the OMNeT++ simulation configuration, (ii) the names of other federates taking part in the federation, (iii) configuration or parameters of the respective simulators/federates. For the communication between the co-Sim framework and other simulators, a generic interface (called *protocol-2*) will be developed using gRPC to communicate with all the federates.

Now, the **NDT MANO** can use *protocol-1*, the launch control interface, to start a simulation through the co-sim framework. The **co-sim framework** will advance each federate's time and synchronize with its global clock time. The complexity of the models and the number of federates are directly related to the execution time of the simulation; hence, the simulation result is not readily available at runtime (or real-time). After performing a simulation, the result will be conveyed to the **NDT MANO** using *protocol-1*. These results can then be used for various purposes including storing, feeding to the AI optimizer or directly displaying to the end user as preconfigured by end user.

The **NDT MANO** is connected to a single instance of the **co-Sim framework** through *protocol-1* and can reuse this connection to run a simulation again. Figure 5 provides a high-level architecture of the **co-simulation framework**. **Deliverables D3.1 and D3.2 will give further details of the co-simulation framework.**

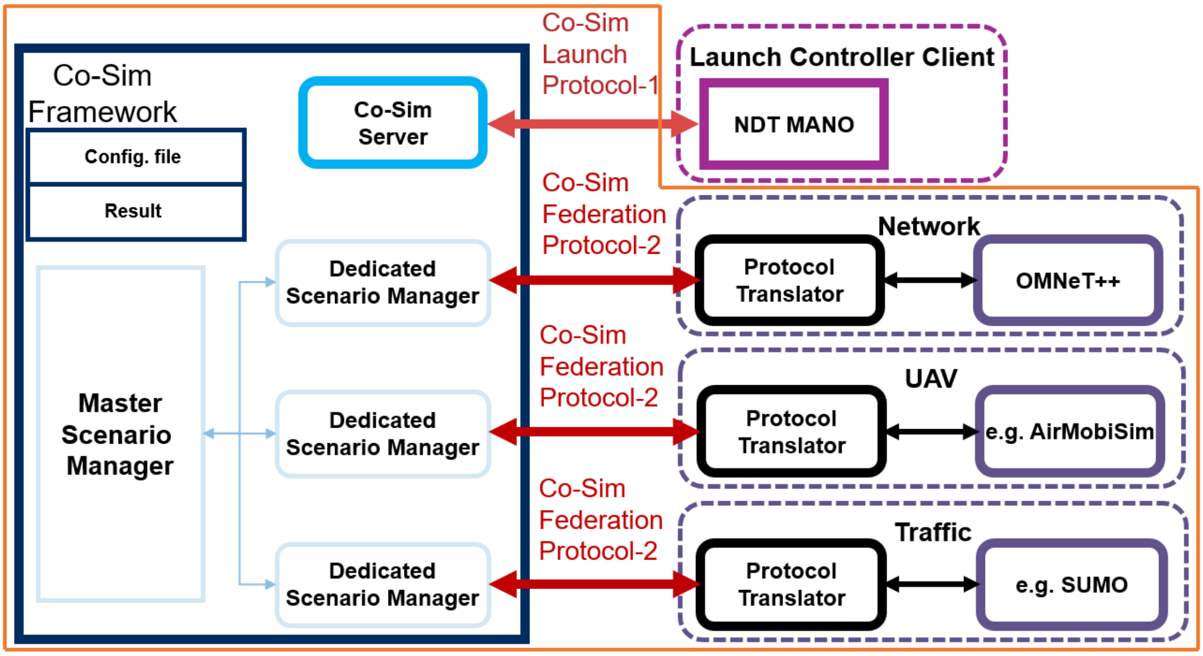

Figure 5. Co-Sim Framework architecture.

2.7. Unified dashboard

The architecture features a unified dashboard (grey) as an interactive interface for stakeholders to configure models, access functionalities, and monitor NDT operations in real-time. This centralized tool enhances network visualization and optimization. Additionally, it supports diverse use-case scenarios, demonstrating adaptability across applications. In our case, two key use cases illustrate its potential: a) teleoperated driving, which leverages real-time simulation and predictive analysis for low-latency, reliable remote vehicle operation, and b) energy savings in dense deployments, where the NDT optimizes resource management to reduce energy consumption.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

3. ZSM-enabled MANO for 6G-TWIN

3.1. Evolution of MANO: from NFV to AI-native ZSM architectures

The increase in the programmability of network slice architecture has raised the overall complexity of network service and orchestration mechanisms, requiring more operational agility from the network to respond to new services and applications enabled by the programmable nature of slices.

The primary objective of the **Network MANO/ZSM** layer is to automate the LCM of network services and operations, reducing reliance on human intervention and improving the responsiveness of device and service updates. This automation is driven by **real-time telemetry data** collected from physical systems and applications, which provides the necessary insight for continuous network monitoring and adaptive control.

This data is crucial, as the NDT is built upon using this data to diagnose problems that can be optimised, or anticipate "what-if" scenarios through **simulating close-to-real configurations of the network and physical devices**. The Network MANO/ZSM layer must focus on automating the collection, transmission, and analysis of telemetry data to benefit the physical environment in real-time, extending the concept of the DT towards the network.

Dedicated physical infrastructures are not sufficient for both the current 5G implementation [46], as well as the next generation of telecommunications. As expected, service provisioning must be flexible to support 6G applications. Through network slicing, an array of core NFs can be **tailored to the specific needs of a service**. This network slice is not bound to a particular physical infrastructure, allowing it to be virtualised on a heterogeneous network of devices where services are provided. This set of core NFs in the network slice is instantiated as Virtual Network Functions (VNFs), which are virtual representations of core NFs, and support the services being provided, allowing for the scaling of virtualised resources according to the client's needs.

In the context of 6G, the orchestration of network slices must be intelligently optimized to reduce service provisioning latency and enhance service continuity. As Virtual Network Functions (VNFs) evolve, they are expected to adopt a cloud-native approach, becoming Cloud-native Network Functions (CNFs). This shift leverages the microservice-based architecture of cloud deployments and the Service-Based Architecture (SBA) [47] introduced in 5G. Additionally, software-defined networking plays a key role by extending control and user plane separation [48], **enabling centralized and programmable network management through software-defined mechanisms**.

ZSM mechanisms must extend across the E2E network architecture, **integrating all management domains** (RAN, transport, core, edge, cloud) encompassing the physical network. CNFs are more flexibly deployed across these heterogeneous domains than VNFs [49], as they do not require an instantiation of a virtual machine on top of physical resources, allowing them to be more easily orchestrated as microservices, providing E2E LCM to network slices. **Automating this network support** and **reducing manual intervention** can boost the efficiency of the NDT. Both proactive problem-solving, in the case of simulations, and reactive problem-solving, in the case of real-time optimisation, are some of the main roles the 6G-TWIN NDT must address.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Within the 6G-TWIN architecture, the Network MANO/ZSM layer serves as a pivotal **orchestrator of telemetry data**, with **automation mechanisms** and **control loops** that support the connection between the physical world and its DT, as well as enabling the network applications that the NDT is supporting. In the physical world, the Network MANO/ZSM layer interacts with the **Data Collection Framework** outlined in D1.2 [2]: a flexible approach to collect, process and distribute both telemetry and device data, over a wide range of network components, from both proprietary and legacy infrastructure.

The design for this data collection framework is focused on efficient data management, pre-processing and synchronisation across digital and physical network element, with the Network MANO/ZSM layer acting as a bridge to make sure the data from this wide range of devices is harmonised, so that it can be sent to the digital world to be stored on the **UDR**. This **UDR** was introduced in D2.1 [50] to implement a common data space for the NDT. Data is uniformly collected, processed, and harmonised in the physical world across multiple devices to increase the efficiency of the NDT, as working with a standardised data format allows for automation mechanisms and control loops to automatically handle dataset utilisation by the **AI Training block**, as well as the **Simulation Framework**, within the NDT.

3.2. Network MANO/ZSM automation mechanisms

The definition of control loops can help the network by defining a cycle of monitoring, analysis, decision-making, and action operations, contributing to the LCM requirements mentioned previously. These control loops are critical for maintaining real-time performance adjustments, optimising resource allocation, and ensuring both service reliability and security. **Closed-loop automation** (CLA) [51] is an approach that fully automates this process. CLA draws inspiration from prominent reference control models for autonomic and self-adaptive systems, such as the Monitor-Analyse-Plan-Execute over a shared Knowledge (MAPE-K) [52], the Observe - Orient – Decide – Act (OODA) model [53] and FOCALE [54]. Despite their differences, they are all composed of four main phases: gathering data (Monitor), analysing this data (Analyse) and reaching a specific decision (Plan) on which to apply to a particular device or system (Execute), all without human intervention and Knowledge of each step and component.

Another way to implement closed-loop management is through an intent-based approach, or **intent-driven closed loops** [55]. Feeding the network a high-level objective, or “intent”, as a desired outcome to achieve, and then having the network translate this intent into specific policies/actions, it is possible to automate operations, by continuously checking whether network behaviour matches the provided intent and adjusting the configuration of devices whenever this intent is violated. A similar approach can be used in the case of the NDT, as the NDT should **translate a specific intent** proceeding from the human world into a predefined set of desired outcomes, such as performance targets or security rules. Continuously monitoring the network (or the NDT) for performance metrics, comparing the current state with the intended state, triggering changes on the network (or its DT) according to detected deviations, revalidating the applied configuration, and restarting the control loop.

While implementing control loops and intent-based automation mechanisms is a programmable way to improve network management, AI and ML [56] can play a role in **developing and automating zero-touch network management mechanisms**, turning real data collected from the network topology and devices into intelligent decisions. Both AI and NDT systems take advantage of massive real-time data collection from network elements and users, allowing an ML model to train on the data collected [57] and integrate AI to make a prediction based on this output. AI/ML can also play a role in mapping high-level intents to specific network policies, as well as defining and validating specific decisions for closed loops;

each decision should have specific feedback, for the model to improve its decision-making process automatically, allowing for continuous model training on new data.

3.2.1. Data and telemetry collection

One of the main goals in defining future 6G systems is guaranteeing standardization and interoperability between the different parts of the network [58], **managing and coordinating multiple subsystems** from the CN to RAN, TN, Edge, and Cloud, unified in a single control plane. The 6G-TWIN proposal of an AI-native, NDT-enabled network architecture seeks to monitor these multiple domains across the physical and digital network, using AI-based methods and control loops to support the lifecycle of its components and decision making.

ETSI's ZSM framework reference architecture [15] details a modular, service-based architecture that oversees a management domain for each specific network subsystem, providing services through standardized endpoints and APIs. A global E2E Service Management domain sits above them, as explained in Section 2.2.2, consuming these domain-specific services for slice LCM and network service orchestration. This orchestration can be applied on all domains, by handling **domain-specific data collection** that interact with **domain-specific intelligence services** (using collected data to support the decision-making process of control loops for CLA) and **domain-specific analytics** (providing insights into the deployed services and physical infrastructure and generating predictions on the behaviour of these same services and infrastructure).

Through the domain-specific data collection services, the NDT has access to the respective telemetry data. This data provides real-time feedback of the physical components of the network, helping the NDT to synchronise with the current state of the network, improving the analysis and decision-making process, such as:

- **RAN:** In the case of gNBs, important radio Key Performance Indicators (KPIs) should be collected, such as Bit Error Rate, throughput, delay, handover performance, and configuration states.
- **TN:** Resource utilisation, latency, jitter, and other statistics can be collected from network devices like Routers and Switches.
- **CN**: deployed NFs can emit control-plane and data-plane-specific network slice 5G E2E KPIs [59].
- **Edge/Cloud:** computation nodes, both located along the edge or centralised in cloud servers, can be monitored for resource usage (CPU, memory, energy consumption) and orchestration details (application/service deployment health).

Our **Data Collection Framework** envisages **telemetry pipelines that ingest, normalize, and expose data from multiple domains**. This integration is possible through model-driven interfaces (YANG, gNMI, REST, TOSCA) that abstract from different sources to enable real-time and batch data flows [60]. Telemetry collection is subscription-based and adaptable to devices with varying specification types. It uses specific intent-driven policies to determine the frequency of updates to optimise network traffic between source and NDT; this avoids potential overloads on the side of devices or networks without requiring dedicated infrastructure.

The **Telemetry Data Collector** is also crucial for outputting specific metrics, such as E2E latency, jitter, packet loss, and even device power consumption. Some of these provide important **information about the device status** for the NDT in specific use-case applications, such as the ones highlighted in D5.1 [61] in the context of 6G-TWIN. This allows for the correlation between metric-powered closed-loop actions, **triggering automated responses** from the NDT, like optimizing traffic flow between the device and NDT or a "what-if" analysis for future device configurations.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

3.2.2. Interfacing with the physical network

The physical network can comprise a wide range of heterogeneous devices, increasing the different types of communication protocols, operating systems, and capabilities the Network MANO/ZSM layer must interface with to integrate them into the NDT. **Service Level Agreements** (SLAs) [62] are formal contracts between service providers and customers that **define a certain level of service**. Establishing SLAs with physical devices simplifies interoperability while ensuring consistent performance and measurable and enforceable service management.

The NDT enables predictive **SLA assurance**, or the process of ensuring the service supported by the NDT meets the performance and quality standards outlined by the provider. The NDT combines telemetry data with AI analytics to fulfil this purpose, enforced through policy-based control and automated prediction triggers. By translating SLA objectives into intents (such as specific latency targets for device communications), these can be measured and acted upon across all domains. The telemetry pipelines set up by the NDT feed relevant KPI metrics, allowing for control loops to **detect at the domain and E2E levels any SLA deviations or anomalies**, guided by specific threshold rules, and feeding observations back into the NDT for further analysis and state updates [63].

By taking advantage of the NDT capabilities, this analysis covers both real-time SLA assurance as well as SLA breach forecast. The collected observations from the devices feed **specific AI/ML models that can estimate and predict** SLA breach trends (such as increasing throughput on specific timeframes to improve latency). These models can also provide key fault management capabilities, working backwards in the prediction timeframe to perform **root-cause analysis** across domains, isolating the cause to a specific SLA violation. Once these SLA violations are identified, intent-aligned policies trigger automated loops by scaling the RAN slice, increasing available resources, migrating specific workloads, etc., via secure domain APIs [64].

To integrate multiple domains, this open telemetry framework must be ruled by zero-trust and federated architectures that seek to provide functionalities of the NDT, without compromising its security. Strict governance rules, trust relationship requirements [65], and a **Threat and Risk Assessment** (TRA) [66] framework are mechanisms explored by ETSI to address these issues. Firstly, the ZSM framework must support the capability to **evaluate the trustworthiness** of ZSM entities across all management domains, allowing also for the **re-evaluation** of the trustworthiness of specific ZSM entities and **rebuilding trust models** that were previously broken, according to the predetermined trust relationship requirements of managed domains, services, and functions.

TRA defines a framework for **analysing the risk of the ZSM framework,** as shown in Figure 6 by identifying the ZSM assets (such as managed services and data), relevant threats to these assets, and vulnerabilities to the system. In doing so, existing control loops are updated continuously with new security controls to prevent current and future threats. The system will also analyse itself continuously for the likelihood and severity of new threats, reporting this information for self-improvement on future events.

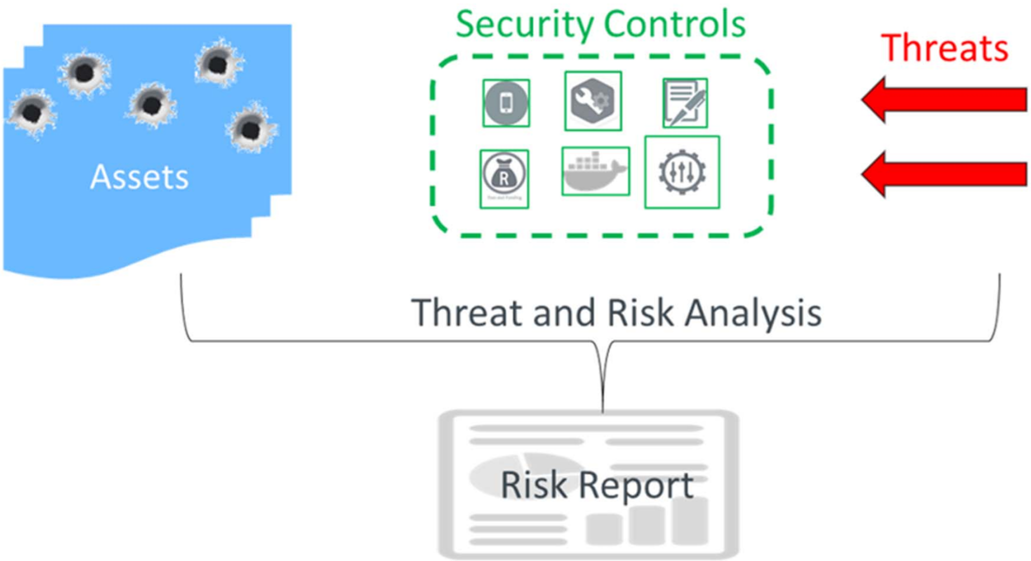

Figure 6. ETSI ZSM threat and risk analysis framework [66].

3.2.3. NDT End-to-End management

The ETSI ZSM architecture [15] is designed to enforce a separation between two important network management tasks: **domain-specific management** and **cross-domain E2E service management**. While each domain is independent, they expose data through standardised interfaces. This simplifies data collection, orchestration, and control mechanisms across multiple devices, allowing for better interoperability between network services and automation of scaling procedures through intelligent, federated coordination via the **F-MANO** layer.

This is also supported by common data services, decoupling operational data from control logic, which enables the NDT to access telemetry and analytics data for E2E decision-making without disrupting real-time data collection. These decision-making processes are intent-based, translated from high-level goals, policies, and constraints associated with the application/service provided, forming **closed-loop autonomous operations** bounded by governance policies [67].

E2E slicing in ZSM is handled in the E2E service management domain, which coordinates multiple network slice deployments across RAN, Transport, Core, and Edge domains. In the 6G-TWIN architecture, the F-MANO interfaces with the network applications to determine the necessary SLAs and high-level intents, while the ZSM layer enforces them as a policy engine role, on the physical network, featuring:

- Feasibility checks against resource and service constraints.
- Inventory services for maintaining logical and physical topology.
- Slice LCM, such as onboarding, instantiation, scaling and healing.

These tasks are supported by the ZSM layer, decomposing **slice-level intent** (for instance, a network slice with <1 ms latency) into **domain-specific allocations** (RAN CQI thresholds, Core Control Plane bandwidth, Edge compute SLAs). The ZSM invokes the domain-level specific orchestrators via model-driven APIs to fulfil these slice specifications, tracking progress through real-time telemetry updates.

3.2.4. AI/ML model management

Control-Loop Governance was formalised by ETSI [51], [65, p. 2] as a framework for creating, coordinating, and scaling CLA across network domains. The **MAPE-K** loop offers a baseline for designing and controlling governance loops, supporting the lifecycle through operator policies and priorities. This ensures consistency across different closed loops (through specific operator policies), handles conflicts (through E2E management loops that trigger domain-specific loops, offloading management responsibilities), and escalation (domain-specific loops can request higher-level intervention). As such, domain loops optimise **local service fulfilment**, while E2E loops maintain **slice-level SLAs** and adjust multiple domains as needed; in the case of the 6G-TWIN architecture, domain loops and escalation triggers are maintained by the Network MANO, while support to higher-level E2E loops is attributed to the F-MANO block.

From this, it is possible to derive that these closed loops maintain their functionalities through the intent-driven nature of the ZSM: operators must **declare specific outcomes** for the application/service instance for each slice (e.g., performance, latency, reliability) instead of directly issuing configuration procedures. The ZSM will receive these intents from the MANO instantiation and **translate them into domain-level goals**, which are maintained by specific closed-loop mechanisms [67].

In the case of the 6G-TWIN architecture, the NDT MANO will only assume intent owner responsibilities, while the Network MANO/ZSM will share both intent owner and intent handler capabilities, interfacing with the NDT MANO for domain-specific and slice-specific intents. Both network management and device management tasks are supported by intent-driven closed-loops, such as **federation** (the Network MANO/ZSM interfaces with devices, referring to all operational requirements, goals and constraints of the specific application/service from the intent owner of the NDT MANO federation), **intent translation** (Network MANO/ZSM will take the high-level intents from the NDT MANO and decompose them into specific operational and management intents for each specific domain and application/service) with conflict management, and even **intent lifecycle** (managing the lifecycle of a given intent object instance from the intent owner with a dedicated intent handler for each dedicated domain).
Besides intent-based mechanisms, AI/ML also plays an important role in zero-touch automation in network management, from high-level slice orchestration to low-level physical device control. AI/ML can be **deployed as an autonomous operational agent**, residing as both domain-specific management functions (RAN, Core, Transport, Edge) and the E2E orchestration layer. They are operationally similar: to achieve **intelligent CLA** by observing, learning, and acting without human intervention. They can be deployed as AI/ML models as a lightweight NF (VNF/CNF, Kubernetes pod, etc.).

In the case of LCM, the Network MANO/ZSM layer should be implemented in an AI-native way [15]: as such, these services are first-class entities of the ZSM framework, supporting MLOps within the Network MANO/ZSM layer and **ensuring automation logic evolves according to the current network state**, employing functionalities such as:

- **AI Model Management service**, for managing the lifecycle of AI models (training, retraining, performance evaluation).
- **Training Data Management**, with data preparation pipeline for preprocessing the collected data (data cleaning, data harmonisation) and distributing model training across edge and core domains (distributed and FL), for both the real-time and batch data cases.
- **Model Assessment**, monitoring the accuracy and consistency of the active AI/ML models under the NDT-surveyed domains, triggering automated pipelines to retrain or switch models according to model inference quality values.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Another approach is for the 6G-TWIN Network MANO/ZSM layer to deploy slice-specific AI agents [68]. These AI agents improve on traditional AI orchestration tools by fully embracing the zero-touch approach: AI Agent orchestration has a **network of specific AI agents**, designed for particular tasks, which **autonomously make decisions based on their pre-defined goal** and domain-specific circumstances. By optimizing to a specific function, the AI Agent orchestration framework can deploy these agents for the demands of specific domains, ensuring the right agent is assigned to the right task and improving overall system performance. These agents can also collaborate, in a **multi-agent system**, to solve complex problems more efficiently than a single AI agent could. While early examples of this approach are already explored in the state of the art [69], [70], for the case of the 6G-TWIN framework, the Network MANO/ZSM would have access to a **Core AI Agent pool**, and for the instantiation of a domain specific slice, a dedicated set of agents would be deployed according to slice specifications, enhancing operational efficiency and improving the decision-making accuracy of the AI models.

3.3. Network MANO/ZSM lifecycle operations

This section explores different examples resulting from Network MANO/ZSM operations in the 6G-TWIN architecture, focusing on how NDT-enabled simulation and intelligent learning mechanisms can proactively identify, analyse, and resolve issues across multiple network stack layers. Firstly, an approach to **automate NDT LCM mechanisms by using generative AI models**; secondly, a **challenge-based FL scenario**, where the NDT serves as a pre-training stimulus to assess client robustness under simulated failure condition; thirdly, an **automated application-level issue resolution**, showcasing how NDT frameworks can detect anomalies, simulate corrective actions, and autonomously deploy solutions to maintain stringent Quality of Service (QoS) requirements for latency-sensitive applications; and finally, a **network-layer triggering and functional model-based optimization for handover management**, emphasizing how the NDT system leverages functional models and tailored simulations to adapt handover strategies and resolve connectivity challenges, especially in high-mobility or congested scenarios.

3.3.1. Beyond traditional methods: unlocking the power of generative AI for NDT LCM automation

Achieving a high level of network autonomy requires moving beyond the current reliance on human expert knowledge, which remains largely manual and problem-specific. Currently, selecting AI models, tuning hyperparameters, and configuring training datasets are tasks that heavily depend on domain expertise and are not easily generalizable across different use cases. To overcome this limitation, an intelligent and adaptive system must be introduced within the NDT management entity of the 6G-TWIN architecture, capable of automating these decisions in a scalable and context-aware manner. This shift is essential to enable self-optimizing, AI-native networks that can efficiently manage the growing complexity and variability of future 6G environments.

A new research direction is emerging, moving toward creating network foundational models, i.e., large models trained on massive and heterogeneous datasets to solve many downstream tasks [71]. Recent efforts apply existing Large Language Models (LLMs) that were designed for natural language processing applications, like GPT-x, LLaMA, and Falcon, for wireless networks optimization [72], [73]. Other works propose wireless-centric foundation models that go beyond applying LLMs directly to solve network optimization problems. For instance, the framework in [74] uses multi-modal data fusion, grounding, and instructibility.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

One known limitation of LLMs is the hallucination problem [75]. Although reasoning techniques have been widely used to mitigate this issue, they do not fully eliminate it [76]. Within the 6G-TWIN architecture, integrating LLM-based solutions into the NDT presents an opportunity to address this challenge: the NDT can act as a sandbox or validation layer where generated outputs are tested within a simulation environment. This allows for comparison against established baselines or among multiple generated solutions to select the most reliable one, thereby reducing the risk of acting on hallucinated outputs. This approach is currently under investigation in 6G-TWIN, as we explore the potential of generative AI to enable secure, efficient, and autonomous NDTs, paving the way for truly autonomous networks. Our findings in this area will be presented in the final version of the deliverable.

3.3.2. Challenge-based federated learning scenario for NDT-aided O-RAN

In the context of increasingly complex and heterogeneous 6G networks, ensuring application-level reliability and adaptability requires proactive and intelligent decision-making mechanisms. Integrating NDT technology as an application-layer component (rApp) within the O-RAN architecture presents a promising solution. By simulating potential failure scenarios and generating dynamic what-if challenges before FL begins, the NDT acts as a trigger for informed participation in the learning process. This approach increases the resilience of the application layer against unexpected behaviours and aligns well with the goals of self-organizing networks in 6G, where automated, real-time adaptation is essential. Furthermore, it has the potential to support automated issue resolution by using the feedback loop created through FL clients' reactions to these simulated scenarios. Only those clients that meet predefined performance thresholds during the challenge phase proceed to the model aggregation step, ensuring robust and fair model training while mitigating the risks of free-riding. From the application layer's perspective, this facilitates a policy-driven and privacy-aware framework capable of resolving potential issues even before they manifest in real deployments. The proposed methodology, therefore, bridges the gap between simulation and real-time application management in multi-vendor environments, contributing to the agility and dependability of future 6G systems.

3.3.3. Automated application-level issue resolution in 6G using NDT technology

Applications are expected to operate under dynamic and demanding constraints within 6G networks. These constraints vary depending on the application's nature, requiring specific network QoS and resource provisioning levels. Ensuring optimal performance under such conditions is challenging, especially when deviations from **normal behaviour** occur.
Let us now consider a concrete application scenario in the domain of **teleoperated driving**, specifically, the **path planning** module, which must operate while meeting strict QoS requirements. In such a use case, a RL-based application can be deployed as a *functional model* within the 6G-TWIN framework to optimize both path selection and network quality. The RL agent aims to minimize connection loss while ensuring efficient traffic routing.

In Figure 7, we represent this use case, using the NDT to minimize the allocated resources for data processing, offloading the necessary computational requirements from the vehicle to nearby edge servers, thereby reducing network congestion. The NDT MANO instructs the simulation framework to perform an analysis of the network configuration based on the current status of the scenario, described through basic and functional models. Based on the simulation framework output, this result is reported to the NDT MANO, which will evaluate if enough edge computing resources are available along the planned route. If any issues are identified, more resources are made available towards the simulation scenario until a simulation is successful. The optimal configuration is then applied to the relevant controllers via the Network MANO/ZSM.

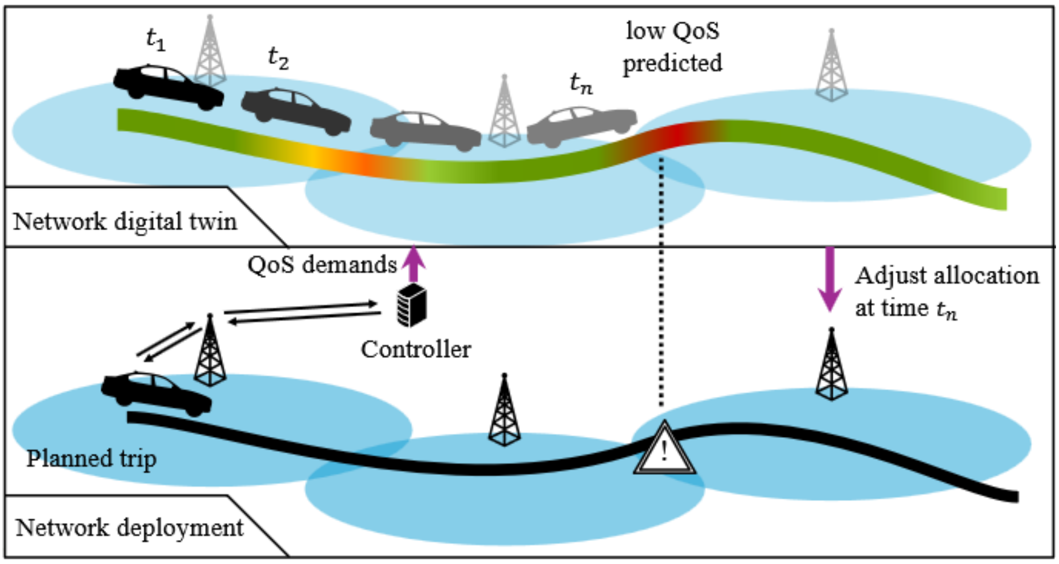

Figure 7. Teleoperated driving use case [3].

However, mobility environments are inherently dynamic, and RL-based algorithms may underperform in edge cases. For instance, performance can degrade in neighbourhoods with poor network coverage or during peak traffic hours when network demand surges. These scenarios highlight the need for an intelligent and adaptive problem-resolution mechanism.

This is where the ***trigger mechanism*** comes into play. An anomaly detection algorithm, also deployed within the NDT, continuously monitors the application's behaviour. An automated *improvement request* is generated upon identifying deviations from expected performance, such as a significant drop in QoS or increased latency.

This *improvement request* initiates a tailored simulation for the specific edge case using the NDT platform. Relevant application data is retrieved from the **UDR**, and the simulation is instantiated with the appropriate scenario parameters.

To resolve **connectivity issues**, the simulation may explore various **network configurations,** such as increasing the density of terrestrial base stations or deploying aerial base stations during high-demand periods. If a suitable network configuration is found, it is validated within the simulation environment. Once validated, the configuration is automatically deployed in the live network, effectively resolving the QoS shortcomings.

In cases where the issue stems not from the network but from **limitations in the RL-based functional model**, the NDT platform enables rapid iteration. The functional model is retrained or adapted with the help of the simulation environment, validated against real-world data, and automatically deployed into the production environment upon meeting performance criteria.

This **E2E automated loop**, from anomaly detection at the application layer to intelligent resolution via simulation and model adaptation, highlights the powerful synergy between 6G physical networks, NDT platforms, and ZSM. It ensures resilient and adaptive service performance in complex, real-world scenarios without the need for manual intervention.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

3.3.4. Network-layer triggering and functional model-based optimisation for handover management

In addition to application-level triggers, the 6G-TWIN NDT platform supports proactive **network-layer optimization**, particularly for critical operations like handover management. The ZSM system, with its real-time global perspective of the network and connected devices, is well-equipped to detect performance degradation. These can be identified based on predefined QoS thresholds, SLA violations, or advanced AI-based anomaly detection algorithms analysing metrics such as handover failure rates, fluctuating signal strength, or declining user throughput.

Yet, **a handover strategy directly and significantly impacts overall network performance**. Therefore, when the ZSM system identifies a persistent degradation in service quality or failure to meet SLAs, it can **trigger** an automated *improvement request*. This request initiates a resolution workflow similar to that used in application layer optimization.

To address the problem, the RL-based functional model responsible for managing handovers is evaluated within a tailored simulation environment. This simulation aims to identify the degradation's root cause and test possible corrective strategies. Predefined scenarios like high user mobility, traffic congestion, signal interference, or physical obstructions are simulated to mirror real-world conditions.

Similar corrective actions can be explored, as in the previous use case involving teleoperated driving. For example, if the degradation is due to **external factors** such as rush-hour traffic or weak area coverage, then network-side adjustments such as deploying additional base stations or optimizing antenna parameters can be simulated and validated. If network-level modifications fail to resolve the issue, this points to a potential shortcoming in the RL-based handover model itself. The model can be **retrained** with simulations using updated datasets and environmental variables in such cases. Once it meets the desired **SLA and QoS** benchmarks, the improved model is validated and deployed in the live network, ensuring sustained performance without manual intervention.

This **closed-loop**, automated feedback system exemplifies how the 6G-TWIN NDT platform leverages both network-layer intelligence and functional model adaptation to maintain service continuity and optimize user experience in complex and dynamic environments.

3.4. Functional and non-functional requirements

Finally, after listing all technologies for zero-touch automation of the 6G-TWIN NDT, it is possible to map them to the Functional and Non-Functional Requirements (FRs and NFRs) outlined in D1.1 [77]. This ensures that the 6G-TWIN architecture is able to achieve its intended purpose, tackling the technical challenges related to the automation of NDT mechanisms in 6G environments. While previously only a list of requirements was outlined, a more complete vision for the 6G-TWIN NDT is available from this deliverable onwards, described in the table below.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Table 2. Functional and Non-functional requirements for ZSM across the NDT

ID	Requirements	How its addressed
FR.ZSM.01	The ZSM shall allow for the implementation of a fully automated network management system where the different data points can be integrated.	Intent-based E2E and Domain-specific orchestration
FR.ZSM.02	The ZSM shall allow for the autonomous and automatic monitoring of the real-time status of the NDT.	Telemetry pipeline data collection
FR.ZSM.03	The ZSM shall enable the MANO of network resources to allow the AI functions on the NDT to optimise and dynamically allocate network resources.	E2E slice orchestration
FR.ZSM.04	The ZSM service shall have standard APIs to ensure the minimum interoperability requirements between the new and existing services.	Cross-domain E2E service management
FR.ZSM.05	The ZSM service must enable the integration of AI-based Network Functions (NF) and Network Services (NS), following the best practices for continuous integration and continuous deployment and automation while promoting the correct deployment and placement of the workloads.	Intent-based programmable ZSM + MLOps
FR.ZSM.05	The ZSM service should support programmable interfaces that AI-based functions and the NDT ecosystem of apps can control.	Cross-domain E2E service management
FR.ZSM.06	The ZSM system must be able to protect its APIs and resources with the necessary authorization tokens. All the AI-powered functions shall communicate with the ZSM service in a secure and authorised manner to enable security, auditing, and reliability across the network.	Threat and Risk Assessment Framework
FR.ZSM.07	The ZSM system must be able to protect its APIs and resources with the necessary authorization tokens. All of the AI-powered functions shall communicate with the ZSM service in a secure and authorised manner to enable security, auditing, and reliability across the network.	Threat and Risk Assessment Framework
FR.ZSM.08	The ZSM service shall allow itself to be discovered in the network. Because of its appearance on the network, it shall be able to register itself on a common control plane (e.g., the NDT control plane).	Interface with NDT MANO federation enabler
NFR.ZSM.01	6G-TWIN KPI1.1 Provide a federated and AI-native network reference architecture that integrates multiple NDTs for real-time data analytics and decision-making across at least three network domains.	Cross-domain AI-native ZSM layer
NFR.ZSM.02	6G-TWIN KPI2.5 Support the integration of Network planning & what-if analysis, Network management and control, Network traffic analysis operations.	Telemetry Data Collector + SLA assurance
NFR.ZSM.03	The ZSM service shall be able to handle complex networks across multiple domains and time scales (scalability).	E2E network slice and service orchestration
NFR.ZSM.04	The ZSM service shall be able to handle failures and have failover mechanisms (reliability)	Fault management through SLA assurance

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

NFR.ZSM.05	The ZSM service shall be able to ensure low latency and be able to handle AI-based workloads (performance)	Slice-specific SLA assurance + AI Agent Orchestration
NFR.ZSM.06	The ZSM service shall adhere to the NDT requirements around security and privacy to enable a compliant environment and network (Security and privacy).	Threat and Risk Assessment Framework
NFR.ZSM.07	All of the ZSM APIs and interfaces shall be interoperable and standard (Interoperability)	Cross-domain E2E service management
NFR.ZSM.08	The ZSM service shall have an energy-aware mode. (Efficiency)	While not fully explored, an energy-aware mode could be implemented through enforcing a set of SLAs across active slices on network domains
NFR.ZSM.09	The ZSM service shall be able to be adapted and maintained by the NDT AI-based functions and the DevOps tooling (Adaptability and Maintainability)	Intent-based programmable ZSM + MLOps

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

4. NDT internal operations

The previous section envisions managing and orchestrating a 6G AI-native network that integrates with NDTs. In other words, it assumes that the NDT has already been created and can be utilized in network operations and processes. In this section, we consider that such a tool has not yet been developed and introduce the necessary elements to build an NDT, along with the procedures for creating it. It is worth mentioning that the ZSM principles discussed in Section 3 are equally applied here, although not fully described, i.e., we aim for a more general approach that assumes no particular tool for implementing the automation in creating the NDT.

4.1. Functional and non-functional requirements

When creating a software tool, one of the first things to do is to list the FRs and NFRs. This phase is pivotal, as it lays the foundation for aligning the NDT with the needs of users, stakeholders, and the technical challenges posed by 6G environments. D1.1 [77] presented most of the FR and NFR for the blocks in the 6G-TWIN MANO layer presented in Figure 1. However, up to now, the FR and NFR for building an NDT have not yet been presented. Table 3 presents such FR and NFR.

Table 3. Functional and Non-Functional Requirements for Network Digital Twins.

ID Requirement	Requirement Name	Description	Rationale
FR.NDT.01	NDT mapping	An NDT should be able to model and represent the physical network with high fidelity.	This is the fundamental capability of an NDT. For doing this, the NDT should be able to build accurate network models that can operate in real-time, to represent network elements and topologies and models for network analysis, prediction and assurance.
FR.NDT.02	NDT interaction	The NDT should interact in real-time with the physical network. Appropriate interfaces are needed for doing this.	An NDT is required to support open and standard southbound and northbound interfaces, to help avoid either hardware or software vendor lock and achieve interoperability. Southbound interfaces of NDT are responsible for information exchange between the PT and an NDT. Northbound interfaces of NDT are responsible for information exchange between NDTs and network applications.
FR.NDT.03	NDT modelling	The NDT should support basic and functional models for network applications in the form of unified data models.	The modelling component of a NDT must be capable of representing network elements and topology through basic models that include detailed maps, network nodes, UE behaviour, channel characteristics, and network slice configurations. Beyond these basic representations, the system should implement functional models that support a wide range of network operations such as analysis, emulation, diagnosis, prediction, and assurance. These models must encompass capabilities like traffic generation, mobility simulation, RAN management and optimization, as well as exploitation functions including

			monitoring, KPI reporting, data analysis, and visualization.
FR.NDT.04	NDT Data	The NDT should support an efficient and unified data repository.	To support the operation of a NDT, it is essential to collect and store a wide range of network data from the physical infrastructure, including configurations, operational states, topology information, traffic traces, and KPIs. This encompasses both real-time data for accurate, up-to-date representation and historical data for deeper analysis. The collected raw data must undergo processing steps such as cleaning, fusing data from multiple sources and scales, and applying dimensionality reduction techniques to enhance the robustness and reliability of the information.
FR.NDT.05	NDT MANO	The NDT MANO block should be able to define, instantiate, update, and delete DT instances.	The NDT management system should be able to interpret the requirements from external users to create NDT instances by identifying the models and the data that should be used for this. After a NDT instance is created, the NDT management system should maintain such instances including monitoring of their resources and accuracy, and deletion after the service is no longer needed.
FR.NDT.06	NDT AI/ML	The NDT should support the integration with AI/ML models.	The NDT serves two main purposes. One is the ability to operate as a platform for AI training and inference prior deployment on the physical network, with a particular emphasis on enabling DRL-based closed-loop control mechanisms. It supports the full AI/ML lifecycle, encompassing the training, testing, and operational deployment of models within the NDT environment. The second purpose is to employ advanced AI/ML techniques, to perform tasks such as anomaly detection, network optimization, and intelligent decision-making, thereby enhancing the automation, efficiency, and resilience of network operations.
FR.NDT.07	NDT modes	The NDT should be able to work as an analytical NDT or a controlling NDT	An NDT can serve various purposes, broadly categorized into analytical and controlling functions, which dictate their design, interaction with the physical network, and overall objectives. An analytical NDT primarily focuses on understanding, evaluating, diagnosing, and predicting network behaviour and performance without directly enacting changes on the physical network. A controlling NDT goes beyond analysis to directly influence, manage, and optimize the physical network, often through a closed-loop system.
FR.NDT.08	NDT Federation	An NDT instance should be able to work together with other	Similar to composing network services, a bigger NDT instance can be formed by aggregating two or more smaller NDT

		NDT instances in a federated manner.	instances. When those instances belong to different administrative domains, a federation must be established to avoid any conflicting policy, e.g., data access and usage policies.
NFR.NDT.01	NDT Scalability	The NDT should support the modelling of large, distributed networks, and managing high data volumes and device counts, including lightweight twin instances for edge computing.	Ensures the NDT can handle future network growth and operate efficiently across a variety of deployment environments accommodating the growth and complexity expected in 6G and large-scale deployments like IoT and smart cities.
NFR.NDT.02	NDT Performance	The NDT should enable low-latency processing, real-time responsiveness, and high-throughput data ingestion; ensures fast operation and response to network changes.	Critical for time-sensitive use cases (e.g., ultra-reliable low latency communication) and maintaining accurate and timely DT representations.
NFR.NDT.03	NDT reliability and availability	The NDT should ensure fault-tolerant data collection and model execution with redundancy and high availability mechanisms.	This ability maintains continuous operation, especially for mission-critical or always-on services.
NFR.NDT.04	NDT Security	The NDT system should implement encryption, access control, authentication, data masking, and integrity checks; secures data collection, transmission, storage, and model behaviour.	This is necessary to safeguard sensitive data, supports regulatory compliance, and ensures trust in federated and multi-tenant environments.
NFR.NDT.05	NDT modularity	The NDT system should be modular, leveraging the SBA for easy updates and agile maintenance.	This facilitates efficient upgrades and system evolution without downtime.

4.2. NDT building blocks

NDT architectures should be modular or block-based to facilitate rapid development, enhance maintainability, and promote the reusability of components [78], [79]. To fulfil the requirements mentioned in Section 4.1, the NDT instantiation and operation should comprise the following main building blocks. These components enable the NDT to accurately represent and analyse the physical network, predicting future states, which in turn allows the exploration of future scenarios ("what-if" analysis).

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

6GSNS

4.2.1. Modelling and representation layer

This core component comprises the models representing the physical network's objects and behaviours. Its purpose is to create and host high-fidelity digital representations of physical network components for later use. As explored in D2.2 [80] and D2.3 [81], these models can be broadly classified into basic and functional models.

- **Basic Models:** These are NF representations of network elements and topology (e.g., scenario topology, gNBs, UEs, channel models, links, network slices).
- **Functional Models:** These include analytical models, simulators, or ML-based models. They provide network analysis, emulation, diagnosis, prediction, and optimization capabilities.

Although the ITU-T [9][ref: ITU-T Y3090] provides high-level definitions for both basic and functional models, it does not delve into how they should be implemented. Within our project, we interpret these models as structured libraries comprising abstract data structures. These libraries are developed using standardized and harmonized principles to support the creation of a modular, interoperable, and universal NDT.

As shown in Figure 1, this block is primarily represented by the real-world elements in the blue box. However, it's important to distinguish between two stages: the **model design phase**, where a broad range of models is created and stored before the NDT becomes operational, and the **runtime phase**, where an active NDT instance uses only a selected subset of these models. During runtime, a **model manager** is responsible for overseeing the models in use, ensuring they remain valid, that they accurately reflect the data received from the PT, and continue to represent real-world conditions effectively.

4.2.2. Interoperability and integration layer

Given the nature of the NDT, appropriate interfaces must be developed, as they facilitate standardized and seamless integration with other systems and domains. These interfaces are essential for seamless interaction between the twin layer and its underlying physical network, as well as with network applications and other NDTs. Several interfaces are proposed in this sense.

- **Southbound Interfaces**: Facilitate efficient bidirectional information exchange (control and user plane data) between the NDT and the physical network.
- **Northbound Interfaces**: Convey requirements or intents from network applications to the NDT and expose NDT features to third-party applications.
- **Intra/Inter-NDT Interfaces**: Enable communication between different NDT models or instances, supporting modularity and collaboration (e.g., for federated operation).

Advanced ML architectures, particularly LLMs, can enhance these interfaces by translating high-level service intents or policy goals into actionable low-level configurations. For instance, LLMs can assist in selecting and chaining the appropriate basic and functional models within the NDT, mapping them to specific domains such as RAN, core, or edge, and generating corresponding orchestration templates or deployment manifests.

Technically, such interfaces must support context-aware translation, semantic validation, and lifecycle integration across heterogeneous domains. This is especially critical in scenarios where the NDT spans across different vendors and administrative domains, requiring modular, scalable, and real-time interactions between orchestrators, controllers, and the AI model repositories. To ensure seamless interoperability and integration into existing network

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

infrastructures, these interfaces must be standardized and vendor-agnostic, leveraging open specifications such as ETSI ZSM for autonomous service management, 3GPP Network Data Analytics Function (NWDAF) for AI-driven analytics, and O-RAN O1/O2 interfaces for RAN disaggregation and control. They should also support YANG/NETCONF or RESTful APIs for model onboarding, monitoring, and triggering actions based on closed-loop feedback.

4.2.3. Management and orchestration (MANO) layer

The MANO layer is responsible for overseeing the entire lifecycle of NDT instances and their associated resources. This includes the deployment, configuration, operation, optimization, maintenance, and termination of the models that comprise an NDT. Given that NDTs are dynamic, on-demand software entities—instantiated only when triggered by external workflows—the MANO layer plays a crucial supervisory role in ensuring their seamless instantiation and coordination.

To fulfil this role, the MANO layer must orchestrate interactions among all relevant components. For example, it ensures that models are properly initialized with the appropriate data from the model repository and continuously validated against real-time data streams. In the case of AI-driven functional models, it manages the full AI/ML lifecycle, including data preprocessing, training, validation, deployment, and inference.

Additionally, the MANO layer handles simulation and emulation processes, enabling the execution of "what-if" scenarios to support predictive and exploratory analysis. It monitors computational resource usage to guarantee the timely execution of NDT processes, triggering scaling or reallocation as needed to maintain performance.

An additional component in this layer is the security and trust block. This is a core component designed to ensure the secure and trustworthy operation of the NDT across all its components, data flows, and lifecycle stages. This block comprises several key subcomponents, including Identity and Access Management systems to enforce robust authentication and authorization policies, Data Encryption and Privacy Preservation Modules that protect sensitive information both at rest and in transit, and Audit Logs coupled with a Compliance Tracker to maintain accountability and support regulatory adherence. Additionally, the framework ensures transparency and traceability for AI/ML models and DT actions, fostering trust among stakeholders and enabling secure, auditable interactions across federated or multi-tenant NDT environments.

This layer is also key to ensuring the long-term stability and adaptability of the NDT. It activates auxiliary services, such as resource monitoring, performance tracking, and cross-domain coordination mechanisms, which are particularly important in federated NDT setups. Finally, this layer must support automation and zero-touch operation, enabling self-configuration, self-optimization, and self-healing functionalities within the NDT ecosystem.

4.2.4. Monitoring layer

The monitoring layer plays a critical role in ensuring the continuous reliability and effectiveness of the NDT by tracking the health, performance, and accuracy of both the NDT and its physical network counterpart. This layer comprises several subcomponents, including a comprehensive logging and tracing infrastructure to support root cause analysis and event correlation, as well as an intuitive visualization dashboard for presenting operational insights. Together, these elements enable real-time alerting and diagnostics, allowing operators to detect anomalies and performance degradations quickly. Moreover, the system provides insightful visualizations and KPIs that support informed decision-making, facilitate proactive maintenance, and ensure the fidelity and synchronization of the digital representation with the physical network

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

4.3. NDT lifecycle workflows

Building upon the concepts and components described in the previous subsections, this section focuses on the dynamic interactions between the various building blocks and layers of an NDT. These interactions are essential for orchestrating the critical stages of the NDT lifecycle, enabling the system to meet the previously outlined functional and non-functional requirements. To capture these complex interdependencies, we adopt a process-oriented view that describes how the system behaves over time. To that end, unified modelling language sequence diagrams represent these workflows, as they provide a precise and intuitive mechanism for illustrating the chronological order of message exchanges among system entities. While the procedures described herein are intended to remain generic and broadly applicable, it is important to note that customized sequences should be developed for each specific use case to reflect domain-specific interactions and constraints. The following subsections introduce and briefly explain the key sequence diagrams underpinning the LCM of an NDT.

4.3.1. Creation

Figure 8 illustrates the process of creating an NDT, a task typically initiated on demand due to the potential computational intensity associated with running high-fidelity simulations and performing “what-if” analyses. In the 6G-TWIN architecture, this process is orchestrated by the MANO layer, specifically through its dedicated NDT Manager component. Upon receiving a creation request, often triggered by an external system or user, the NDT Manager interprets the operational and analytical requirements and translates them into a formal NDT descriptor [82]. This descriptor encapsulates all essential information needed to instantiate and execute the NDT. The NDT descriptor includes, but is not limited to:

- Initial scenario deployment, including the conditions that should be evaluated, and initialization of parameters, and simulation setup parameters.
- Basic and Functional models to be used by the NDT, indicating whether AI/ML-based models are involved—requiring additional monitoring pipelines, performance bounds, and learning metrics to manage their lifecycle effectively.
- Storage data or live data, and where to retrieve it.
- Desired output format, which may range from user-facing suggestions (in open-loop configurations) to direct network actuation (in closed-loop scenarios) or predictive alerts.
- Software and runtime dependencies, such as required simulators or emulators
- Interconnection of components to form a service, including any constraints on network characteristics like bandwidth and latency.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

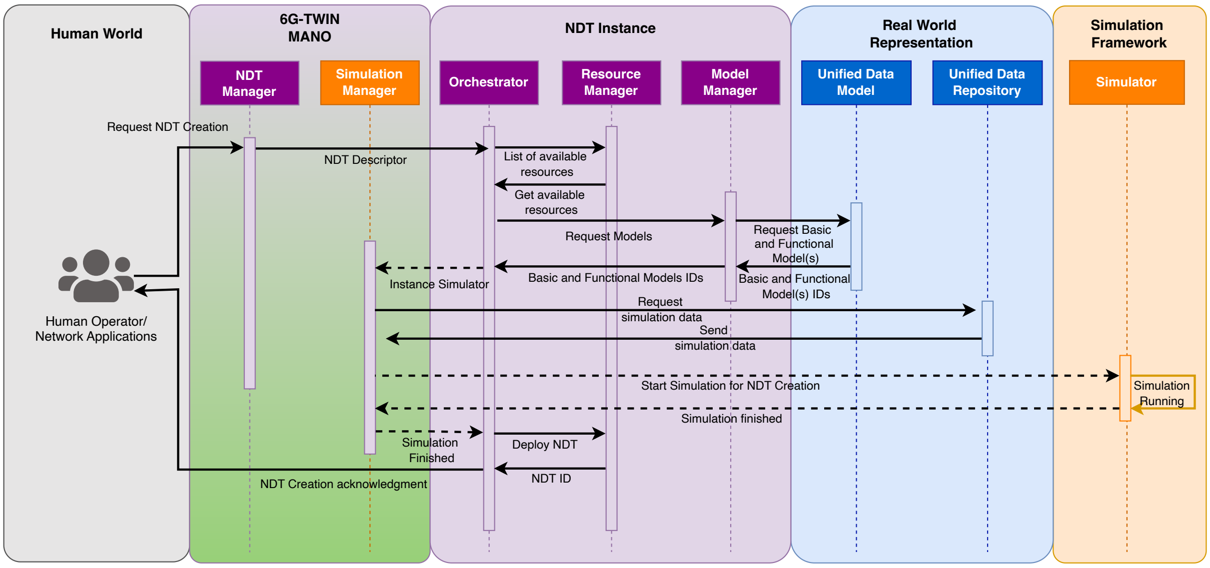

Figure 8. NDT Creation Sequence Diagram.

It is important to highlight that the information and requirements necessary for operating an NDT are typically provided by the external world, often through user-facing interfaces such as a dashboard. For example, a network operator may initiate a request to assess the impact of shutting down a base station under current network conditions. However, such initial requests are often too high-level or vague to be executed directly. As a result, the system must engage in a refinement process to gather additional details. This may involve prompting the user to specify further parameters such as the types of models to be used (e.g., simulation-based or AI-driven), the source and nature of the input data (live or historical), the time span over which predictions or analyses should be performed, and the preferred format for output results. The latter may range from visual insights and performance metrics to actionable suggestions or automated responses. This interaction facilitates the creation of the NDT descriptor and ensures that the NDT is instantiated with a clearly defined and context-aware configuration, thereby increasing its utility and reliability in supporting real-time network operations and decision-making.

Once the request for an NDT is received, the NDT Manager initiates the process by generating a mock-up of a generic NDT instance, in which some foundational blocks are preconfigured, as illustrated in Figure 8. This preliminary structure acts as a template, which the system refines based on the specific requirements outlined in the NDT descriptor. The orchestrator takes over by analysing this descriptor and coordinating with the resource manager to verify whether the underlying infrastructure can support the deployment. The resource manager evaluates the availability of computing resources, such as CPUs or GPUs, necessary to run the NDT components. If sufficient resources are available, the orchestrator consults the model manager to ensure that all required basic and functional models are already created and accessible. These models are expected to be stored in the unified data model repository; if any models are missing, the model manager may initiate auxiliary processes to generate or retrieve them. Additionally, the system may need to pull relevant input data, either historical or real-time, from the Unified Data Repository to feed into the basic models.

At this stage, any simulation tools required for the NDT's operation are instantiated by engaging the simulation manager, which interfaces with the appropriate simulation frameworks. These simulations may also require live or recorded data from the physical network, again accessed through the Unified Data Repository by the simulation manager and sent to the simulation framework when starting a simulation. Once all necessary components—including models, data, and simulators—are properly instantiated and configured, the orchestrator finalizes the deployment of the NDT instance and returns an acknowledgment to the original requester, signalling that the NDT is now operational and ready for use.

4.3.2. Update

The NDT is designed to be intelligent and continuously evolving, constantly updating its models based on new data from the physical network to ensure a high-fidelity representation and co-evolution between physical and NF spaces. As far as the basic models are concerned, they are assumed to be a mere representation of the information coming from the physical network. In that sense, basic models are updated in the same basis as the data collection process.

On the other hand, the functional models can be updated in at least two ways: (i) their performance is continuously monitored and an update is triggered when there is a deviation from the performance bounds in the NDT descriptor; (ii) an automated update regularly programmed by the orchestrator. In the remaining of this section, we assume that the update is reactive rather than scheduled (first case).

Therefore, to update an NDT, several scenarios may occur, according to the type of functional model(s) the NDT is based on. Note also that during this update the NDT service should not be available to be used, so the user should get notified, e.g., through displaying a message in the dashboard, that the service is unavailable or might be producing outdated outputs.

Analytical functional models: Suppose one or more functional models are based on analytical techniques. In that case, the monitoring system in the NDT detects that there is a mismatch between the expected quality metrics and the ones being measured, triggering a quality alert to the orchestrator. The orchestrator then asks the model manager to update the functional model. The model manager communicates with the **UDM repository** to get the latest version of the functional model. A new model must be created if the latest version is the same as the current version. If a new version exists, the orchestrator will be notified and make the respective change. Additionally, it is the responsibility of the orchestrator to update the NDT descriptor to include the new version of the model and to stop the alert triggered by the monitoring system. Figure 9 shows this process. Notice that a request to update an analytical model may happen when a physical principle has changed, e.g., propagation mode is changed or a transmission technology changed.

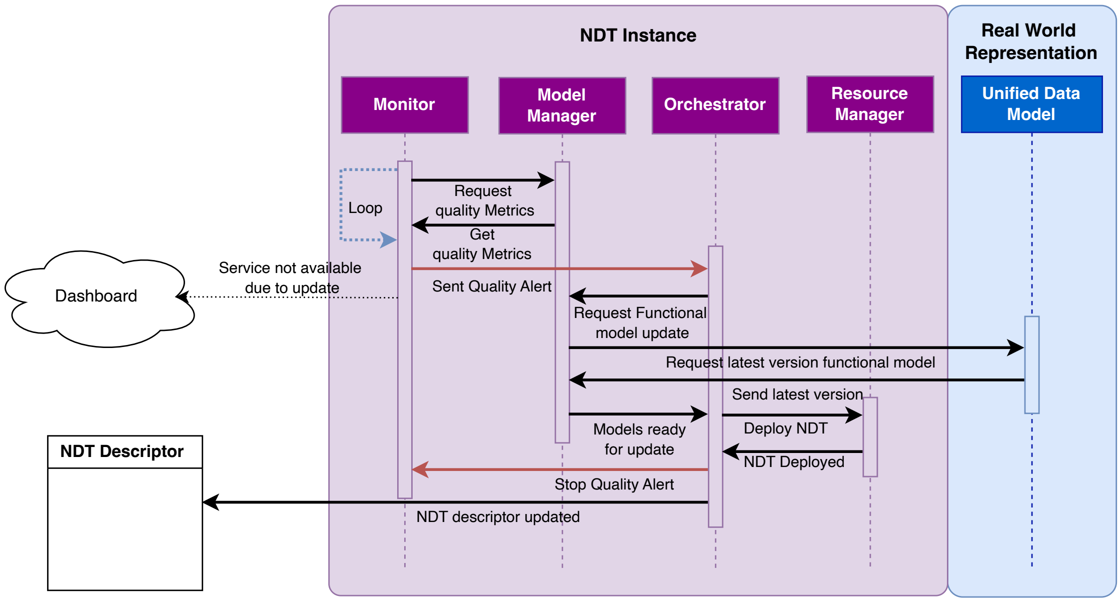

Figure 9. Update analytical functional models.

In the case the functional models are based on AI/ML techniques, a different procedure must be followed. As mentioned in Section 2.5, different pipelines are followed if the functional model is based on supervised DL techniques or on RL techniques. Here, we follow the same approach proposing two different procedures for updating such models.

Functional models based on DL techniques: Figure 10 shows this sequence. Deep learning techniques are based on data. This data can come from different type of sources (directly from the physical network, synthetic data generated by, e.g., a simulator, historical data saved into a database). This data is preprocessed, formatted into the smart data model, and stored in the **UDR** so it can be used for other purposes in the NDT. Similar to the previous case, the monitoring system in the NDT detects that there is a mismatch between the expected quality metrics and the ones being measured, triggering a quality alert to the orchestrator. The orchestrator then asks the model manager to update the functional model. In case a new version of the functional model does not exist, the orchestrator should trigger the creation of one. For that, the NDT manager is contacted, triggering a new training pipeline to the AI workflow block. Since the functional model is based on DL, data, ML libraries, and appropriate infrastructure are needed to train a new model. Once this is done, the AI pipeline notifies the NDT manager that the new functional model is created, then the new version is pushed to the **UDM repository**, and the orchestrator is notified that a new version of the functional model is created. The orchestrator makes the model change, updates the NDT descriptor, and stops the alert triggered by the monitoring system.

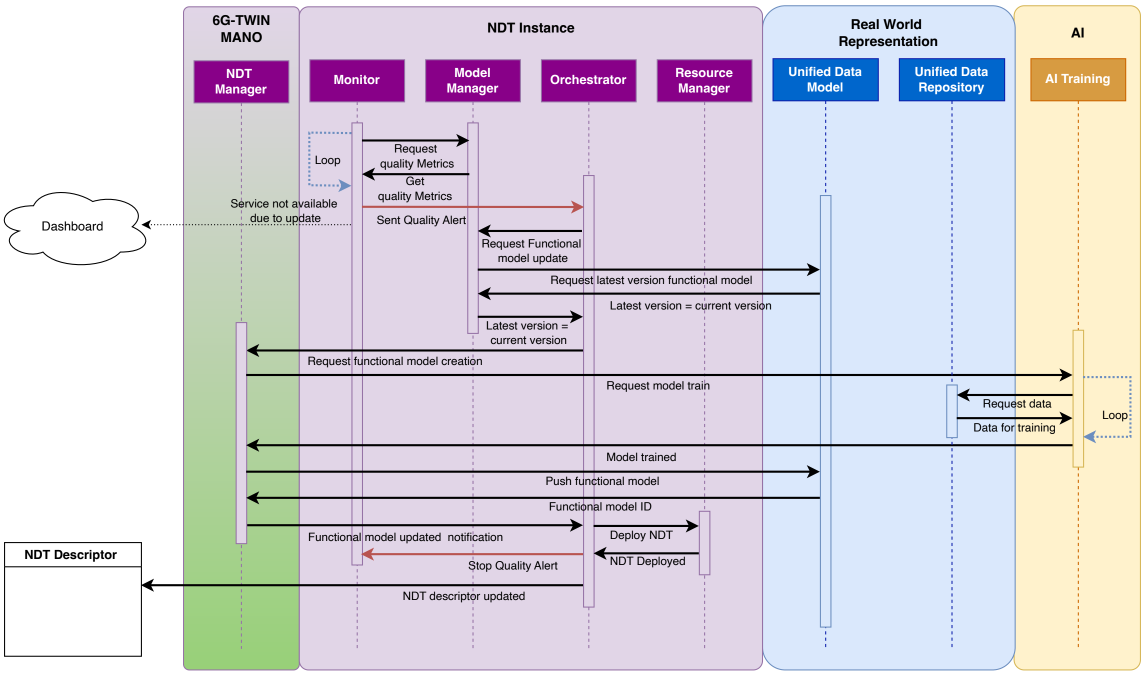

Figure 10. Update DL based functional models.

Functional models based on RL: Figure 11 shows this sequence. RL techniques require interaction with a given environment to learn. In this case, a simulator can be used as an environment. The process follows the same initial steps as for updating a functional model based on DL. The main difference resides in the model creation. After creating an RL-based functional model, the NDT manager interacts with both the AI training and the simulation framework: In each iteration of the AI training, a simulation is conducted, and the simulation results are transferred from the simulation framework to the AI training via the simulation manager. The simulation results are then used in the AI pipeline to train the model. Once the model is trained, the new model version is saved in the **UDM repository**. The NDT manager then notifies the Orchestrator that a new functional model is ready to be deployed in the NDT;

the orchestrator makes the model change, updates the NDT descriptor, and stops the alert triggered by the monitoring system.

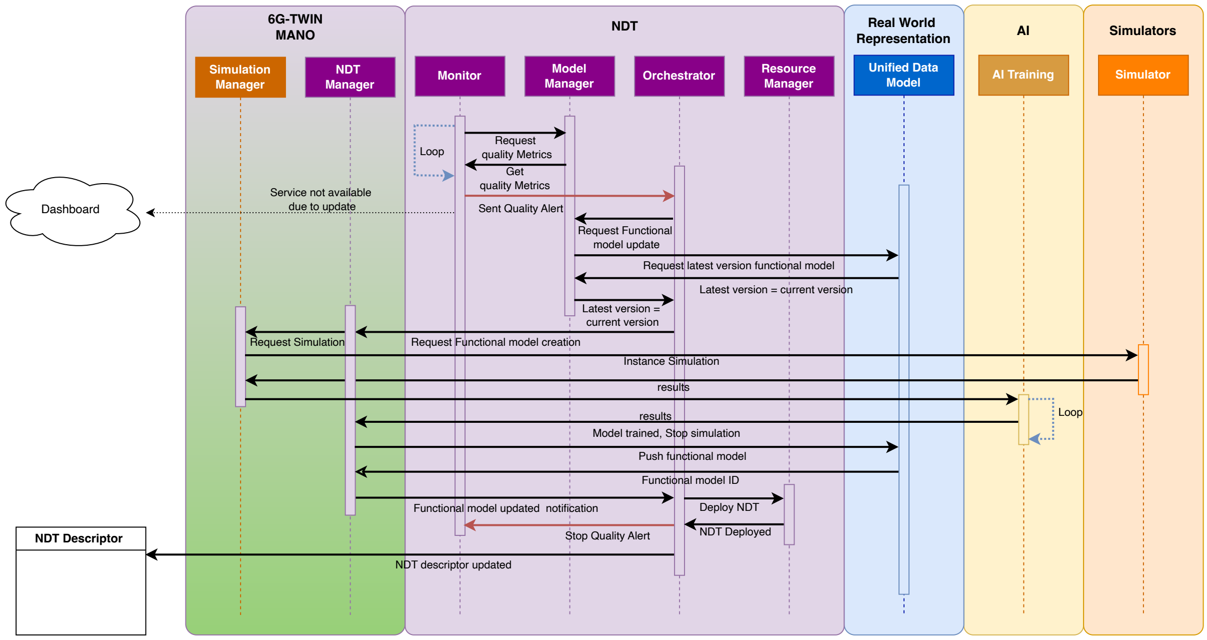

Figure 11. Update RL based functional models.

4.3.3. Deletion

For deleting an NDT, the process is relatively simple, as shown in Figure 12. First, the request for NDT deletion may come from an external source such as the application that initially requested or a human operator in the loop. Notice that the NDT might be also deleted because it exceeds the timespan as indicated in the NDT descriptor. The NDT manager processes the NDT deletion request (if it is coming from an external source) and generates the NDT delete instruction for the orchestrator.

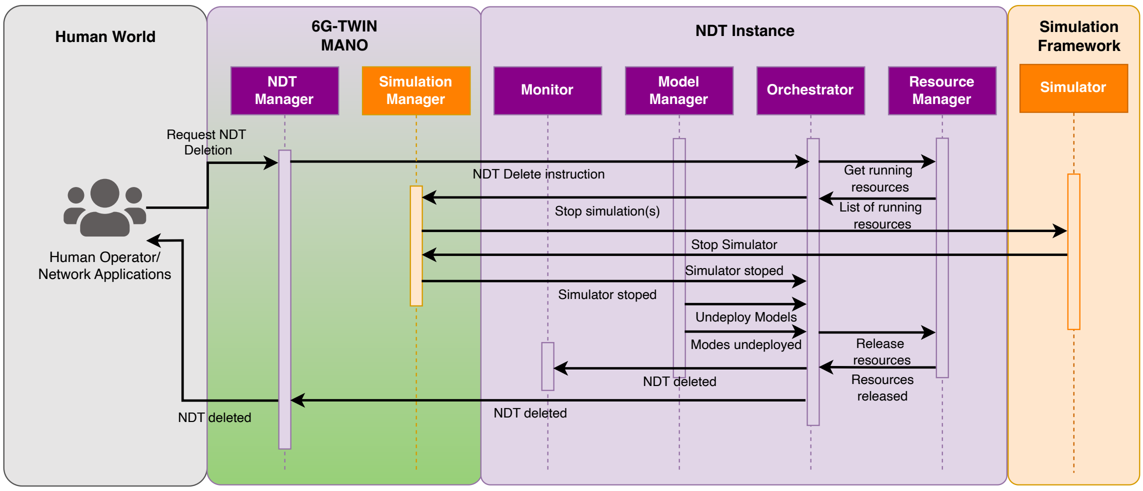

Figure 12. NDT deletion.

The orchestrator then requests the list of occupied resources from the resource manager and checks the status of each one. Additionally, it undeploys any functional or basic model. After every component is stopped, the orchestrator then releases all the occupied resources and notifies the external process that the NDT is deleted.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

5. NDT workflows' early implementation and examples

Building on the theoretical foundations outlined in earlier sections, where the architectural components enabling NDTs as a service and creating them were introduced, this section shifts focus to the practical realization of these concepts. It presents a series of early implementations and examples that demonstrate the feasibility and effectiveness of the proposed NDT framework within the context of the 6G-TWIN project, to be further extended in WP4 deliverables.

Section 5.1 showcases a live demonstration of the 6G-TWIN data collection and telemetry integration framework, as detailed in D1.2 [2]. This setup highlights how real-time telemetry and AI-native modules can support concrete NDT-driven use cases, including UE localization and energy-efficient network management. Section 5.2 explores the creation of a complete NDT instance, combining basic models of network state with functional models for radio coverage prediction, an essential capability for advanced scenarios such as teleoperated driving and dynamic RAN resource optimization. This implementation leverages GreyCat, a graph-based data engine, to manage telemetry and measurement data and integrates DL techniques for model-based inference. Finally, Section 5.3 details an automated pipeline for maintaining the reliability and accuracy of functional models within an NDT instance, demonstrating a closed-loop update mechanism driven by real-time performance monitoring.

5.1. Data ingestion for basic model creation

The 6G-TWIN project is investigating novel frameworks for integrating NDTs with AI-native architectures to enable real-time, intelligent RAN control. As an initial step towards this goal, we created a demo highlighting how advanced telemetry and data modelling can support practical use cases such as energy savings and UE localization.

Revising the 6G-TWIN Architecture, Figure 3 presents a modular and scalable architecture for the data collection framework based on a layered stack:

- The Telemetry Data Layer ingests, processes, and outputs raw data from RAN, core, and transport nodes.
- The Harmonization Layer transforms raw telemetry into Smart Data Models, aligned with 3GPP/O-RAN standards.
- The Digital Twin Core manages basic models (real-time state) and functional models (predictive simulation, optimization).
- A Simulation Framework supports training, scenario testing, and synthetic data generation.
- The MANO Layer provides closed-loop control, coordination, and automation across the stack.

This demo was showcased at EuCNC 2025, validating the 6G-TWIN architecture through a real-time UE localization scenario. The setup simulated two days of symmetrical uplink/downlink traffic, compressed into a 5-minute test cycle. Key Performance Metrics (KPMs) were collected via the Telemetry Gateway (TGW), which handled data ingestion and harmonization. As shown in Figure 13, the TGW collects information from different sources such as NATS-protobuf for the Control Unit (CU), UDP for AI-RSG and Distributed Unit (DU), and third party via TCP. Then, such data is transformed into 3GPP data structures and converters for data harmonization. Finally, the TGW posts this data into influx and Kafka buses, for KPI and performance metrics usage.

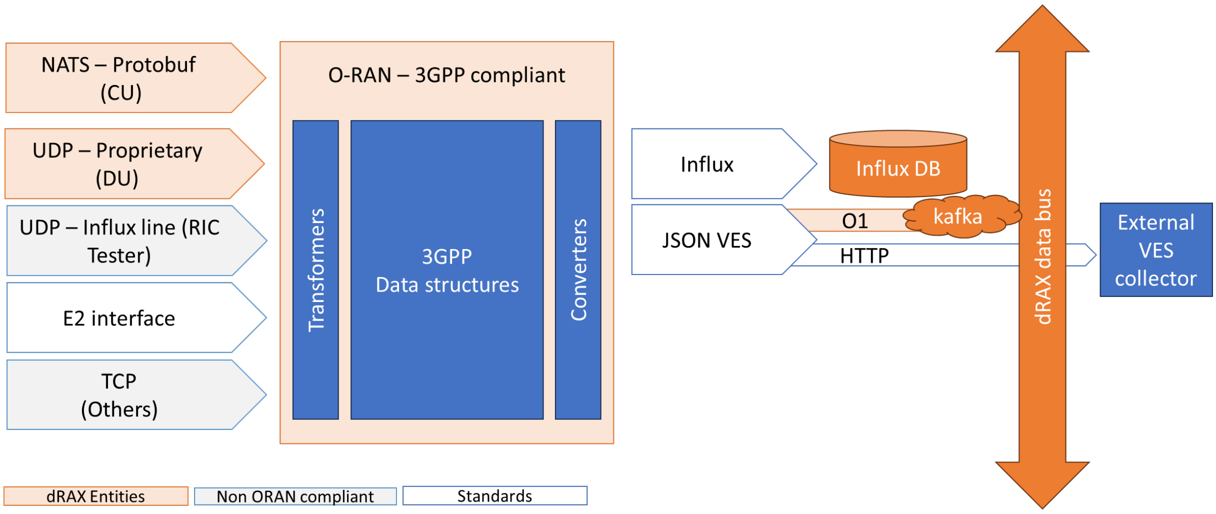

Figure 13. dRAX Telemetry Gateway architecture.

The setup of this demo consists of 3 main components:

- **Emulated RAN:** The emulated RAN was implemented via the VIAVI AI-RSG, which provides a realistic 6-cell and 50 UEs scenario with a traffic profile from the Spanish operator (see Figure 14) in collaboration with the BeGREEN SNS project [83], [84]. The traffic profile shows a 2-day (x-axis) scenario of the traffic in Mbps (y-axis) of the Spanish operator projected on six cells with three different frequency bands. Every 15 minutes, a change in the average traffic is mapped into the system, hence it is a step-like lookalike. Two low traffic periods can be seen, describing early hours in the morning.
- **Telemetry collection:** The TGW is fed with data from the emulated RAN via E2 interface and published in the dRAX databus as described previously.
- **Scenario modelling:** The UE location modelling is provided via an rApp using a Weighted Least Squares algorithm. This algorithm dynamically estimates UE positions based on Reference Signal Received Power (RSRP) values. This can be seen as an initial NDT implementation that only uses a functional model to estimate UE location. More complex functional models, such as beam forming and UE/beam association, will be developed as part of the energy-saving use case.

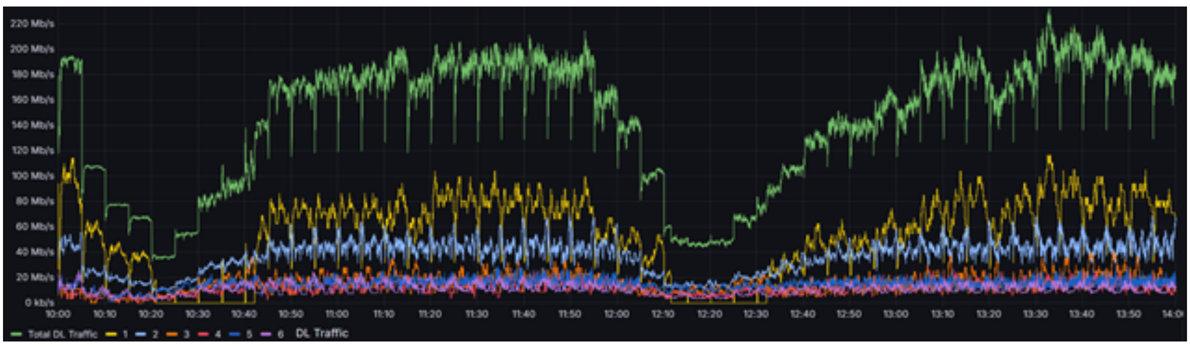

Figure 14. Realistic traffic used in the demo.

In this way, the architecture of a simple NDT can be shown, where data from the RAN can be used to model an NDT working with basic models. This demonstrates how xApps/rApps can consume harmonized data to provide adaptive, location-aware NFs as presented in Figure 15.

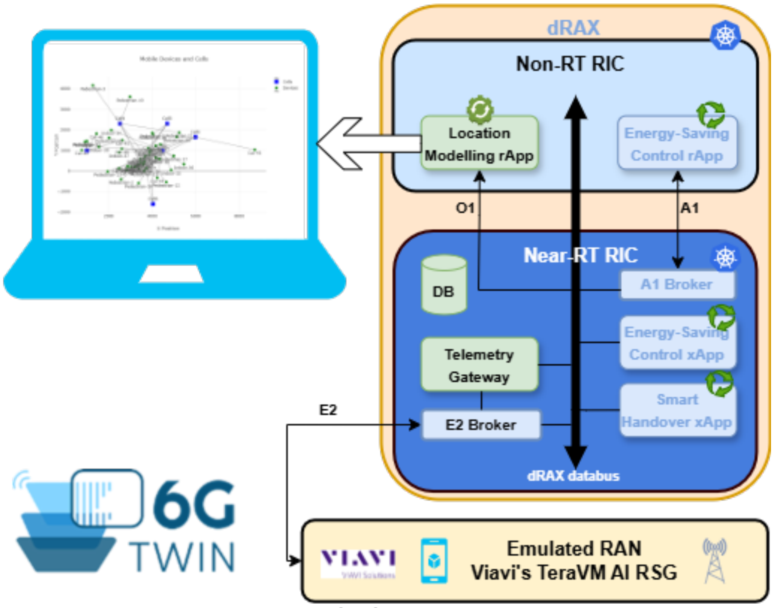

Figure 15. EuCNC 2025 demo set up.

To show the implementation, a dashboard was created to present the UE location results and the network raw data to oversee the network status (see Figure 16). On the left side, the estimated location map is presented with the UE to gNB links. In the upper right, two plots show the energy efficiency and the power consumption of each gNB. At the bottom, the status of the network presents the power of each cell, its consumption, the number of UEs attached, and the traffic of each cell. Initial results show that the location estimation achieved an accuracy of approximately 50 meters, both indoors and outdoors, with data harmonized to sub-second granularity. The rApp extracted real-time metrics such as the number of cells, connected UEs, and over-the-air radio indicators. Future extensions will integrate additional data sources to enhance spatial resolution, such as GIS maps [x], and support more complex control strategies.

Figure 16. Demo dashboard.

Finally, the 6G-TWIN demonstration at EuCNC 2025 exemplifies the power of AI-native architectures when supported by real-time, standards-aligned telemetry. By bridging physical

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

and digital domains, this architecture provides a real implementation for the proposed NDT architecture towards energy-efficient 6G networks.

5.2. NDT creation: predicting radio coverage

Enabling advanced applications such as teleoperated driving requires the integration of several functional models across various domains. This includes predicting radio coverage, efficient resource allocation in the RAN, and optimized slicing at the CN level. As a primary step towards these NDT-based use cases, this work focuses on radio coverage prediction as a representative functional model.

Instantiating an NDT entails populating basic model templates with real-world data and selecting appropriate functional models based on the specific application or use case. This process is overseen by the NDT MANO component. The NDT MANO is tasked with the creation, configuration, and LCM of NDT instances. While the NDT MANO can be fully automated to align with the ZSM paradigm, current approaches rely on human expert knowledge for building these instants.

During the instantiation process, the NDT MANO selects functional models that correspond to the intended application. At the same time, it populates the basic models with data retrieved from the **UDR**, ensuring alignment between structural and functional layers. The functional models are then calibrated or trained using the data-augmented basic models, resulting in parameterized variants tailored to the application context. These trained models are saved in a dedicated repository, managed by the MANO, where they are indexed for traceability and evaluation. This repository also enables comparative analysis of different algorithms or configurations and supports offline validation via simulation, particularly useful for use cases involving network control, facilitated by the interaction of the NDT MANO with the network MANO in the physical network.

Figure 17 illustrates an example of the NDT instantiation process, from the viewpoint of model management only, followed by the implementation details in the following subsections.

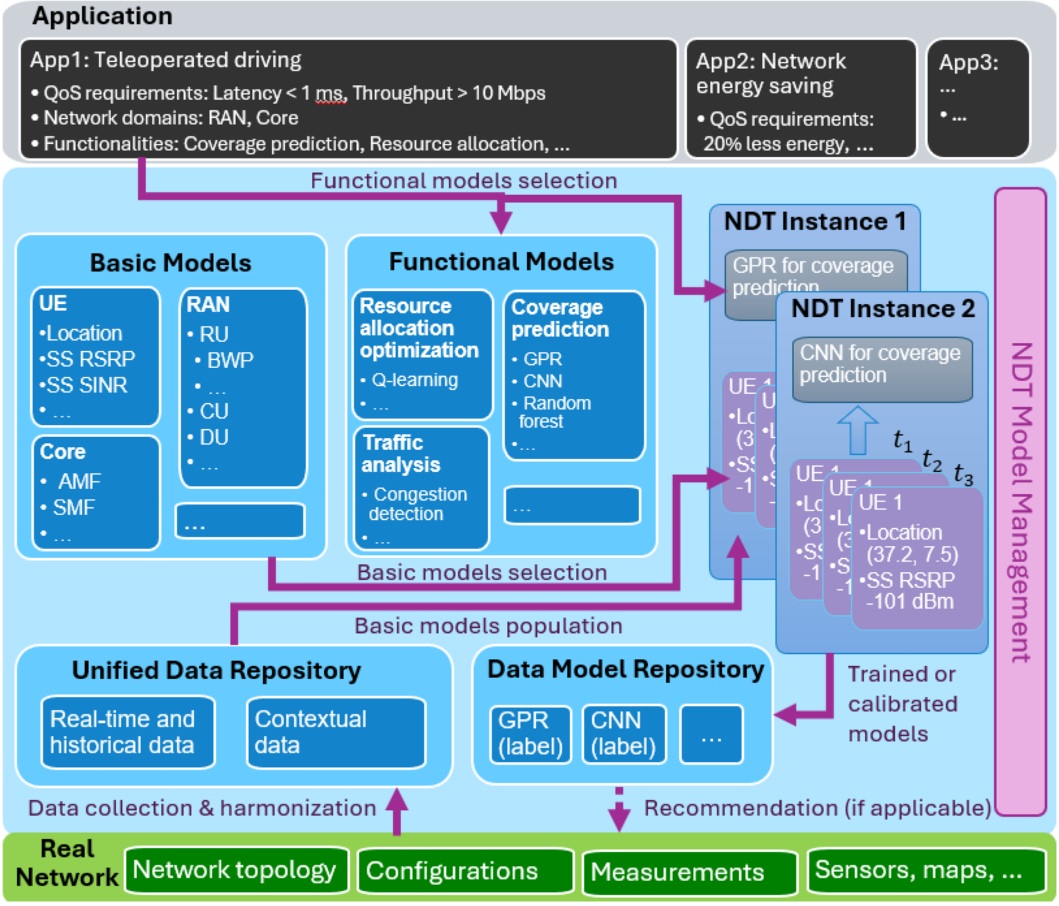

Figure 17. An example of NDT instantiation for radio coverage prediction.

5.2.1. Functional models and radio coverage estimation

Due to the limitations of exhaustive field measurements, contemporary approaches employ model-based, data-driven, or hybrid techniques to reconstruct complete coverage maps from limited data points. Traditionally, these predictions relied on physics-based methods like ray tracing and ray launching, which deliver high accuracy but demand significant computational resources and detailed environmental knowledge. Alternatively, empirical models (e.g., 3GPP and COST) are easier to implement and broadly applicable but often underperform in diverse environments that stray from ideal conditions [85].

Recent advancements have embraced data-driven strategies, learning signal propagation directly from collected data [86]. These techniques utilize statistical inference or ML to predict coverage. Popular methods include Gaussian Process Regression (GPR), which leverages spatial correlation through kernel-based interpolation, and Convolutional Neural Networks (CNNs), which approach the task as an image-to-image transformation. Such methods have shown accuracy comparable to ray tracing while reducing computational demands [87]. In our implementation, we employ GPR and CNN models for their contrasting levels of complexity and responsiveness to data variability.

5.2.2. Implementation using greycat

GreyCat provides the framework for implementing and managing the NDT basic models, maintaining a graph-based structure to store both network knowledge and data. It incorporates the data model depicted in the network schema, which includes attributes and measurements

Co-funded by the European Union
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

collected from network components, especially NRCellCU (the CU object associated with a gNB) and associated NRCellDUs (the DUs objects). These are connected to elements like bandwidth part and duplexing mode, either in time (TDD) or frequency (FDD).

The platform also records UE locations and synchronization signal values, focusing particularly on indicators such as RSRP, Reference Signal Received Quality (RSRQ), and Signal-to-Interference-plus-Noise Ratio (SINR). GreyCat integrates with external data sources using connectors (e.g., MQTT) and supports importing log files. A standout feature is its use of data sharding, which facilitates efficient time- and space-specific data retrieval using low-memory node pointers.

GreyCat also allows coverage extrapolation to unmeasured areas using predictive algorithms and visualizes the results. Although GreyCat supports native model execution, GPR and CNN models were externally developed in Python and interfaced with GreyCat data

5.2.3. Evaluation approach

To evaluate the accuracy of the DT, we tested the predictive performance of the functional models while maintaining a uniform dataset. This ensured the observed differences in results were solely due to model differences. The scenario involved collecting RSRP measurements from known coordinates within a controlled indoor setting. These values were used for both training and validating the coverage prediction models.
The experimental data was gathered on a laboratory floor spanning 46m by 12m, shown in Figure 18. Coordinates of each measurement point were manually annotated on a floor plan due to the absence of native location data in the network outputs.

A software-defined radio-based 5G gNB was deployed using the OAIBOX (built on OpenAirInterface), enabling full stack control. This platform was selected for its flexibility in logging and real-time configuration.

Configuration Summary:

- Duplexing: TDD
- Frame Format: DDDDDFUUUU
- Frequency: 3809.28 MHz
- Bandwidth: 40 MHz
- Antennas: 3 dB gain dipoles
- UE: iPhone-14 Pro

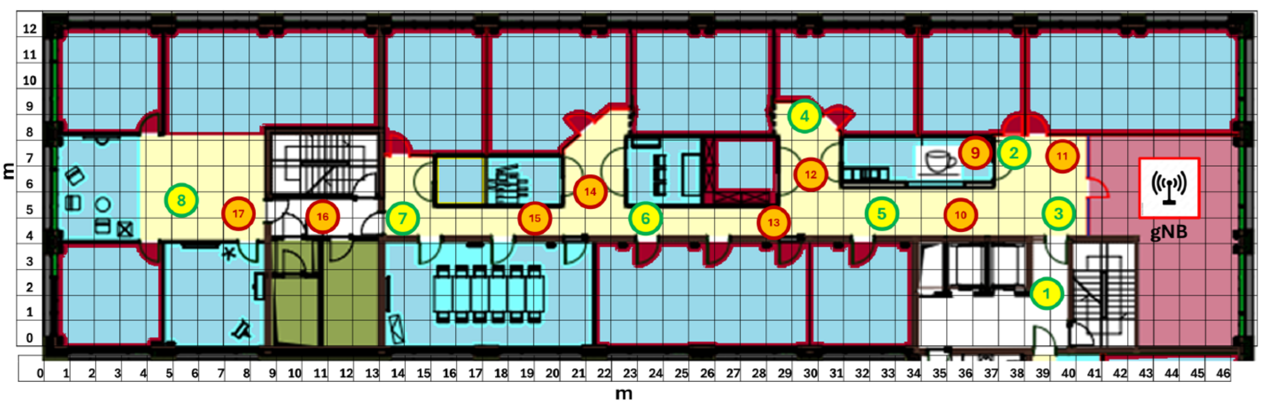

Figure 18. Floor plan of the indoor measurement setup at the LIST premises, indicating the location of the gNB, the coverage area, and the georeferenced measurement points used for both training and evaluating the model. The axes are labelled in meters (m).

5.2.4. Data collection methodology

After connecting the UE to the gNB, logs capturing RSRP, RSRQ, and SINR were continuously gathered while the UE remained stationary for about two minutes per location. The focus was placed primarily on RSRP due to minimal interference in the licensed band.
Data logs were then ingested into GreyCat and structured according to the internal schema. Measurements were timestamped and spatially linked, allowing for efficient manipulation and predictions. An example of the GreyCat internal view for a UE at one location is illustrated in Figure 19.

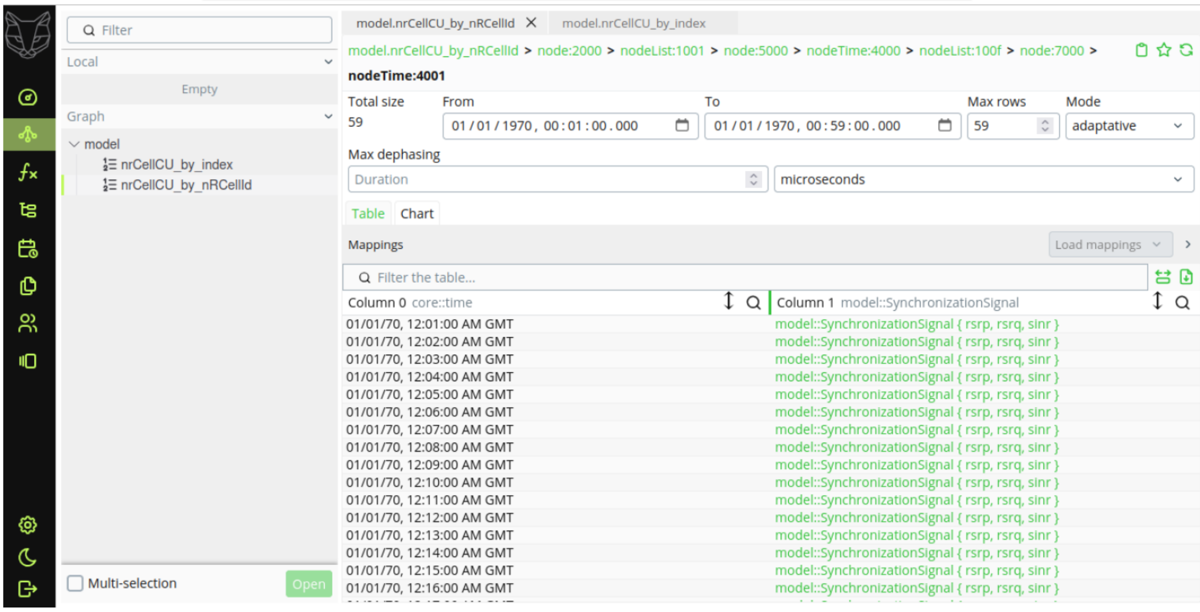

Figure 19. Internal view in GreyCat showing the collection of synchronization signals for a UE at a specific location.

5.2.5. Algorithmic implementation

To demonstrate the application of functional models within the NDT framework, we implemented two approaches for radio coverage estimation: a probabilistic, kernel-based method (GPR) and a DL approach (CNN). These models serve as functional abstractions that learn the relationship between spatial coordinates and signal strength (RSRP) from partial measurements. Their design reflects different trade-offs between complexity, generalization capacity, and responsiveness to data variability. The following describes the input configuration, training procedure, and inference process for each model.

GPR Implementation:

- Input: (x, y) location and average RSRP.
- Kernel: Radial Basis Function.
- Training: Hyperparameter tuning via likelihood estimation.
- Prediction: Grid-based coverage interpolation.

CNN Implementation:

- Input: 2D RSRP grid with missing data masked.
- Architecture: Two convolutional layers.
- Training: Directly on full data, using MSE on known values.

- Prediction: Entire map inferred in one pass.

5.2.6. Quantitative evaluation

Figure 20 visualizing GPR and CNN coverage maps highlights the models' ability to reconstruct signal strength, with training and testing datasets separated. Prediction errors were evaluated for each point.

CNN showed greater robustness in sparse-data conditions, while GPR performed well in line-of-sight areas but degraded in near-line-of-sight locations. These findings emphasize the NDT's potential in supporting comparative evaluation and selection of suitable models based on operational scenarios.

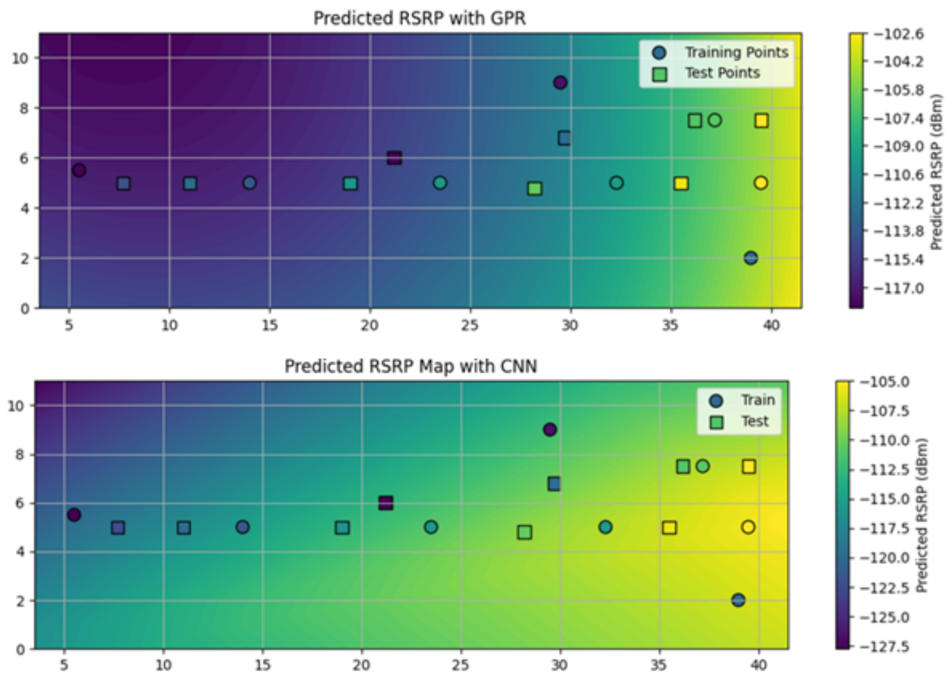

Figure 20. Predicted RSRP map with GPR and CNN.

5.3. NDT update: functional model based on deep learning

Once the NDT is instantiated, maintaining its fidelity and alignment with the physical network becomes a critical task. This responsibility is jointly managed by the NDT MANO layer and the network monitoring system. To illustrate this, consider a scenario where a functional model within the NDT begins to deviate from its expected performance bounds. In such cases, corrective action must be taken to preserve the integrity and utility of the NDT.

As shown in Figure 21, the system follows a predefined workflow to automatically update or replace the underperforming model. This process is triggered when the model no longer satisfies the QoS constraints or performance thresholds set for the NDT instance. This update is orchestrated by retrieving new models or retraining existing ones, ensuring that the NDT continues to deliver accurate, real-time insights and decisions aligned with network conditions.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

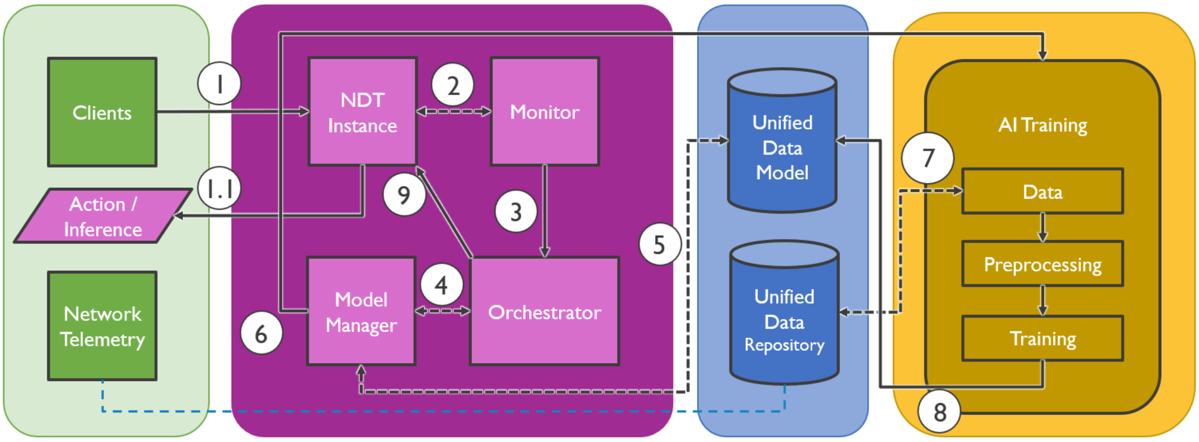

Figure 21. NDT Update general workflow.

The NDT instance receives packets from the UE (1) and performs inference by assigning a label to each packet to be classified within the network (1.1). Meanwhile, the monitor checks response times, system load, and algorithm accuracy (2). When these parameters fall outside the system's required thresholds, the monitor alerts the Orchestrator, indicating that the current model cannot meet the network's requirements (3). Next, the Orchestrator requests the Model Manager to update to the latest version of the implemented model (5). The Unified Data Model reviews the last available update (5). If the newest version in the Unified Data Model is the same as the one already implemented in the NDT instance, the Model Manager requests training in a new version (6). Then, the AI Training module uses the normalized data storage on the Unified Data Repository to train a new model; this data is collected from network telemetry (7). Once enough data is collected, a new model will be built through the training phases. When the new model is ready, it is stored in the Unified Data Model (8), which sends a message to the Model Manager (5) to notify that the new version is available. The Model Manager notifies the Orchestrator about the new model's version (4), then the Orchestrator deploys the new model into the NDT instance (9). From that point on, the NDT continues making decisions based on the new model version.

5.3.1. Implementation

To illustrate this case, an NDT instance was created using DynamicSim [88], [89], [90]. Python, Prefect, Kubernetes and a dataset from [91], provided in pcap format and containing 1,803,059 packet samples across three traffic classes: TCP (999765 samples), UDP (803087 samples), and Multipath TCP (MPTCP) (207). The validation setup involved a desktop machine with an AMD Ryzen 9 5900 12-core processor, an Nvidia RTX 3080 graphics card, 64 GB of RAM, and Windows 11, while Docker Desktop with Kubernetes v1.29.1 was used for container management.

Figure 22 shows our Grafana monitor. In it, you can see the client is a container that sends the packet stream to the NDT, which will later be classified by a functional model based on DL. When the NDT monitoring system detects that accuracy and reliability parameters are not being met, it notifies the NDT orchestrator to trigger the pipeline that trains and deploys a new model to replace the current one. The image shows the time required for preprocessing and training, based on the amount of data collected up to that point. Prefect manages the pipeline, an open-source orchestration engine that turns Python functions into production-grade data workflows. Figure 23 shows the model training process, the Prefect tool handling the training workflow, the name "blazing-wildcat" is a random token created by Prefect to identify this pipeline, and the timeline shows when the pod to perform data preprocessing is started. In the Kubernetes jargon, a pod is the smallest deployable computing unit, similar to a container. When these pods finish their work, they are undeployed.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

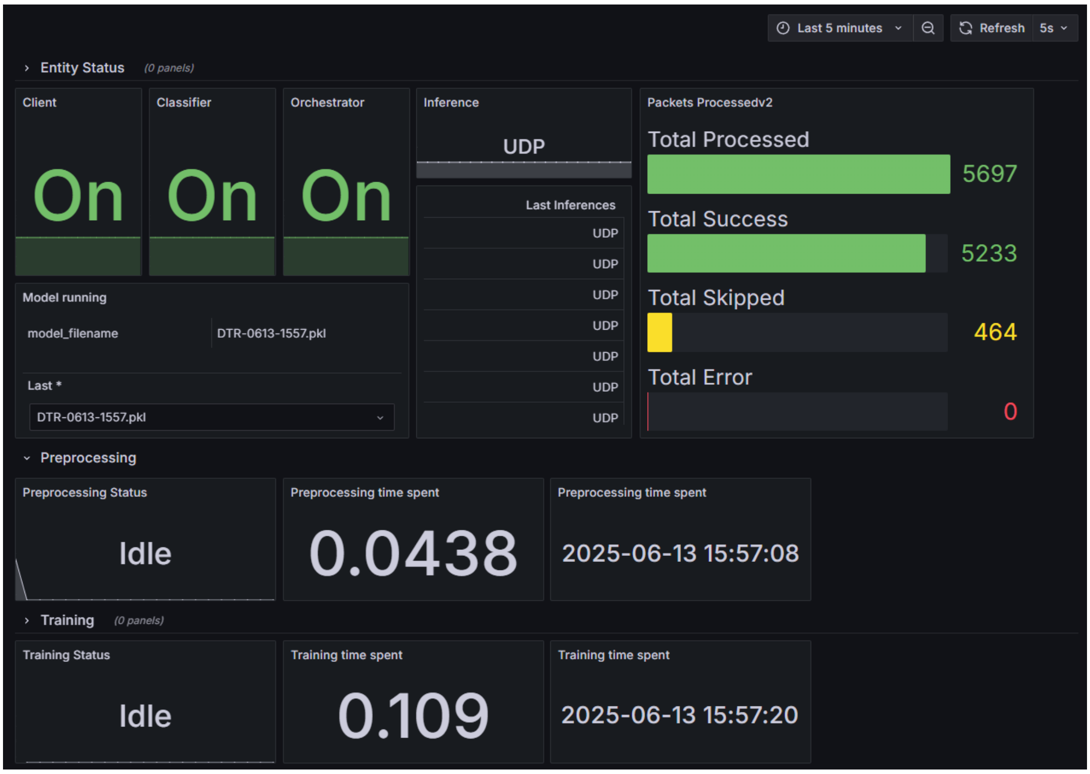

Figure 22. Grafana Monitor acting as dashboard.

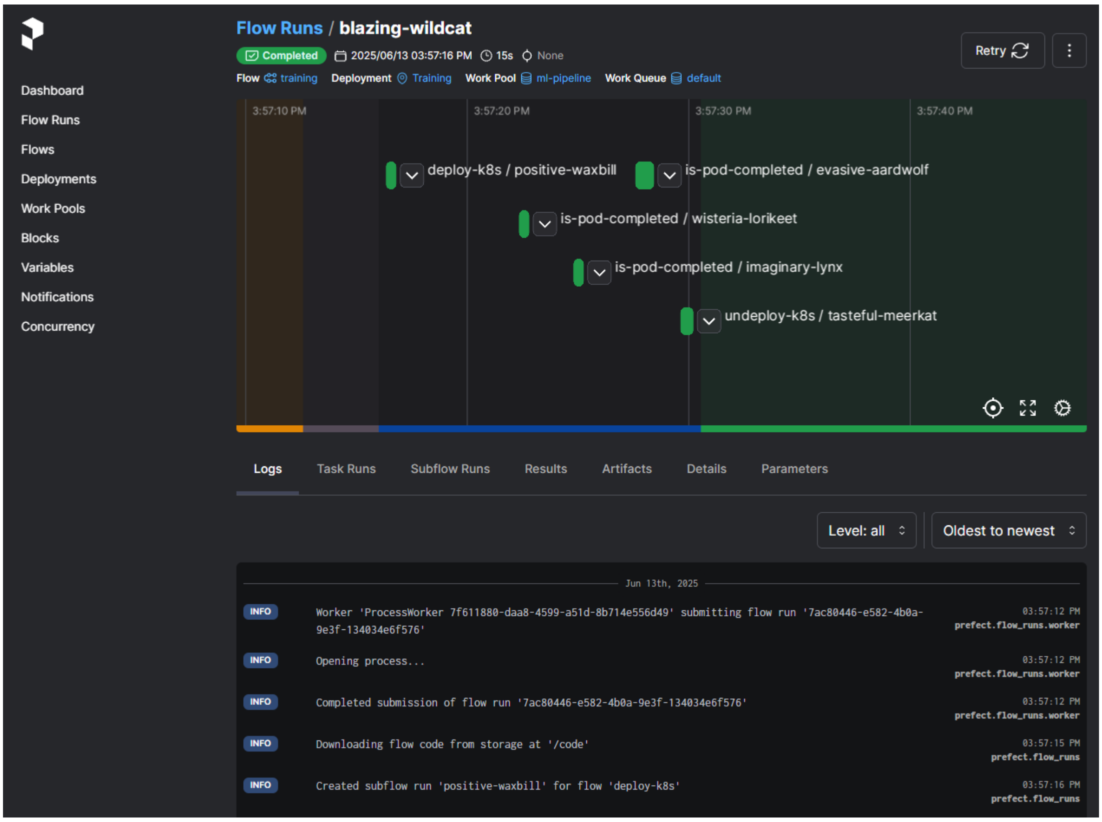

Figure 23. Prefect AI training pipeline.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

The functional models used are a Lineal Regressor (LR), a Decision Tree Regressor (DTR), Deep Neural Network (DNN), These models were selected because they offer a balance between interpretability, performance, and scalability for different types of traffic classification scenarios.

Figure 24 presents the confusion matrices for the three classification models, highlighting their accuracy. Among them, the DNN model delivers the best performance, even when dealing with highly imbalanced class distributions. It is closely followed by the DTR model. While both models perform comparably for the dominant classes, the DNN surpasses the DTR by 4% in correctly identifying the MPTCP class. In contrast, the LR model struggles to classify MPTCP accurately but demonstrates solid performance for the two most common classes, TCP and UDP.

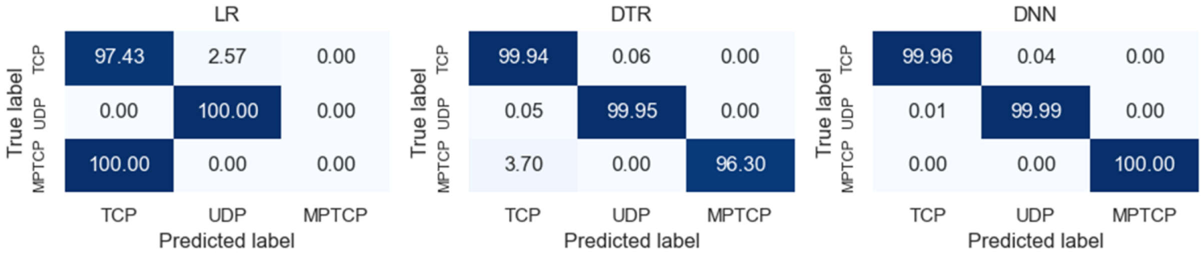

Figure 24. Confusion matrices of the different functional models.

An essential consideration is that model selection greatly influences processing time, as shown in Table 4. The DTR model offers the most favourable trade-off between accuracy and efficiency. It benefits from the unencrypted nature of the input data and effectively utilizes multicore processing capabilities for faster training and inference. Conversely, the DNN model, due to its higher complexity, is significantly slower up to 19 times compared to LR and DTR, despite providing only modest gains in accuracy. This observation is consistent with prior studies on ML applications in traffic classification [92].

Table 4. Total deployment time of different models.

Model	Data preparation	Development and Training	Integration	Deployment	Predictions Packets/Second
LR	5	4	112	5	264027.6
DTR	5	6	108.5	5	264217.8
DNN	5	1033.5	112.5	5	13829.34

However, in scenarios involving encrypted traffic, simpler models like DTR may be inadequate. More sophisticated models such as DNN might be required, even with their higher computational cost [93], [94]. In such cases, implementing hardware acceleration could help mitigate latency, enabling a more practical compromise between precision and inference speed.

6. Conclusions

This deliverable marks a key milestone in the 6G-TWIN project by finalizing the functional architecture of an AI-native, NDT-enabled framework for 6G networks. Building on the foundations laid in Deliverables D1.1 and D1.2, D1.3 introduces the 6G-TWIN MANO layer, a critical vertical layer responsible for coordinating the managers and orchestrators of the subcomponents of this architecture. This layer enables autonomous, closed-loop control in highly dynamic, distributed network environments. D1.3 will be further updated in D1.6.

In addition, this document, together with the paper presented in EuCNC 2025 [3], updates the 6G-TWIN architecture presented in D1.1 [1], integrating the following essential new components:

- An AI training module designed to develop functional models using ML techniques.
- This deliverable focuses on a 6G-TWIN MANO layer. This layer governs the integration and operation of all components within the AI-Native, NDT-enabled architecture, ensuring synchronized interactions between the physical network and the NDT elements. It is defined as a vertical layer encompassing the managers and orchestrators of various NDT components.
- A unified dashboard, serving as an interactive interface for stakeholders to configure models, access functionalities, and monitor NDT operations in real-time.
- Fulfilling the objectives of Task 2.4, this document details the decomposition of the NDT into basic and functional models, describes their respective roles, and outlines key workflows for data ingestion, simulation, AI training, and real-time model updates. It also elaborates on the architectural requirements and technical enablers for federated orchestration across multiple domains, leveraging AI for scalable and secure network service management.

Furthermore, this deliverable emphasizes the importance of aligning NDT functionality with ZSM principles, highlighting the integration of intent-based control and AI/ML mechanisms.

In its last section, this document showcases early implementations that demonstrate the practical realization and effectiveness of the proposed NDT framework. These proof-of-concept implementations validate the architecture's viability in real-world use cases like energy-efficient RAN operation and coverage prediction.

Following the architectural consolidation presented in this deliverable, the 6G-TWIN project will now shift its focus toward the practical implementation, integration, and validation of the proposed components. In particular, the next phases will operationalize the 6G-TWIN MANO layer, enabling dynamic instantiation, control, and LCM of NDTs in real network environments.

Future work will involve deploying the architecture into federated testbeds to validate its support for key use cases, such as teleoperated driving and energy-aware RAN optimization. These deployments will serve as a platform to test the viability of the proposed closed-loop intelligence, including the performance of AI-based functional models and the responsiveness of model update workflows.

Additionally, ZSM principles will be further explored and integrated through the Network MANO/ZSM layer, particularly using intent-based interfaces, AI-native automation, and FL strategies. Open challenges such as privacy-preserving model sharing, multi-tenant orchestration, and real-time performance at scale will guide future research directions.

Finally, insights gathered from implementation will inform refinements of the 6G-TWIN architecture and feed into standardization efforts, ensuring that the developed framework aligns with emerging 6G requirements and industry best practices.

References

[1] 6G-TWIN Project, “Deliverable D1.1: 6G-TWIN architecture and technical foundations(initial).” Accessed: Jun. 15, 2025. [Online]. Available: https://6g-twin.eu/resources/#deliverables

[2] 6G-TWIN Project, “Deliverable D1.2: Secured, scalable, and distributed data exposure and collection framework (initial).” Accessed: Jun. 15, 2025. [Online]. Available: https://6g-twin.eu/resources/#deliverables

[3] S. Faye *et al.*, “A Functional Framework for Network Digital Twins.” Zenodo, Apr. 2025. doi: 10.5281/zenodo.15304760.

[4] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks,” *IEEE Commun. Surv. Tutor.*, vol. 16, no. 3, pp. 1617–1634, 2014, doi: 10.1109/SURV.2014.012214.00180.

[5] M. Liyanage *et al.*, “A Survey on Zero Touch Network and Service Management (ZSM) for 5G and Beyond Networks,” *J. Netw. Comput. Appl.*, vol. 203, p. 103362, 2022, doi: https://doi.org/10.1016/j.jnca.2022.103362.

[6] NASA, “Digital Twins and Living Models at NASA.” Accessed: Jun. 15, 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20210023699

[7] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital Twin: Enabling Technologies, Challenges and Open Research,” *IEEE Access*, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[8] Z. Wang *et al.*, “Mobility Digital Twin: Concept, Architecture, Case Study, and Future Challenges,” *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17452–17467, 2022, doi: 10.1109/JIOT.2022.3156028.

[9] ITU-T, “Digital twin network – Requirements and architecture,” no. ITU-T Y.3090. 2022. [Online]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.3090-202202-I!!PDF-E&type=items

[10] ITU-T, “Digital twin for management and orchestration in IMT-2020 networks and beyond,” no. ITU-T Y.3092. 2024. [Online]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.3092-202409-I!!PDF-E&type=items

[11] ETSI, “Zero touch network & Service Management (ZSM).” Accessed: Jun. 15, 2025. [Online]. Available: https://www.etsi.org/technologies/zero-touch-network-service-management

[12] T. O. Olwal, K. Djouani, and A. M. Kurien, “A survey of resource management toward 5G radio access networks,” *IEEE Commun. Surv. Tutor.*, vol. 18, no. 3, pp. 1656–1686, 2016.

[13] 3GPP, “5G;NG-RAN; Architecture description,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.401, Jul. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138400_138499/138401/16.02.00_60/ts_138401v160200p.pdf

[14] ITU-T, “Autonomous networks – Architecture framework,” ITU, Recommendation, 2023. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3061-202312-I

[15] ETSI, “Zero-touch network and Service Management (ZSM); Reference Architecture,” ETSI, Group Specification, Aug. 2019. [Online]. Available: {https://www.etsi.org/committee/zsm}

[16] H. Kokkonen *et al.*, “Autonomy and intelligence in the computing continuum: Challenges, enablers, and future directions for orchestration,” *ArXiv Prepr. ArXiv220501423*, 2022.

[17] E. Zeydan and Y. Turk, “Recent advances in intent-based networking: A survey,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp. 1–5.

[18] A. Leivadeas and M. Falkner, “A survey on intent-based networking,” *IEEE Commun. Surv. Tutor.*, vol. 25, no. 1, pp. 625–655, 2022.

[19] S. Ertaş, M. Babaei, M. Ziauddinm, D. Roeland, S. Das, and M. Saimler, “Empowering 6G Communication Service Providers: A Seamless SLA-to-Intent Translation

Framework," in *2025 28th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, IEEE, 2025, pp. 166–172.
[20] D.-Z. Zeng, L. Fang, K. W. Hipel, and D. M. Kilgour, "Policy stable states in the graph model for conflict resolution," *Theory Decis.*, vol. 57, pp. 345–365, 2004.
[21] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM workshop on hot topics in networks*, 2016, pp. 50–56.
[22] S. Wang, Q. Zhuo, H. Yan, Q. Li, and Y. Qi, "A network traffic prediction method based on LSTM," *ZTE Commun.*, vol. 17, no. 2, p. 19, 2019.
[23] D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs, and X. Costa-Perez, "Network slicing meets artificial intelligence: An AI-based framework for slice management," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 32–38, 2020.
[24] E. Edozie, A. N. Shuaibu, B. O. Sadiq, and U. K. John, "Artificial intelligence advances in anomaly detection for telecom networks," *Artif. Intell. Rev.*, vol. 58, no. 4, p. 100, 2025.
[25] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends® Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.
[26] H. Zhang, Y. Zhao, C. Yang, A. Farhan, and F. Johnston, "Towards Secure and Private AI: A Framework for Decentralized Inference," in *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*,
[27] M. T. Hammi, P. Bellot, and A. Serhrouchni, "BCTrust: A decentralized authentication blockchain-based mechanism," in *2018 IEEE wireless communications and networking conference (WCNC)*, IEEE, 2018, pp. 1–6.
[28] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-art and Research Challenges," *ACM Comput Surv*, vol. 56, no. 3, Oct. 2023, doi: 10.1145/3625558.
[29] C. Zhou *et al.*, "Network Digital Twin: Concepts and Reference Architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-10, Feb. 2025. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/10/
[30] "Recommendation ITU-T Y.3092 (09/2024) - Digital twin for management and orchestration in IMT-2020 networks and beyond".
[31] TM FORUM, "Autonomous Networks: Empowering digital transformation for smart societies and industries," TM FORUM, White Paper, Oct. 2020. [Online]. Available: {https://www.tmforum.org/resources/whitepapers/autonomous-networks-empowering-digital-transformation-for-smart-societies-and-industries/}
[32] S. Yoon, "Building digital twinning: Data, information, and models," *J. Build. Eng.*, vol. 76, p. 107021, Oct. 2023, doi: 10.1016/j.jobe.2023.107021.
[33] M. Van Den Brand, L. Cleophas, R. Gunasekaran, B. Haverkort, D. A. M. Negrin, and H. M. Muctadir, "Models Meet Data: Challenges to Create Virtual Entities for Digital Twins," in *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, Fukuoka, Japan: IEEE, Oct. 2021, pp. 225–228. doi: 10.1109/MODELS-C53483.2021.00039.
[34] I. Vilà, O. Sallent, and J. Pérez-Romero, "On the Design of a Network Digital Twin for the Radio Access Network in 5G and Beyond," *Sensors*, vol. 23, no. 3, p. 1197, Jan. 2023, doi: 10.3390/s23031197.
[35] S. Si-Mohammed, A. Bardou, T. Begin, I. Guérin Lassous, and P. Vicat-Blanc, "NS+NDT: Smart integration of Network Simulation in Network Digital Twin, application to IoT networks," *Future Gener. Comput. Syst.*, vol. 157, pp. 124–144, Aug. 2024, doi: 10.1016/j.future.2024.03.038.
[36] H. Song, F. Qin, P. Martinez-Julia, L. Ciavaglia, and A. Wang, "Network Telemetry Framework." May 2022. doi: 10.17487/RFC9232.
[37] A. Sayin, Mohammedreza B., Fuladi Ramin, and U. Gülen, "Secure and privacy preserved data management system for 6G Network Digital Twin," in *14th Computing Conference 2026)*, SAI, 2025, p. 1.

[38] C. Qian, X. Liu, C. Ripley, M. Qian, F. Liang, and W. Yu, "Digital twin—Cyber replica of physical things: Architecture, applications and future research directions," *Future Internet*, vol. 14, no. 2, p. 64, 2022.
[39] Y. Wang, Z. Su, S. Guo, M. Dai, T. H. Luan, and Y. Liu, "A survey on digital twins: Architecture, enabling technologies, security and privacy, and future prospects," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 14965–14987, 2023.
[40] A. Raghuramu *et al.*, "Network Digital Twins: A Threat Analysis," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2023, pp. 733–739.
[41] C. L. Stergiou, E. Bompoli, and K. E. Psannis, "Security and privacy issues in IoT-based big data cloud systems in a digital twin scenario," *Appl. Sci.*, vol. 13, no. 2, p. 758, 2023.
[42] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *ArXiv Prepr. ArXiv170205659*, 2017.
[43] V. Zangirolami and M. Borrotti, "Dealing with uncertainty: Balancing exploration and exploitation in deep recurrent reinforcement learning," *Knowl.-Based Syst.*, vol. 293, p. 111663, 2024.
[44] T. Blochwitz *et al.*, "Functional Mockup Interface 2.0: The Standard for Tool Independent Exchange of Simulation Models," in *9th International Modelica Conference*, The Modelica Association, 2012, pp. 173–184.
[45] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)," IEEE, Std 1516–2010.
[46] M. Yaqoob, R. Trestian, M. Tatipamula, and H. X. Nguyen, "Digital-Twin-Driven End-to-End Network Slicing Toward 6G," *IEEE Internet Comput.*, vol. 28, no. 2, pp. 47–55, 2024, doi: 10.1109/MIC.2023.3332252.
[47] M. LLC, "5G System Overview." Accessed: Jun. 15, 2025. [Online]. Available: https://www.3gpp.org/technologies/5g-system-overview
[48] N. M. Akshatha, P. Jha, and A. Karandikar, "A Centralized SDN Architecture for the 5G Cellular Network," in *2018 IEEE 5G World Forum (5GWF)*, 2018, pp. 147–152. doi: 10.1109/5GWF.2018.8516960.
[49] Z. Zhang *et al.*, "6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, 2019, doi: 10.1109/MVT.2019.2921208.
[50] 6G-TWIN Project, "Deliverable D2.1: Data governance, privacy and harmonization," 6G-TWIN Project, 2024. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[51] ETSI, "Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 1: Enablers," no. GS ZSM 009-1. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/00901/01.01.01_60/gs_ZSM00901v010101p.pdf
[52] IBM, "An architectural blueprint for autonomic computing," IBM, White Paper, Jun. 2006.
[53] J. R. Boyd, "The Essence of Winning and Losing." [Online]. Available: https://ooda.de/media/john_boyd_-_the_essence_of_winning_and_losing.pdf
[54] J. Strassner, N. Agoulmine, and E. Lehtihet, "Focale: A novel autonomic networking architecture," 2006.
[55] ETSI, "Zero-touch network and Service Management (ZSM); Intent-driven Closed Loops," no. GS ZSM 016. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/016/01.01.01_60/gs_ZSM016v010101p.pdf
[56] ETSI, "Zero-touch network and Service Management (ZSM); Enablers for Artificial Intelligence-based Network and Service Automation," no. GS ZSM 012. 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/012/01.01.01_60/gs_ZSM012v010101p.pdf
[57] T. Kreuzer, P. Papapetrou, and J. Zdravkovic, "Artificial intelligence in digital twins—A systematic literature review," *Data Knowl. Eng.*, vol. 151, p. 102304, 2024, doi: https://doi.org/10.1016/j.datak.2024.102304.

[58] ITU-R, “Framework and overall objectives of the future development of IMT for 2030 and beyond,” ITU, Recommendation, 2023. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2160-0-202311-I!!PDF-E.pdf
[59] 3GPP, “Management and orchestration; 5G end to end Key Performance Indicators (KPI),” no. TS 28.554. 2025. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/28_series/28.554/28554-j30.zip
[60] 6G-TWIN Project, “Deliverable D1.2: Secured, scalable, and distributed data exposure and collection framework (initial).” Accessed: Jun. 15, 2025. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[61] 6G-TWIN Project, “Deliverable D5.1: Evaluation methodology, planning and coordination.” Accessed: Jun. 15, 2025. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[62] IBM, “What is an SLA (service level agreement)?” Accessed: Jun. 15, 2025. [Online]. Available: https://www.ibm.com/think/topics/service-level-agreement
[63] Y. Ren, S. Guo, B. Cao, and X. Qiu, “End-to-End Network SLA Quality Assurance for C-RAN: A Closed-Loop Management Method Based on Digital Twin Network,” *IEEE Trans. Mob. Comput.*, vol. 23, no. 5, pp. 4405–4422, 2024, doi: 10.1109/TMC.2023.3291012.
[64] ETSI, “Zero-touch network and Service Management (ZSM); ZSM security aspects,” no. GS ZSM 014. 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/014/01.01.01_60/gs_ZSM014v010101p.pdf
[65] ETSI, “Zero-touch network and Service Management (ZSM); Closed-Loop Automation; Part 2: Solutions for automation of E2E service and network management use cases,” no. GS ZSM 009-2. 2022. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/00902/01.01.01_60/gs_ZSM00902v010101p.pdf
[66] ETSI, “Zero-touch network and Service Management (ZSM); General Security Aspects,” no. GS ZSM 010. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/010/01.01.01_60/gr_ZSM010v010101p.pdf
[67] ETSI, “Zero-touch network and Service Management (ZSM); Intent-driven autonomous networks; Generic aspects,” no. GS ZSM 011. 2023. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/ZSM/001_099/011/01.01.01_60/gr_ZSM011v010101p.pdf
[68] IBM, “What is AI agent orchestration?” Accessed: Jun. 15, 2025. [Online]. Available: https://www.ibm.com/think/topics/ai-agent-orchestration
[69] R. Vrabič, J. A. Erkoyuncu, M. Farsi, and D. Ariansyah, “An intelligent agent-based architecture for resilient digital twins in manufacturing,” *CIRP Ann.*, vol. 70, no. 1, pp. 349–352, 2021, doi: https://doi.org/10.1016/j.cirp.2021.04.049.
[70] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, “SliceOps: Explainable MLOps for Streamlined Automation-Native 6G Networks,” *IEEE Wirel. Commun.*, vol. 31, no. 5, pp. 224–230, 2024, doi: 10.1109/MWC.007.2300144.
[71] T. Zanouda, M. Masoudi, F. G. Gebre, and M. Dohler, “Telecom Foundation Models: Applications, Challenges, and Future Trends,” *ArXiv Prepr. ArXiv240803964*, 2024.
[72] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, “Large generative ai models for telecom: The next big thing?,” *IEEE Commun. Mag.*, vol. 62, no. 11, pp. 84–90, 2024.
[73] A. Mekrache, A. Ksentini, and C. Verikoukis, “Intent-based management of next-generation networks: An LLM-centric approach,” *Ieee Netw.*, 2024.
[74] S. Xu, C. K. Thomas, O. Hashash, N. Muralidhar, W. Saad, and N. Ramakrishnan, “Large multi-modal models (LMMs) as universal foundation models for AI-native wireless systems,” *IEEE Netw.*, 2024.
[75] L. Huang *et al.*, “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions,” *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, 2025.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

[76] J. Wei *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24824–24837, 2022.
[77] 6G-TWIN Project, “Deliverable D1.1: 6G-TWIN architecture and technical foundations (initial),” 6G-TWIN Project, 2024. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[78] X. Lin, L. Kundu, C. Dick, E. Obiodu, T. Mostak, and M. Flaxman, “6G Digital Twin Networks: From Theory to Practice,” *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 72–78, Nov. 2023, doi: 10.1109/MCOM.001.2200830.
[79] H. M. Muctadir, D. A. Manrique Negrin, R. Gunasekaran, L. Cleophas, M. van den Brand, and B. R. Haverkort, “Current trends in digital twin development, maintenance, and operation: an interview study,” *Softw. Syst. Model.*, vol. 23, no. 5, pp. 1275–1305, Oct. 2024, doi: 10.1007/s10270-024-01167-z.
[80] 6G-TWIN Project, “Deliverable D2.2: Basic models (initial),” 6G-TWIN Project, 2024. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[81] 6G-TWIN Project, “Deliverable D2.3: Functional models (initial).” Accessed: Jun. 15, 2025. [Online]. Available: https://6g-twin.eu/resources/#deliverables
[82] ETSI, “Network Functions Virtualisation (NFV) Release 5; Management and Orchestration; Performance Measurements Specification,” ETSI, Group Specification, Sep. 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/nfv-ifa/001_099/027/05.02.01_60/gs_nfv-ifa027v050201p.pdf
[83] BeGREEN, “Beyond 5g Artificial Intelligence Assisted Energy Efficient Open Radio Access Network.” Accessed: Jun. 15, 2025. [Online]. Available: https://www.sns-begreen.com/
[84] M. Catalan-Cid, G. Castellanos, and D. Reiss de Fez, “BeGREEN/6G-TWIN - ML-driven Cell on/off switching dataset (BeGREEN’s WP5 PoC1-UC3).” [Online]. Available: https://doi.org/10.5281/zenodo.15688756
[85] S. Sun *et al.*, “Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5G wireless communications,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 2843–2860, 2016.
[86] B. Feng, M. Zheng, W. Liang, and L. Zhang, “A recent survey on radio map estimation methods for wireless networks,” *Electronics*, vol. 14, no. 8, p. 1564, 2025.
[87] R. Levie, Ç. Yapar, G. Kutyniok, and G. Caire, “RadioUNet: Fast radio map estimation with convolutional neural networks,” *IEEE Trans. Wirel. Commun.*, vol. 20, no. 6, pp. 4001–4015, 2021.
[88] D. De Vleeschauwer, C.-Y. Chang, P. Soto, Y. De Bock, M. Camelo, and K. De Schepper, “A Method to Compare Scaling Algorithms for Cloud-Based Services,” *IEEE Trans. Cloud Comput.*, 2024.
[89] P. Soto *et al.*, “Network Intelligence for NFV scaling in closed-loop architectures,” *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 66–72, 2023.
[90] P. Soto *et al.*, “Towards autonomous VNF auto-scaling using deep reinforcement learning,” in *2021 Eighth International Conference on Software Defined Systems (SDS)*, IEEE, 2021, pp. 01–08.
[91] A. Khan, “Network Traffic DataSet.” 2019. [Online]. Available: https://www.kaggle.com/datasets/asfandyar250/network
[92] K. Ismailaj, M. Camelo, and S. Latré, “When Deep Learning May Not Be The Right Tool For Traffic Classification,” in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 884–889.
[93] M. Camelo, P. Soto, and S. Latré, “A General Approach for Traffic Classification in Wireless Networks Using Deep Learning,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 4, pp. 5044–5063, 2022, doi: 10.1109/TNSM.2021.3130382.
[94] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, “Mobile Encrypted Traffic Classification Using Deep Learning: Experimental Evaluation, Lessons Learned, and Challenges,” *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 2, pp. 445–458, 2019, doi: 10.1109/TNSM.2019.2899085.

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Smart Networks and Services Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.