Received 12 December 2025; revised 13 February 2026; accepted 20 February 2026. Date of publication 26 February 2026; date of current version 4 March 2026.

Digital Object Identifier 10.1109/OJCOMS.2026.3668035

A Reference Functional Architecture for Network Digital Twins in 6G Systems

AYAT ZAKI-HINDI[1], PAOLA SOTO[2], GERMAN CASTELLANOS[3], TOUHID HOSSAIN PRITOM[4], GAETANO VOLPE[5], ISKANDER ZELLAGUI[6], BURKHARD HENSEL[4], JULIAN JIMENEZ[2], MICHELE MARVULLI[5], LUÍS SANTOS[7], AYSE SAYIN[8], JEAN-SÉBASTIEN SOTTET[1], WASIM ALI[5], INES EL-KORBI[6], ION TURCANU[1], ANDREY BELOGAEV[2], ANDRÉ DUARTE[7], ULTAN KELLY[9], SUMIT KUMAR[1], GEORGY MYAGKOV[10], STEPHEN PARKER[3], MARIO FRANKE[4], SHAJJAD HOSSAIN[6], RAJARSHI SANYAL[11], NIDA SHAFI[10], CHRISTOPH SOMMER[4], MIGUEL CAMELO[2], JULIEN BAUDOUIN[11], RÉGIS DECORME[12], MARIA PIA FANTI[5] (Life Fellow, IEEE), RAMIN FULADI[8], CHRIS MURPHY[10], ANA PEREIRA[7], SIDI MOHAMMED SENOUCI[6], SIMON PRYOR[3], AND SÉBASTIEN FAYE[1]

[1]Luxembourg Institute of Science and Technology (LIST), 4362 Esch-sur-Alzette, Luxembourg
[2]IDLab, imec, University of Antwerp, 2000 Antwerpen, Belgium
[3]Accelleran, 2018 Antwerpen, Belgium
[4]Faculty of Computer Science, TU Dresden, 01069 Dresden, Germany
[5]Politecnico di Bari, 70126 Bari, Italy
[6]Esplanade Erasme, Université de Bourgogne, 21078 Dijon, France
[7]Ubiwhere LDA., 6230-421 Fundão, Portugal
[8]Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri A.Ş., 34469 İstanbul, Türkiye
[9]Viavi Solutions Ireland Ltd., Dublin 4, A96 T927 Ireland
[10]Viavi Solutions U.K. Ltd., SG1 2AN Stevenage, U.K.
[11]Proximus Luxembourg S.A., 8070 Bertrange, Luxembourg
[12]R2M Solution, 06330 Roquefort-les-Pins, France

CORRESPONDING AUTHOR: S. FAYE (sebastien.faye@list.lu)

This work was supported by the Integrating Network Digital Twinning into Future AI-based 6G Systems (6G-TWIN) Project from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe Research and Innovation Program under Grant 101136314.

ABSTRACT AI-native, programmable, and disaggregated 6G networks will be highly dynamic and distributed, demanding tools that can explain, predict, and safely optimize behavior across the edge–cloud continuum. Network Digital Twins (NDTs) promise this capability, yet current efforts in research and industry are fragmented and lack widely accepted formal definitions and architectural guidelines. This paper proposes a structured framework for NDTs in 6G, addressing these gaps by refining the conceptual foundations of NDTs, introducing a functional architecture, inherited from the 6G-TWIN EU consortium, and clarifying key components such as AI-driven workflows, the place of simulation, data management, and orchestration. Concrete examples illustrate how these components enable network automation, optimization, and predictive analytics. The paper proceeds by reviewing related work and standardization efforts, specifying functional and non-functional requirements, presenting the architecture and its various domains, and detailing lifecycle management across cloud to edge. We then report early implementations and evaluation results, and discuss security, privacy, and governance considerations, concluding with directions for validation and uptake. The key objective is to offer a cohesive reference model that guides the community in shaping NDT development, ensuring interoperability, scalability, adaptability, and seamless integration into AI-native 6G networks for improved intelligence and efficiency.

INDEX TERMS 6G, AI-native network orchestration, closed-loop network automation, edge–cloud continuum management, network digital twin.

I. INTRODUCTION

FUTURE 6G networks will operate under unprecedented levels of complexity, driven by ultra-dense deployments, heterogeneous access technologies, stringent latency and reliability constraints, and pervasive intelligence at the network edge. The number of connected applications and services will far exceed those of previous generations, with traffic demand expected to be nearly ten times higher than when 5G was first introduced in 2018 [1]. These trends will also introduce new constraints from verticals such as connected mobility, digital health, and public protection and disaster relief. Each of these domains will impose distinct requirements on reliability, latency, and distributed intelligence across the edge-cloud continuum, while ensuring energy efficiency and seamless operation across diverse and dynamic network environments.

To manage such systems, network operators will require continuous visibility, predictive control, and automated assurance across the entire service lifecycle, from design and planning to real-time operation and optimization. Conventional management frameworks, largely static and rule-based, can no longer cope with the dynamics, heterogeneity, and scale of next-generation networks. Instead, networks must evolve toward data- and model-driven operation, combining predictive intelligence (i.e., anticipating behaviors, trends, and anomalies beyond the limits of traditional simulation) with real-time adaptive control (i.e., ensuring continuous optimization and resilience under rapidly changing conditions).

In this context, Network Digital Twins (NDTs) are emerging as a cornerstone of next-generation network management. An NDT is a virtual, continuously synchronized replica of the physical network that captures its topology, configuration, performance, and environment. Through bidirectional data exchange, the NDT mirrors the state of the live network, enabling real-time monitoring, predictive analytics, "what-if" experimentation, and closed-loop optimization. NDTs thus act as cognitive mirrors and safe sandboxes of the network to test policies, train Artificial Intelligence (AI) models, and anticipate failures before they affect live services.

The integration of NDTs into 6G architectures is therefore essential. The 6G vision extends beyond connectivity to create intelligent, sustainable, and self-adaptive infrastructures spanning the edge-cloud continuum. Achieving this vision requires a management paradigm in which the network continuously reasons about its own behavior, learns from historical and contextual data, and refines its operation autonomously. NDTs provide the means to achieve this: they support closed-loop learning and optimization at multiple timescales, facilitate the safe introduction of AI-based control, and underpin key functions such as network planning, slice assurance, and energy optimization. Incorporating NDTs into the operational lifecycle of 6G, including design, deployment, operation, and assurance, ensures that the network remains both agile and trustworthy as conditions evolve.

However, orchestrating NDTs efficiently across distributed 6G infrastructures introduces significant challenges [2]. The heterogeneity of data sources, compute nodes, and administrative domains demands federated, context-aware orchestration and semantic interoperability [3]. From a literature perspective, existing efforts remain fragmented: while several standardization bodies (e.g., ITU-T SG13, 3GPP, ETSI) and research works have proposed concepts or domain-specific digital twins, proposals are either too abstract for implementation or too narrow to generalize beyond a particular segment (e.g., RAN, core, or transport). Critically, there is no widely accepted, formal scope and lifecycle for NDTs, limited integration between AI training and simulation pipelines, and insufficient coordination across the edge-cloud continuum. This fragmentation underscores the need for an architecture that explicitly governs orchestration and lifecycle management across the entire stack.

This paper addresses these gaps by proposing a comprehensive functional architecture for NDTs in 6G systems, designed to enable scalable, interoperable, and AI-driven orchestration from cloud to edge. Building upon our previous work [4], which introduced the first conceptual framework from the Horizon Europe 6G-TWIN project [5], [6], [7], [8], [9], [10], [11], we extend and consolidate that vision through four key advancements:

- We formalize a working definition, scope, and lifecycle for NDTss in 6G, consolidating fragmented literature/industry efforts and deriving functional and non-functional requirements to guide design and evaluation.
- We present a reference functional architecture, decomposed into application, physical, digital, and management domains, with clear interfaces and a harmonized data pipeline.
- We specify lifecycle-aware procedures for NDT instances (instantiation, synchronization, model selection/update, federation, and safe actuation) that operate coherently across the full cloud–edge continuum.
- We integrate AI and simulation via coordinated MLOps and co-simulation workflows, distinguishing basic and functional models to support explainable "what-if" experimentation and closed-loop optimization.
- We validate the approach through early implementations, showcasing different but complementary assets of the NDTs while discussing security, privacy, and data governance.

Beyond the proposed functional architecture, the paper also introduces preliminary implementation results demonstrating early NDT deployments that focus on specific architectural aspects such as data harmonization, AI-based modeling, and simulation-driven optimization. Furthermore, we outline deployment considerations critical to ensuring secure, privacy-preserving, and sustainable integration of NDTs into operational networks.

The remainder of the paper is structured as follows. Section II reviews related work and existing architectural proposals. Section III presents functional and non-functional requirements, together with our proposal for an AI-native functional architecture that integrates NDT as a central element. Sections IV to VI detail the application, physical, digital, and management domains. Section VII discusses NDT lifecycle management and mapping to representative 6G scenarios. Section VIII discusses implementation details and evaluation results. Section IX concludes the paper by

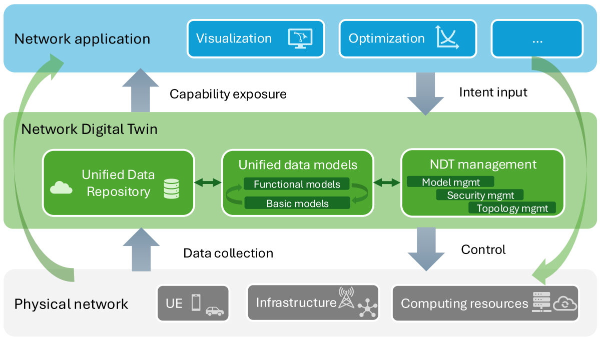

FIGURE 1. **Baseline reference architecture from ITU-T.**

addressing security, privacy, governance, and future research directions.

II. RELATED WORK

This section reviews the current state of NDT architectures as defined and explored by various technology-shaping entities, including standardization bodies, academic research, and industrial initiatives. The discussion highlights each sector's key contributions, conceptual orientations, and distinguishing features.

A. STANDARD DEVELOPMENT ORGANIZATIONS (SDOS)

The International Telecommunication Union - Telecommunication Standardization Sector (ITU-T), through its recommendation Y.3090 [12], has established one of the baseline reference architectures for NDTs, as abstracted in Fig. 1. The framework is structured into three layers: the application layer, the Digital Twin (DT) layer, and the physical network layer. The *physical layer* includes all real and virtualized elements of the network, such as gNodeB (gNB), User Equipments (UEs), spectrum resources, and core network functions. The *DT layer* defines the essential NDT functionalities, hosting the data, models, and management entities necessary to represent and control the network. It integrates components such as the Unified Data Repository (UDR), Unified Data Model repository (UDM), and DT entity management. The *application layer* operates above, guiding the creation of NDT instances in response to application-specific requirements and transmitting control commands toward the physical network via the twin.

The ITU-T identifies four key enablers underpinning the architecture: data, mapping, models, and interfaces. *Data* constitutes the foundation, providing accurate and continuously updated information within a unified repository. *Mapping* mechanisms ensure real-time synchronization between the digital and physical domains, distinguishing NDTs from traditional simulation systems. *Models* within the DT embody both basic and functional aspects of the physical network entities. Finally, standardized *interfaces* ensure scalability and interoperability, where southbound interfaces connect the DT to the physical network, and northbound interfaces link it to external applications and management systems.

Other standardization entities have also advanced NDT-related initiatives. The 3rd Generation Partnership Project (3GPP), within the technical report 28.915 [13], investigates how NDTs can integrate with network automation and management functions. This includes the role of NDTs in enabling predictive and closed-loop automation and in supporting specific use cases relevant to future 6G networks. Likewise, the Open RAN Alliance (O-RAN) Alliance recognizes NDTs as a pivotal enabler for next-generation Radio Access Network (RAN) intelligence and automation, emphasizing their application to RAN optimization and Lifecycle Management (LCM) [14].

The European Telecommunications Standards Institute (ETSI) [15] and the Internet Engineering Task Force (IETF) [16] provide complementary architectural viewpoints consistent with ITU-T's recommendations. ETSI outlines how NDTs can be incorporated into the Zero-touch Service and network Management (ZSM) framework, highlighting their role in enabling automated, closed-loop service orchestration. It details the necessary functional components, standardized interfaces, and architectural capabilities to integrate NDTs within ZSM-based management systems. Conversely, the IETF proposes a reference architecture that delineates NDT functional components, their interactions, and the corresponding interfaces to ensure seamless coupling with live network infrastructure. Its approach emphasizes the use of NDTs for real-time modeling, analytics, simulation, and feedback mechanisms to enhance decision-making, assurance, and LCM across network operations.

B. INDUSTRY

The telecommunications industry has increasingly embraced the concept of NDTs as a key enabler for managing the growing complexity of network infrastructures. Industrial stakeholders recognize NDTs as tools that enhance operational efficiency, support predictive maintenance, and facilitate data-driven network planning and automation. The following examples illustrate how leading vendors and operators are advancing NDT-related technologies.

A prominent contribution from the industrial domain is presented in Spirent's white paper [17], which conceptualizes NDTs as combined software and hardware emulations of real 5G networks. This approach enables iterative prototyping, validation, and performance assurance of network configurations. Spirent's proposed architecture emphasizes modularity, allowing flexible testing and integration of different network components and functions. Similarly, ZTE's white paper [18] introduces an evolved NDT architecture incorporating an additional service layer that aggregates functionalities into microservices. This design enhances scalability and maintainability by structuring nine independent microservices, each encapsulating distinct capabilities, within a framework that remains agnostic to programming languages, data storage formats, or underlying architectures.

Ericsson's work explores the convergence of NDTs and AI/Machine Learning (ML) technologies within 5G and future 6G RANs [19]. The company emphasizes the importance of standardization, particularly through collaboration with 3GPP, to define common interfaces, data flows, and control mechanisms for AI/ML-enabled functions. Such efforts aim to ensure interoperability across vendors and to pave the way for AI-native cellular networks capable of autonomous optimization. Ericsson has also demonstrated practical advancements in collaboration with Deutsche Telekom and Google Cloud, deploying a cloud-native 5G Core on Google Distributed Cloud Edge to improve deployment speed, efficiency, and compliance with EU data protection regulations [20].

Nokia's vision for AI-native 6G integrates AI/ML directly into the air interface, enabling adaptive radio systems that dynamically adjust to network and environmental conditions [21]. Its *Dynamic Digital Twin* initiative exemplifies this approach, creating continuously updated virtual representations of networks to support predictive operations, fault prevention, and adaptive model optimization [22]. These real-time capabilities enhance both network resilience and management efficiency.

Huawei similarly advocates for AI-centric architectures in the evolution toward 6G networks. Its "Intelligent World 2030" strategy highlights the role of DTs in achieving self-learning, predictive, and self-optimizing networks [23], [24]. By embedding AI throughout the network stack: from perception to control, Huawei envisions intelligent systems capable of autonomous management and sustainable energy-efficient operation, further reinforcing the convergence of NDT and AI-native paradigms in future communication networks.

Keysight positions the NDT as a dynamic, real-time virtual replica of the physical network that evolves alongside its physical counterpart across the entire lifecycle [25]. In contrast to conventional simulators, Keysight emphasizes continuous synchronization with real-world measurements, enabling closed-loop validation, prediction, and optimization. A concrete example is the 6G AI neural receiver use case, where ray-traced channel models and emulation tools are used to generate standard-compliant synthetic data for training and validating AI-based air interface components in realistic O-RAN setups [25]. This illustrates the role of the NDT as a sandbox for experimentation, performance assurance, and AI-native system evolution toward 6G.

C. ACADEMIA

Academic research began exploring the application of AI/ML techniques to communication networks well before the emergence of the NDT concept. As early as 2015, Wang et al. [26] published a comprehensive survey detailing how AI-based methods can address challenges in management, optimization, and maintenance of heterogeneous and fast-evolving mobile networks. The use of the term "Digital Twin (DT)" in the context of networks appeared later, with Dong et al. [27] introducing a framework that employs Deep Neural Networks (DNNs) to enhance energy efficiency in Mobile Edge Computing systems, targeting both latency-critical and delay-tolerant services.

Subsequent academic studies have proposed alternative perspectives on NDTs compared to those defined by Standard Development Organizations (SDOs). For example, Zhou et al. [28] introduced a hierarchical DT framework for satellite communication networks. Their design deploys edge-DTs co-located with ground stations, each maintaining a model of its associated physical entity, comprising satellites, terminals, and wireless links. These edge-DTs perform real-time tasks such as beam scheduling, fault diagnosis, resource allocation, and data processing, while exchanging information with central-DTs hosted in a centralized control facility. Central-DTs oversee global network management and can create multiple, isolated virtual instances for verification, optimization, traffic engineering, and slicing control. By ensuring that each physical asset maps to a single edge-DT, the architecture effectively minimizes resource redundancy and system overhead.

Several studies have also concentrated on RAN-centric NDT architectures. Vilá et al. [29] proposed a RAN-focused framework aligned with both O-RAN Alliance and IETF guidelines [16]. Their architecture is structured around three main components: a data repository, a suite of service mapping models (including both basic and functional models), and a twin management entity. The design allows each application to operate its own RAN NDT instance, customized according to specific requirements such as active service mapping models, target Key Performance Indicators (KPIs), data selection rules, and visualization or emulation capabilities.

Kuruvatti et al. [30] offered a comprehensive survey summarizing requirements, challenges, and use cases for NDTs. Their work synthesizes recommendations from both the IETF and ITU-T, identifying four fundamental components of a DT: data, mapping, model, and interface. They propose a three-layer, three-domain, and dual closed-loop architecture consistent with the principles outlined in ITU-T Recommendation Y.3090 [12].

Beyond network-specific proposals, broader DT literature provides complementary foundations relevant to NDTs. Minerva et al. [31] survey DTs architectures and synchronisation assumptions applicable to networking/IoT systems, while Lehner et al. [32] detail platform requirements to operationalise DTs at scale (e.g., lifecycle management, interoperability, and model/data governance). Recent work further emphasises dependability aspects for operational use: Bicocchi et al. [33] propose a blueprint for end-to-end DT trustworthiness, and Bellavista et al. [34] introduce the Overall Digital Twin Entanglement (ODTE) metric to characterise how strongly a DT remains coupled to its physical counterpart, based on the timeliness and completeness of the collected data. Overall, these perspectives motivate treating

TABLE 1. DT/NDT architectural perspectives and positioning of this work.

Perspective	Focus	Strengths	Typical scope	Gaps for operational NDT	Contribution of this work
SDOs	Reference architectures; interfaces	Interoperability; common baseline	Cross-domain (high-level)	Executable data models; AI workflow integration; simulation and lifecycle management	Operational integration of data, AI, and simulation workflows for executable NDT realization with lifecycle management.
Industry	Deployable solutions	Scalable stacks; mature tooling	Vendor- or RAN-centric deployments	Cross-domain abstraction; unified model lifecycle; standardized AI integration	Cross-domain NDT lifecycle with model instantiation, training, validation, and orchestration.
Academ. (NDT)	Prototype NDT designs	Conceptual innovation; experimentation	Use-case or domain specific	End-to-end integration; operational management; traceable model lifecycle	MANO-driven operational framework integrating basic and functional models with structured data pipelines.
Academ. (DT)	Generic DT concepts	Fidelity and trust metrics; formal requirements	Cross-domain, non-network-specific	Telecom constraints; network entity modeling; orchestration aspects	Mapping DT principles to telecom-specific entities, KPIs, and lifecycle-driven NDT orchestration.

modelling, architecture, and trust-related properties as first-class concerns in practical NDT design.

D. DISCUSSION

Table 1 showcases a concise comparison of the perspectives reviewed in this section, and highlights how the present work addresses the main gaps identified.

The work on NDT architectures and their interaction with the physical network has been mainly led by prominent SDOs such as ITU-T [12] and IETF [16]. While agreeing on the main building blocks and NDT capabilities such as mapping, models, and data, both works lack in detailing the interaction between the physical and the digital network, how the data should be collected, how AI and non-AI models should be created, and what the role of simulators is within the NDT.

Industry and most research work in academia focus on the application of NDTs within the RAN domain [29], [35], [36], as this represents one of the most intricate and resource-intensive components of 6G infrastructure. In general, we can see that existing works lack a unified, practical framework, often being too abstract or too narrow. Key aspects like AI integration, automation, and scalability remain insufficiently addressed, motivating our proposal for a more concrete and adaptable NDT architecture.

Some studies [37], [38], [39] propose AI-native architectures for future communication networks and acknowledge the potential role of NDTs in supporting network operations; however, they do not provide detailed methodologies for their integration or explicitly incorporate NDTs within the proposed architectural frameworks.

Finally, our previous work presents an initial AI-native network architecture that incorporates NDTs to optimize, manage, and control future networks in real-time [40] and its updated version [4]. While the initial architecture was heavily inspired by the ITU-T, the updated version included four key pillars: (i) a data collection framework for dynamic data acquisition and harmonization, (ii) ZSM for AI-driven automation, (iii) Federated Management for decentralized network control, and (iv) the simulation framework for predictive modeling.

The architecture presented in this paper, as it will be shown in Section III-B, further refines the architecture to introduce an NDT MANO layer as a unifying vertical layer that integrates all the aforementioned domains. This layer not only incorporates the MANO functions of both the physical and digital domains, including federated management, simulation, and AI workflows, but also establishes a clear structural boundary between the core NDT components and the instantiated NDT subcomponents, thereby defining the operational interface between the digital and physical domains.

III. SYSTEM DESIGN

When creating a software tool, one of the first things to do is to list the Functional Requirements (FRs) and Non-Functional Requirements (NFRs). This is not different for an NDT since it lays the foundation for aligning the NDT with the needs of users, stakeholders, and the technical challenges posed by 6G environments, making this phase extremely important.

A. NDT FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

To ensure effective implementation, an NDT must accurately model and represent the physical network in real time, relying on unified data models and a harmonized data repository. It must support full instance lifecycle management, seamless integration with AI/ML components, and operate either as an analytical or controlling twin within closed-loop systems. Federated cooperation between multiple NDT instances is also required.

To do so, the NDT system should have a MANO layer, enhanced with ZSM, that orchestrates NDT lifecycle processes including creation, deployment, updates, and deletion, while supporting AI-based workflows and standardized Application Programming Interfaces (APIs). Similarly, to support what-if capabilities, the NDT should integrate and synchronize heterogeneous simulators, manage configurations, and maintain closed-loop interoperability with other system components.

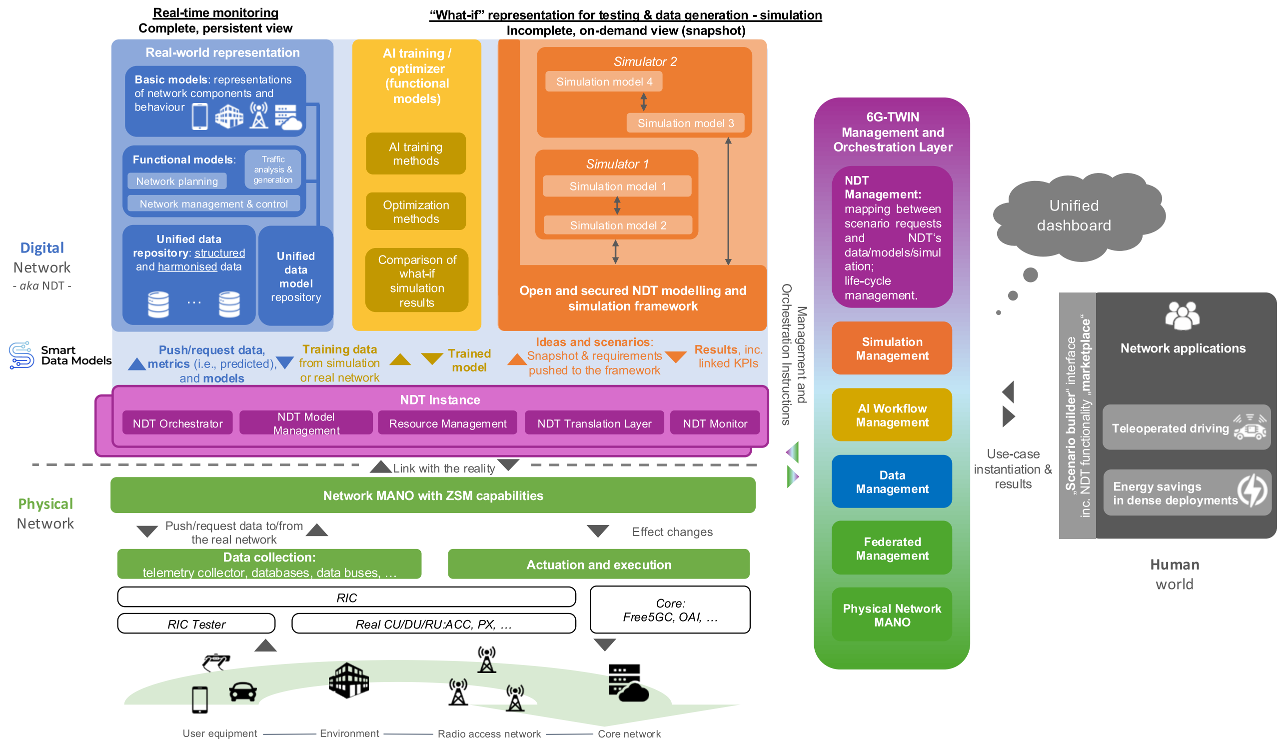

FIGURE 2. 6G NDT functional architecture.

Representing the physical network with high-fidelity cannot occur without a robust data collection framework that ensures multi-domain data integration across the Cloud-to-Far-Edge continuum, harmonization of heterogeneous data formats, and securing, protocol-agnostic communication with connected devices. Similarly, all the processes and interactions of the NDT with its physical counterpart should follow the automation principles outlined by a ZSM framework who discovers, monitors, and orchestrates network resources, while ensuring interoperability and secure communications via standardized APIs and Authorisation/Authentication/Accounting mechanisms. Regarding the non-functional requirements, they define the performance, scalability, and reliability constraints of the NDT architecture. The NDT architecture must guarantee scalability, low latency, reliability, and security, with mechanisms for redundancy, access control, and fault tolerance. Its modular, service-based design should ensure maintainability and compatibility across technologies. Moreover, it must integrate Developer Operations (DevOps), Machine Learning Model Operationalization Management (MLOps), and cloud-native principles to achieve efficient, secure, and flexible orchestration of distributed AI-enabled network functions and simulations in real time.

B. NDT ARCHITECTURE OVERVIEW

Building upon the requirements outlined in Section III-A, Fig. 2 represents the proposed functional architecture, designed as an integrated framework that unifies NDTs with AI-driven functionalities and simulation components under a unified management layer. This functional architecture is designed to operate as a closed-loop system, where data flows continuously between the real-world network, its digital representation, and the management functions that synchronize them. To structure this interaction, the architecture is organized into four interconnected domains: *application*, *physical*, *digital*, and *management*. Each domain fulfills a distinct role while contributing to the end-to-end integration of the NDT.

The application domain contains the dashboard equipped with the interfaces through which humans interact with the NDT. The dashboard serves as the human-in-the-loop entry point, enabling stakeholders to visualize network state, and to configure NDT and monitor NDT operations in real time.

The physical domain comprises the operational infrastructure, including the RAN, Core Network (CN), Transport Network (TN), and edge/cloud computing resources. It plays a dual *sense–act* role in the NDT ecosystem by providing telemetry and contextual data to the digital domain, and then by enforcing optimization decisions, configuration updates, and control actions generated by the NDT.

The digital domain contains the digital representation of the physical network and the mechanisms needed to analyze and predict its behavior. It integrates the data-driven digital representation of the physical network, simulation engines, and AI training pipelines. This domain ensures that the NDT continuously reflects the state of the physical network while enabling predictive modeling, what-if analysis, and the creation of *deployable functional models* that are validated in the NDT before being instantiated in the physical domain through the NDT MANO.

The management domain oversees the instantiation, synchronization, and lifecycle of NDT components. It orchestrates interactions between domains, ensuring that data flows are harmonized, models remain consistent, and simulations or AI workflows are executed efficiently. By coordinating telemetry ingestion, model updates, and decision deployment, this domain is what makes the NDT "alive," transforming static models into an adaptive, self-evolving system.

An important aspect of the proposed architecture is the distinction between the *NDT* and an *NDT instance*, a differentiation that is rarely emphasized in the existing literature. The NDT architecture covers the full set of components, representations, and functionalities that, in principle, could replicate the physical network in all its detail. However, maintaining such a global and continuously accurate digital replica is both costly and unnecessary for most applications. In practice, the NDT is instantiated to create an NDT instance, designed to *perform a specific task for a given time duration*. For example, an application may require an NDT instance dedicated to planning new gNB deployments within a certain geographic area under specific Quality of Service (QoS) constraints. In such cases, only a subset of models and functionalities is activated to serve the task. This distinction is essential: treating the NDT as a single, monolithic instance reduces its value as a flexible, general-purpose architecture, whereas recognizing instances allows the architecture to support multiple, concurrent objectives efficiently. The following sections detail the building blocks within each domain and their role in enabling scalable, adaptive, and AI-driven NDTs.

IV. APPLICATION AND PHYSICAL DOMAINS

The physical and application domains represent the external entities that drive the creation and operation of the NDT instance. Applications express intents that specify what aspects of the physical network infrastructure must be represented and what actions should be performed, thereby determining the scope and configuration of each NDT instance.

A. APPLICATION DOMAIN

The unified dashboard serves as the primary interface between NDT stakeholders and the system. It provides users with the means to interact with both the NDT and, indirectly, the physical network. The range of functionalities exposed through the dashboard depends on the autonomy level of the NDT [41]. In low-autonomy settings, the NDT operates only in the digital domain, and the dashboard is limited to monitoring and analysis tasks. At higher levels of autonomy, the NDT can issue control actions to the physical network, making the dashboard a critical channel for decision-making and oversight.

At its core, the dashboard supports the visualization of the current network state, enabling users to query specific parameters or network elements. More advanced features extend this role by offering predefined models and commands that can be configured or personalized. A further step is the integration of natural-language interfaces, for example, through Large Language Model (LLM)-based chatbots pretrained on domain-specific tasks. In this case, users can describe their goals in natural language, and the chatbot translates these objectives into actionable tasks within the NDT. This direction aligns with emerging research trends in network management, as illustrated in [42], which demonstrates the potential of conversational AI to simplify network operations. Interfaces between the dashboard and the NDT are realized through APIs, which are defined and managed within the NDT MANO.

To ensure reliability, such interfaces can incorporate a verification step, where the chatbot presents its interpretation and planned actions for user confirmation before execution. The dashboard also acts as the NDT output channel. For tasks requiring control over the physical network, different levels of automation are possible. In a conservative mode, the NDT provides recommended solutions, and users retain responsibility for applying them, thus ensuring compliance with organizational policies or regulatory constraints.

In fully autonomous settings, the NDT may apply changes directly, while still presenting a summary of actions to the user. In hybrid approaches, changes are executed only after explicit user approval, although this may limit applicability in real-time scenarios.

B. PHYSICAL DOMAIN

The physical domain constitutes the tangible environment on which the NDT operates and provides feedback and control. As illustrated in Fig. 3, this domain is composed of three main components. The first is the *physical network*, representing the real-world infrastructure, including its relevant entities, interfaces, and resources that the NDT is designed to replicate, monitor, optimize, and when required, control. The second component is the *data collection framework*, which enables the NDT to monitor the state of the physical network by continuously acquiring, processing, and harmonizing operational data to maintain synchronization between the physical and digital worlds. The third component is the *actuation and execution framework*, responsible for translating the insights, recommendations, or control policies generated by the NDT into deployable configurations or algorithms within the live network.

Together, these components establish the basis for closed-loop operation, where the NDT continuously performs a cycle of *sensing* (observing the physical network), *thinking* (analyzing and deciding within the NDT), and *acting* (executing optimized actions in the physical domain), followed by renewed sensing to assess outcomes and refine subsequent decisions. These components are further detailed in the following subsections.

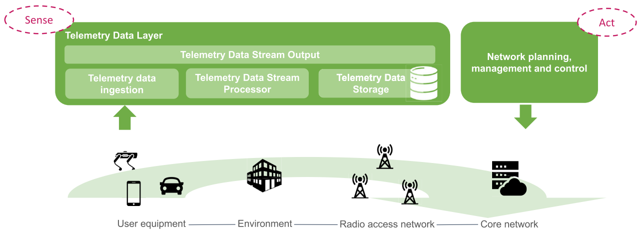

FIGURE 3. **Structure of the physical domain illustrating its components: physical network, data collection, and actuation.**

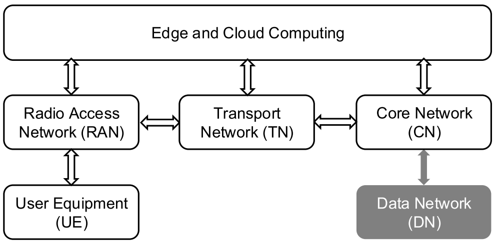

FIGURE 4. **5G and 6G physical network domains.**

1) PHYSICAL NETWORK

To construct the NDT, it is first necessary to define the network domains that the twin must represent. Since the focus of this work is on next-generation systems, where 6G is considered a natural evolution of 5G, the NDT must capture the components that remain fundamental to these networks while accommodating emerging architectural extensions. As illustrated in Fig. 4, the network can be divided into several main domains that collectively form the foundation of end-to-end communication systems.

The RAN and the CN constitute the structural backbone of contemporary 5G deployments and are expected to retain this central role in 6G [43]. These domains are typically managed by Mobile Network Operators (MNOs), who oversee most of the optimization and orchestration activities. Network planning, mobility management, traffic steering, and resource allocation are primarily performed within these domains. Complementing them, the TN interconnects the RAN and CN, ensuring reliable data forwarding and synchronization. Although the TN is not always the primary target of optimization, it plays a vital role in supporting end-to-end capabilities such as network slicing and latency-sensitive service delivery [44]. The edge and cloud computing domains are also essential components of next-generation architectures. As in 5G, these domains enable the processing of large volumes of data closer to where it is generated, reducing latency and improving service efficiency. Their integration is tightly coupled with QoS and Quality of Experience (QoE) guarantees, which are key drivers of future intelligent communication systems. Consequently, they must be represented within the NDT to ensure accurate modeling of distributed computation and data offloading behaviors. At the final stage of the communication chain lies the Data Network (DN), which provides connectivity to external application servers and the broader Internet. Since the DN generally falls outside the administrative control of MNOs, a full-scale digital replication may not be feasible. Nonetheless, aggregated application-level statistics exposed by the CN can serve as proxies to capture the DN's impact on end-to-end performance metrics. Finally, the UE is an essential part of the physical network representation, as many NDT applications target user-centric performance optimization through adaptive and context-aware resource management. Including the UE ensures accurate reproduction and prediction of network behavior from both infrastructure and user perspectives.

Driven by the principles of *network programmability* [45] and *ZSM* [46], the physical network adopts an AI-native architecture that enables data-driven automation and dynamic, closed-loop optimization. It exposes the telemetry data, programmability hooks, and orchestration interfaces needed for the NDT to operate as an analytical and decision-support service or to evolve into a fully controlling component in highly autonomous networks [47]. In this role, the NDT serves as a safe, isolated sandbox where decisions, configurations, and experiments can be evaluated before deployment on the live network, ensuring no impact on its real-time operation or performance. This capability supports the training and validation of AI and non-AI algorithms, while enabling the continuous refinement of decision-making mechanisms. Ultimately, it allows network controllers to dynamically adapt to changing conditions through programmatic, feedback-driven loops.

2) DATA COLLECTION

Data collection is considered one of the essential requirements to build and maintain an accurate digital replica, as specified by several standardization bodies [12], [48]. Data collection must be driven by specific operational objectives rather than performed indiscriminately, as excessive data acquisition increases resource consumption and reduces system efficiency. To address this, operators can define adaptive data collection parameters that respond to network conditions or predefined events, ensuring that only relevant and actionable data is gathered, an essential feature in large-scale networks where bandwidth and storage are limited.

Our approach adopts a multi-layer design comprising a *Telemetry Data Layer (TDL)* in the physical domain and a *Harmonization Data Layer (HDL)* in the digital domain. The TDL captures and pre-processes raw telemetry streams from network interfaces. The HDL, on the other hand, aggregates, harmonizes, and structures these pre-processed datasets into standardized *smart-data models* [49] for use in modeling, simulation, and decision-making. This separation enables telemetry data to be leveraged independently for network performance monitoring, without invoking the full NDT processing chain. Details of the HDL and its integration into the UDR are further discussed in Section V-A.1. The two layers are interconnected via Data Communication Buses (DCBs), which ensure reliable and adaptive data flow between domains through dynamic routing, prioritization, and integrity checking mechanisms.

Focusing on the physical domain, the TDL, as shown in Fig. 3, is subdivided into four key functional elements. The *telemetry data ingestion* collects raw telemetry data from various sources in real-time, forming the initial entry point into the system. This raw data is then passed to the *telemetry data stream processor*, which processes and analyzes the ingested data streams, performing tasks such as filtering, aggregation, and enrichment to generate actionable insights. The processed telemetry data is then stored on the *telemetry data storage*, enabling historical analysis, reporting, and use in future decision-making processes. Finally, the *telemetry data stream output* facilitates the delivery of processed telemetry data to the next stage, which includes preparation for network planning, management, and control.

The data collected by the TDL undergoes an initial phase of pre-processing, filtering, aggregation, and storage to prepare it for further analysis. This process ensures the data is consistent and usable by external applications or different layers within the NDT architecture. A benefit of this pre-processing is that it can reduce communication overhead by eliminating the need to transmit raw data. The software component responsible for data extraction, normalization, and exposure in the NDTs context is a Digital Twin Connector. This connector is a composition of protocols, communication buses, and specific instances of the TDL's ingestion, processing, and output functions. This layer collects real-time data using various protocols, depending on the network domain. For instance, in the management plane, the collected data is used for performance metrics, logs, warnings, and state data, interfacing with network management systems and using protocols such as Simple Network Management Protocol [50], Network Configuration Protocol (NETCONF) [51], among others. In the control plane, the collected data is used to monitor the health of network control protocols, e.g., routing protocols, to aid in real-time issue detection and optimization using protocols such as sFlow [52], [53] and Border Gateway Protocol-Monitoring Protocol [54]. Finally, the data plane telemetry involves extracting and analyzing data from user packets to ensure efficient network operations, balancing collection overhead with traffic processing using protocols such as Alternate Marking Technology [55], In Situ OAM [56], or Packet Sampling [57].

In this sense, the TDL enables the "sense" phase of the Sense–Act cycle by providing continuous telemetry to the upper digital and management layers, which transform this information into actionable insights and decisions. The complementary "act" phase, where these decisions are applied back to the physical network, is described in Section IV-B.3.

3) ACTUATION AND EXECUTION

Once the NDT has been instantiated and validated within the digital domain, its outputs such as optimized configurations, predictive models, or control strategies can be deployed back into the physical network through the NDT MANO. These outputs represent the actionable intelligence derived from the NDT and are used to enhance network operation and management across domains. The process of translating NDT insights into deployable actions, however, differs depending on the characteristics of each network domain defined in Section IV-B.1. 6G systems are expected to integrate multi-vendor infrastructure through O-RAN and rely heavily on AI/ML-based automation. Achieving seamless interoperability and coordinated operation requires advanced Service Management and Orchestration (SMO) capabilities. The following subsections describe how NDT-derived actions and models are applied within each domain, including the RAN, CN, TN, and multi-domain SMO, through their respective control and management mechanisms. It is important to note that while UEs are key elements of network performance optimization, they are not direct targets of actuation. Instead, the NDT actions focus on optimizing the network itself to improve the overall UE experience.

a: RAN ACTUATION

The O-RAN architecture introduces the RAN Intelligent Controller (RIC), a software-based controller that enables intelligent, flexible, and vendor-agnostic management of the RAN [58]. The RIC monitors, controls, and optimizes key radio functions such as resource allocation, interference coordination, power control, and mobility management, and exposes standardized APIs for deploying control applications. The RIC is composed of two hierarchical components that operate over distinct timescales:

- *Non-real-time RIC:* Located in the SMO layer, it operates above one second to perform long-term optimization, policy management, and model training. Its applications, or *rApps*, handle non time-critical tasks such as configuration, radio resource policy generation, and performance forecasting.
- *Near-real-time RIC:* Deployed at the network edge, it operates between 10 ms and 1 s to execute time-sensitive control tasks, including handover, scheduling, and interference mitigation. Its *xApps* interact directly with RAN nodes through standardized interfaces.

Within the proposed NDT architecture, the RIC serves as the RAN actuation interface. Control policies or parameter adjustments derived from functional models in the digital domain are dispatched to either the non- or near-real-time RIC, which applies them through the appropriate rApps or xApps. This establishes a closed-loop cycle in which the RIC enforces NDT-generated decisions while telemetry continuously updates and aligns the digital replica.

b: CORE NETWORK ACTUATION

In the CN, the orchestration and management of Virtual Network Functions (VNFs) are essential for achieving flexibility and scalability in cloud-native network infrastructures. The Network Function Virtualization (NFV)-MANO framework, standardized by ETSI [59], defines how VNFs are deployed, scaled, and maintained across distributed environments.

It consists of three key entities: the NFV Orchestrator, which coordinates network service lifecycles; the VNF Manager, which controls configuration and scaling of individual VNFs; and the Virtualized Infrastructure Manager, which manages compute, storage, and networking resources. Modern orchestration mechanisms increasingly rely on policy-driven automation and AI-assisted optimization [60], [61], enabling dynamic adaptation to traffic conditions and service requirements. Open platforms such as Open Source MANO [62], the Open Network Automation Platform [63], and Cloudify [64], further support closed-loop automation. Within the proposed NDT architecture, functional models continuously analyze CN telemetry to forecast load, detect anomalies, or optimize routing. Their outputs are translated into actionable configurations through the MANO layer which triggers orchestrator-driven adjustments through standardized APIs, establishing a closed-loop control cycle in which NDT-generated insights directly steer policies and service chaining in the live network.

c: TRANSPORT NETWORK ACTUATION

In the transport domain, Software-Defined Network (SDN) separates the control and data planes, enabling programmable and dynamic management of connectivity, bandwidth, and QoS. This flexibility supports network slicing, where multiple logically isolated networks run over the same physical infrastructure, each tailored to service categories such as Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), or massive Machine-Type Communications (mMTC). To coordinate this process, the ETSI slicing management framework [65] defines four key functions. The Communication Service Management Function (CSMF) translates high-level service requirements into slice specifications. The Network Slice Management Function (NSMF) orchestrates slices end-to-end across domains, while the Network Slice Subnet Management Function (NSSMF) manages the instantiation and optimization of slice subnets within each domain. The Network Function Management Function (NFMF) handles lifecycle management of individual network functions. Within the NDT architecture, functional models continuously analyze telemetry from the TN to predict congestion, optimize routing, and anticipate resource bottlenecks. The resulting decisions are translated into reconfiguration actions and enforced through the SDN controller and the NSMF/NSSMF control interfaces. Through this closed-loop integration, the TN evolves from a statically configured backbone into a self-optimizing transport substrate that adapts proactively to service demands.

d: MULTI-DOMAIN ORCHESTRATION

In large-scale 6G environments, services often traverse multiple network domains with distinct control mechanisms and administrative boundaries. The multi-domain SMO framework coordinates orchestration across the RAN, TN, and CN, ensuring coherent control, synchronized operation, and global optimization of network resources. This approach supports end-to-end service provisioning by federating domain-specific functions, monitoring cross-domain performance, and adapting configurations dynamically. It enables unified policy enforcement, service assurance, and optimization across heterogeneous domains, representing a key enabler for autonomous and collaborative network management.

V. DIGITAL DOMAIN

The digital domain represents the central analytical and decision-making environment of the NDT. It comprises modular components that can be instantiated and orchestrated on demand to construct a specific NDT instance. This domain integrates three principal functional dimensions. First, it maintains an *accurate digital representation* of the physical network through the UDR, basic models, and functional models, ensuring coherence between the physical and virtual entities. Second, it provides an *AI training and optimization pipeline*, essential for developing and refining data-driven models. Finally, it incorporates a *simulation environment* that enables predictive analytics, network planning, and validation of trained or calibrated models under diverse operational conditions. Together, these components form the “thinking” layer of the NDT, transforming collected telemetry into actionable knowledge that supports control, optimization, and autonomous decision-making across network domains. The following subsections describe these dimensions in detail.

A. REAL-WORLD REPRESENTATION

The foundation of the digital domain is *data*, which fuels every process from representation to simulation. Within this domain, representation operates along two dimensions. The first concerns the *physical domain*, which is mirrored through *basic models* populated with telemetry data collected from the operational network. This representation captures the topology, configuration, and dynamic state of the physical infrastructure. The second dimension concerns the *application domain*, represented by *functional models* and application-level data, including service intents, operational goals, and target KPIs. Together, these representations ensure that the NDT captures both the operational behavior of the network and the performance requirements that drive its optimization. The following sections detail the components that enable this dual representation.

1) UNIFIED DATA REPOSITORY

The UDR constitutes the main interface between the physical and digital domains and contains the HDL, which consolidates data streams from the TDL into smart-data models, as described in Section IV-B.2. The repository is considered *unified* because it combines network-generated data with contextual datasets from external sources (e.g., geographical information, mobility statistics, environmental conditions), which enhance the relevance and accuracy of digital-domain models for applications such as coverage optimization, mobility prediction, and energy efficiency planning. The UDR is thus responsible for data sharing, harmonization, and storage, as follows.

a: DATA SHARING

Given the heterogeneous and multi-stakeholder nature of 6G networks, data sharing is a core challenge for developing and operating NDTs. Data exchange must ensure privacy, security, interoperability, and regulatory compliance. To support this, we propose a dedicated *data space* to be integrated within the UDR to enable secure and governed data sharing among operators, service providers, and third-party stakeholders.

A data space provides both the organizational framework and technical infrastructure for structured data exchange [66]. Unlike open repositories, it enforces data ownership, access control, and governance policies. Technically, it acts as middleware that enables trusted interactions between participants while enforcing usage policies aligned with the European Data Strategy [67]. Within the NDT architecture, it federates data from multiple domains, allowing NDTs to operate as both data consumers (telemetry and operational data) and data producers (simulation outputs or synthetic datasets). Interoperability is achieved through data connectors and standardized metadata catalogs [68], while identity providers ensure authentication and access control. This guarantees traceable and compliant exchanges. Governance and security aspects are discussed in Section IX, and data harmonization is detailed in Section V-A.1.b.

b: DATA HARMONIZATION

Harmonization is essential to ensure interoperability and consistency across heterogeneous network data sources. Although it introduces processing overhead, particularly in multi-vendor environments, it enables uniform semantics and reusability. In our approach, the harmonized data stored in the UDR is encoded using smart-data models expressed in NGSI-LD, ensuring semantic interoperability, machine readability, and long-term maintainability.

Telemetry data forms the primary input for constructing and updating the network representation, which includes performance indicators, configuration parameters, and fault or event notifications collected from Network Elements (NEs) through standardized interfaces [69], [70], [71]. Our methodology [3] organizes harmonized data into two categories: *attributes*, representing static or configurable parameters [72], and *measurements*, representing dynamic operational metrics [69], [73]. Each NE is mapped to its corresponding set of attributes and measurements, defining its operational profile and enabling consistent interpretation across systems.

The classification and representation of NEs follow 3GPP management standards, which define NEs through *Information Object Classes (IOCs)* [74]. Each IOC specifies the attributes and measurements describing the configuration and behavior of the resource it represents. For example, in the RAN, the gNB is decomposed into the Radio Unit (RU), Distributed Unit (DU), and Centralized Unit (CU), along with finer-grained components such as antenna beams and bandwidth parts. In the CN, VNFs such as the Access and Mobility Management Function (AMF), Session Management Function (SMF), and User Plane Function (UPF) are modeled similarly, each with its own IOC and associated data schema. This modular representation supports scalability, interoperability, and automated management.

Contextual and external data, which complement but do not form part of the main network representation, are incorporated through the data space framework, which ensures harmonized semantic representations across domains.

c: DATA STORAGE

The UDR must support efficient data management, retrieval, and persistence strategies. This involves three main storage types: *data indexing*, *real-time storage*, and *historical storage*. Data indexing organizes and tags incoming data to facilitate efficient querying and retrieval. Real-time storage captures and processes data as it is generated, enabling immediate analytics and operational responses. Historical storage archives past data for trend analysis, validation, and model retraining.

Furthermore, data storage in the proposed architecture follows a distributed design, where each network component maintains its own local repository, collectively forming a data fabric that ensures scalability, resilience, and fault tolerance. These distributed repositories are federated into a cohesive framework that enables unified access to heterogeneous data sources. However, during the execution of an active NDT instance, a central and continuously updated data store is required to support efficient model training, inference, and control operations. This central repository, instantiated during the creation of the basic model as detailed in Section V-A.2, consolidates relevant data at runtime and renders it “active” by capturing inter-component relationships and enabling functional models on top of it.

2) BASIC MODELS

Basic models build on the harmonized data from the UDR to provide a structured representation of the physical network, its constituent NEs, and their interrelations, reflecting both topology and operational state. These models can be instantiated using graph-based representations, which emphasize structural dependencies, or simulation-based representations, which capture system behavior under various conditions.

Although NDT modeling is still emerging, useful insights can be drawn from DT practices in domains such as manufacturing, where standardized modeling approaches (e.g., ISO frameworks [75]) define structured representations of assets and their interdependencies. These frameworks commonly adopt hierarchical modeling [76], distinguishing unit-, system-, and system-of-systems levels. By analogy, in 6G networks, individual NEs correspond to unit-level entities, while the network as a whole can be treated as the system level. Given that the distinction between system and system-of-systems primarily reflects the degree of granularity, we treat the network as a cohesive system for the scope of basic model construction.

Graph-based formalisms have long been recognized as effective means of representing interconnected and dynamic systems. The comprehensive survey presented in [77] explores methodologies for modeling *complex networked systems*, emphasizing representations that preserve both structural and behavioral heterogeneity. The authors categorize these approaches into three main paradigms: basic graph-based, probabilistic graph-based, and network embedding models, each offering progressively higher expressiveness and fidelity in capturing network complexity. A complementary perspective is offered in [78], which proposes a dynamic knowledge-graph framework for DTs grounded in semantic web principles. The framework distinguishes between a "base world", synchronized with live data streams, and multiple "parallel worlds", enabling the exploration of hypothetical configurations or design alternatives without impacting the operational model. This concept demonstrates how semantic and graph-based representations can enhance adaptability and reasoning in DTs. Other studies have extended such approaches to the networking domain. For instance, [79] defines an NDT reference architecture through a software engineering lens, mapping modeling concepts from ETSI standards and established network simulators such as ns-3 and OMNET++. This work underscores the importance of metamodels that formally capture NDT entities, their attributes, and relationships. Similarly, [80] presents a vendor-neutral data model designed to simplify network management, representing logical and physical dependencies, such as shared sessions or resource linkages, through graph abstractions. These contributions collectively illustrate how graph-based representations provide a natural foundation for defining the structure and semantics of NDTs.

a: GRAPH-BASED REPRESENTATION OF NDT BASIC MODELS

The proposed methodology for constructing and utilizing basic models within the NDT views the network as an integrated system of interconnected NEs. Each NE is represented using the harmonized data structure introduced in Section V-A.1.b, consisting of its *attributes* (configuration and state) and *measurements* (dynamic performance indicators). The *basic model* is therefore defined as the set of these NE metamodels together with the logical or physical relations that interconnect them [3]. An example of a RAN-level basic model is illustrated in Fig. 5, where interconnected NEs are represented as nodes annotated with their respective attributes and measurements. This skeleton graph serves as the structural foundation upon which network instances are dynamically constructed. When an application triggers an

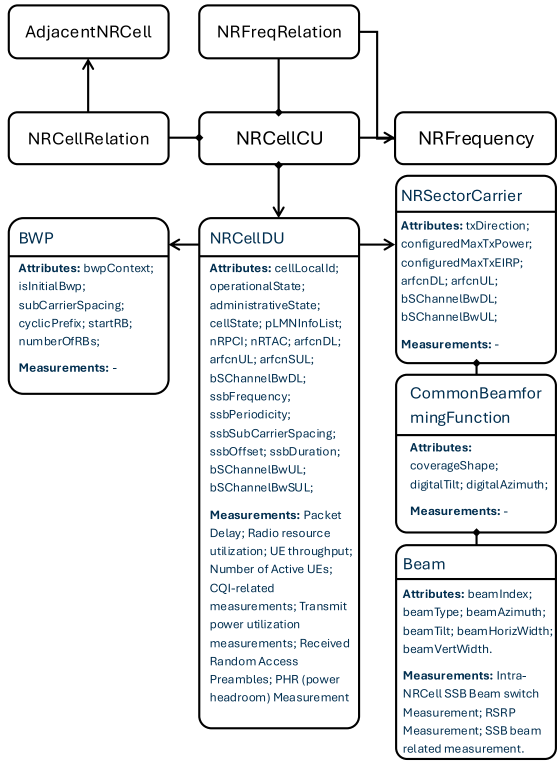

FIGURE 5. Example of a graph basic model schema for RAN NEs.

NDT instance, specifying a network domain or geographical area, the NDT manager (see Section VI-A) populates the skeleton with real-time and historical data retrieved from the UDR. Contextual data, when relevant, can also be incorporated at this stage to enrich the representation.

Basic models, therefore, complement the UDR by providing a dynamic operational layer instantiated on demand. While the UDR serves as a persistent, system-wide repository, the basic model centralizes the relevant and up-to-date subset of data necessary for efficient model training, simulation, and execution, avoiding redundant storage and enabling fast querying, reasoning, and dynamic population. If an application requires emulating network operations such as resource allocation, scheduling, or beam management, the basic model triggers the use of existing simulation tools that implement these procedures, thereby avoiding redundant re-implementation. In this case, the graph-based representation of the network is translated into simulator-compatible structures to reproduce the current network state.

As networks scale to multi-domain deployments, managing the size and complexity of these graph representations becomes an important consideration. The adoption of scalable graph analytics and efficient querying tools is therefore essential, as further discussed in Section V-A.2.b.

b: TOOLS FOR GRAPH-BASED NDTs

While most DT platforms were initially designed for Internet of Things (IoT) and industrial systems, their underlying concepts can be adapted for 6G networks. In the following,

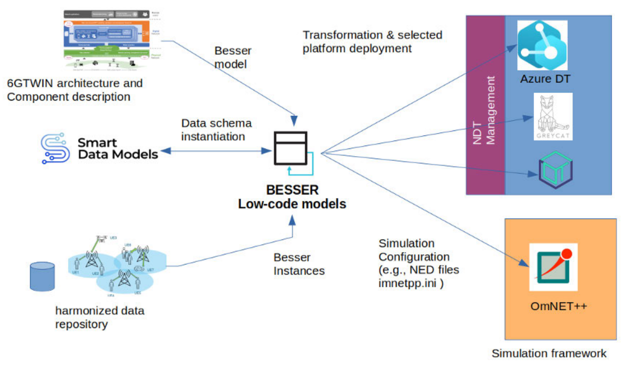

FIGURE 6. Low-code approach: targeting different platforms according to the case requirements.

we review existing tools for graph-based modeling and discuss their suitability for our NDT framework.

DT platforms support data structuring and modeling through description languages, where [81] classifies these tools according to DT capabilities and aligns such classification with the architectural requirements for basic model definition and integration. *Azure Digital Twin* provides an IoT-oriented DT environment using the JSON-based *Digital Twin Definition Language* [82], [83]. It supports graph-based representations of entities and relationships but relies on proprietary back-end services. *Eclipse BaSyx* implements the Asset Administration Shell [84], a hierarchical model representing assets and sub-models via HTTP/REST or MQTT interfaces [85]. While rich in interoperability features, it lacks native graph abstractions. *Open Twins (Eclipse Ditto)* [86] enables the creation of entity-relationship graphs combining IoT data, metadata, and external simulation or AI tools. *FIWARE* and its NGSI-LD context broker [87], [88] manage contextual information as linked data graphs, offering high interoperability but limited schema validation. *GreyCat* [89], [90] adopts a dynamic graph structure for real-time and historical data management, combining graph and time-series processing in scalable, programmable data models.

Among these, GreyCat and Azure DT offer the strongest support for graph-based modeling. GreyCat's fully graph-oriented data structure natively integrates both data and functional models, whereas Azure DT provides robust graph representation but limited openness. FIWARE and BaSyx are valuable for data harmonization and interoperability, but lack advanced graph semantics. For these reasons, GreyCat is selected as the main tool for implementing our basic models.

To operationalize this approach, we adopt an open, low-code, and platform-agnostic methodology for developing and integrating graph-based NDTs, as illustrated in Fig. 6. The proposed workflow leverages smart-data models to ensure semantic consistency across network domains (Section V-A.1) and is implemented using the *BESSER* framework [91], [92], which automates model generation, schema synchronization, and deployment to the target execution platform, GreyCat. Within this workflow, GreyCat acts as the core runtime environment, supporting the representation and execution of basic and functional models, enabling real-time analytics, and managing the continuous evolution of the NDT throughout its lifecycle.

3) FUNCTIONAL MODELS

While basic models describe the physical network, functional models operate at a higher abstraction layer and focus on application-driven objectives. Within the NDT architecture, functional models serve as the analytical and decision-making components that translate network observations from the structured basic models into actionable intelligence, using algorithmic, mathematical, or AI/ML-based methods to support both real-time and predictive control [93], [94]. Section V-A.3.a introduces the main categories and characteristics of functional models. Section V-A.3.b then presents the proposed taxonomy for 6G-oriented NDTs, enabling systematic labeling, selection, and orchestration of functional models by the NDT MANO according to their intended role and operation.

a: TYPOLOGY OF FUNCTIONAL MODEL

The construction of a functional model depends on the primary function it serves, its complexity, and the technology in which it is embedded. Accordingly, functional models can be classified into five principal categories as follows.

- *Analytical models.* Based on mathematical formulations and physical laws, these models simulate or forecast network behavior through closed-form or approximated solutions [95]. They are effective for performance analysis and capacity planning [96], such as queuing models for RAN congestion or radio propagation models for signal attenuation. However, they struggle with nonlinear or adaptive dynamics.
- *AI-based models.* These models use ML and Deep Learning (DL) techniques to capture complex and nonlinear network behaviors [97]. Examples include DNN for traffic prediction, deep reinforcement learning (DRL) for dynamic resource allocation, and Graph Neural Networks (GNN) for topology optimization. They offer high adaptability but demand extensive training data and computing resources.
- *Probabilistic models.* Designed to handle uncertainty and stochastic behavior, they enable probabilistic reasoning and decision-making under incomplete data [98], [99]. Typical approaches include Bayesian Networks, Markov Decision Process (MDP), and Monte Carlo simulations. While robust, they can be computationally demanding.
- *Deterministic models.* These models apply predefined rules or algorithms that yield identical outputs for identical inputs [100]. Examples include finite-state machines used to emulate protocol behavior. They are simple and efficient but lack flexibility in dynamic environments.

- *Hybrid models.* Combining analytical, probabilistic, deterministic, and AI-based methods [101], hybrid models exploit the strengths of each to balance interpretability and adaptability. They are well-suited to simulation frameworks but introduce integration and validation challenges.

Each model type presents distinct advantages and limitations in terms of interpretability, scalability, computational complexity, and adaptability. The choice of a functional model, or a combination thereof, depends on the objectives and operational constraints of the NDT application.

b: TAXONOMY OF NDT FUNCTIONAL MODELS

The type and nature of a functional model determine how it is built, deployed, and orchestrated within the overall NDT architecture. To organize this diversity, we propose a multi-dimensional taxonomy that classifies functional models according to their role, scope, and deployment context. Based on the ITU-T recommendations [12], which distinguishes models by *network type* (single- or multi-domain), *function* (e.g., traffic analysis, security management, fault diagnosis), and *generality* (general- or specific-purpose), our taxonomy extends these dimensions to include *functionality*, *operational mode*, *deployment domain*, *computing element*, and *use-case domain*. This expanded classification provides a unified basis for describing and comparing functional models across heterogeneous systems, enabling systematic labeling, facilitating their orchestration by the NDT MANO, and aligning model selection with application objectives and performance requirements.

- Classification by functionality: (a) *Planning and design*: optimize topology, coverage, and capacity pre-deployment. (b) *Management and control*: ensure Service Level Agreement (SLA) compliance and dynamic adaptation through policy-based control. (c) *Diagnosis and security*: detect anomalies, faults, and threats, enabling predictive and self-healing actions. (d) *Monitoring*: observe system health and performance, providing input to higher-level models.
- Classification by use-case domain: (a) *Mobility*: Vehicle-to-Everything (V2X), traffic optimization, teleoperation. (b) *Industrial IoT*: predictive maintenance and real-time control. (c) *Smart cities*: monitoring of utilities and infrastructure. (d) *e-Health*: reliable low-latency communications for medical use. (e) *XR/immersive reality*: synchronization between physical and virtual spaces.
- Classification by operational mode: (a) *Reactive*: respond to real-time network events. (b) *Proactive*: predict and prevent issues using AI/ML. (c) *Hybrid*: combine both reactive and predictive behaviors.
- Classification by deployment domain: (a) *RAN*: manage radio resources, power, handovers, and QoS. (b) *TN*: support path computation, traffic engineering, and SDN-based slice control. (c) *CN*: automate VNF/MANO operations (e.g., AMF, SMF).
- Classification by computing element: (a) *Edge*: low-latency inference for time-critical tasks. (b) *Cloud*: large-scale analytics and prediction. (c) *Federated*: distributed learning and cross-domain coordination.

In this proposed classification, functional models are first organized according to their *functionality* and further distinguished by their *use case domain*. Additional characterization dimensions such as the model type described in Section V-A.3.a, operational mode, deployment domain, and computing element, provide a consistent framework for describing and categorizing models across diverse environments. This structure enables the NDT Manager to effectively identify, select, and coordinate functional models that best align with the objectives and operational requirements of each NDT instance. Once deployed, these models are continuously refined through the live basic models. This continuous feedback loop preserves alignment between the digital and physical domains, supporting adaptive learning, predictive decision making, and ongoing network optimization.

4) UNIFIED DATA MODEL REPOSITORY

The UDM serves as a central repository for storing metadata of functional models that have been trained, calibrated, or parameterized using an NDT instance. Each functional model in the repository can have multiple versions, corresponding to diverse operational conditions such as indoor or outdoor environments, urban or rural areas, specific geographic zones, or distinct QoS requirements. These versioned models are systematically labeled following the functional model taxonomy introduced in Section V-A.3.b, and are linked to the corresponding basic models used during their training when relevant. Once a model has been validated and tested within the NDT environment, it can either be deployed directly to the physical network or stored for future evaluation and deployment. The NDT MANO (cf. Section VI-A) is responsible for maintaining this repository. It oversees the versioning, labeling, and application procedures of functional models, ensuring that the correct model variant is selected for a given scenario and that the NDT remains synchronized with the deployed network. This unified repository enables systematic model reuse, facilitates rapid deployment, and supports consistent performance across heterogeneous network environments.

B. SIMULATION ENVIRONMENT

Simulation is a core feature of NDTs, providing the ability to study *what-if* scenarios, i.e., to analyze the consequences of network configuration changes before applying them to the physical infrastructure. This capability relies on the availability of an accurate and reliable virtual representation of the real network, capturing both its structural and behavioral characteristics. Simulators are complementary to the graph-based basic models introduced in Section V-A.2. While graph-based models define the static topology, components, and relationships of the network, simulators endow these representations with dynamics, enabling the *generation of realistic network behaviors* that closely mirror those of the physical system. The simulation environment thus serves as an experimental ground for *testing, evaluating, comparing, and training functional models* under controlled and reproducible conditions. In this context, the complete *AI training* pipeline of functional models (further detailed in Section V-C) can integrate simulators as part of the real-time or online training process. Given the scarcity and sensitivity of real operational data, simulators also serve as a critical *source of synthetic data*, supporting the expansion of training datasets and the exploration of edge-case scenarios that may be rare or infeasible to reproduce in real networks.

Many simulation tools exist in the field of 6G network simulation [102]. As 6G networks are expected to exceed the complexity and heterogeneity of current systems, no single existing simulator can encompass all the features, protocols, and technologies required to address the challenges of future networks. To overcome this limitation, our approach adopts a *federated discrete-event simulation framework*, designed to interconnect and orchestrate multiple domain-specific simulators. This federated approach allows for modularity and extensibility, enabling the coupling of simulators that model different network segments (e.g., RAN, TN, CN), radio environments, or service types, while ensuring synchronization and consistency across the simulation instances.

The interaction and coordination of the simulation framework are managed by the NDT management layer (detailed in Section VI). This layer retrieves the required models, configurations, and parameters from the UDM and the AI training module, instantiates and supervises the execution of the simulation instances, and collects performance and behavioral metrics from the simulation outcomes. The resulting insights are then reintegrated into the AI training process for continuous model refinement and optimization. Finally, the optimized parameters and validated functional models can be leveraged by the NDT management layer to update or reconfigure the physical network, thus closing the loop between simulation, prediction, and real-world operation.

1) SIMULATION ECOSYSTEM

The simulation ecosystem includes a diverse range of tools and frameworks that support the development, testing, and optimization of complex systems across multiple domains. Each simulator type provides distinct capabilities, enabling accurate modeling, real-time experimentation, and validation of network and physical processes. Collectively, they form the foundation of the NDT's ability to emulate realistic behaviors and evaluate system performance prior to deployment. The following sections outline the main categories of simulators relevant to NDTs.

a: REAL-TIME SIMULATION

Real-time simulation executes models at the pace of actual time, providing immediate feedback for testing, control validation, and operator training. It is widely applied in Hardware- and Software-in-the-Loop environments to validate interactions between real and simulated components before deployment. Some frameworks also operate faster than real time, enabling accelerated scenario exploration and long-term performance evaluation. Simulator coupling is an effective way to achieve real-time capabilities. For instance, Veins [103] integrates OMNeT++ and SUMO for bidirectional interaction between mobility and communication layers, enabling realistic assessments of vehicular networks. Similarly, [104] extends this approach by federating additional simulators for heterogeneous real-time testing of connected vehicle scenarios.

b: SIMULATION WITH HIGH LEVEL ARCHITECTURE (HLA)

The High Level Architecture (HLA) standard provides a framework for synchronizing distributed simulators, enabling heterogeneous systems to exchange data and maintain temporal consistency. Such co-simulation environments are essential for integrating network, mobility, and physical-layer models within a unified experimental setup. Representative examples include the VSimRTI framework [105], which combines SUMO, JiST/SWANS, and eWorld for realistic V2X simulation across mobility, network, and environmental domains. More recently, [106] extends this concept through flexible HLA-inspired coupling of Veins, SUMO, and domain-specific simulators for vehicles, UAVs, and Low Earth Orbit (LEO) satellite networks, improving scalability while retaining modularity.

c: DOMAIN-SPECIFIC SIMULATION

Domain-specific simulators provide realistic, modular environments for testing and optimizing complex systems such as Unmanned Aerial Vehicles (UAVs), robots, and vehicular networks. They integrate dynamics, communication, and control models to evaluate connectivity, coordination, and performance under realistic conditions. Examples include UAV frameworks combining aerial mobility with communication models [107] and vehicular co-simulation platforms coupling traffic and network layers for V2X and 5G-enabled scenarios [103]. Collectively, these simulators enable scalable, cross-domain experimentation for developing and validating networked intelligent systems.

2) SIMULATION TOOLS

Simulation tools are essential for network planning, optimization, and performance evaluation. Industrial tools, typically licensed, address deployment challenges such as coverage analysis, capacity planning, and KPI optimization, while research-oriented simulators provide flexible, open environments for detailed modeling and experimentation. Commercial suites like ASSET [108] and Atoll [109] support advanced 5G New Radio (NR) planning with propagation modeling, beamforming, and KPI-driven optimization, whereas iBwave [110] focuses on small-cell and indoor network design. On the research side, simulators such as NS-3 [111], OMNeT++ [112], and OpenAirInterface [113] enable detailed 3GPP-compliant studies of radio, core, and end-to-end network behavior. The Vienna 5G Simulator [114] further supports large-scale system-level evaluation with realistic channel models, offering a bridge between analytical design and experimental validation for NDT-driven 6G research.

3) INTERACTION BETWEEN SIMULATION AND FUNCTIONAL MODELS

The integration between the simulation framework and the NDT components is established through a structured data translation process. Specifically, the data contained in the basic models are converted into simulator-readable formats to build the virtual environment in which the simulation is executed. This translation ensures that the virtual network accurately reflects the structure and parameters of the real system, enabling consistent testing and validation of models.

It is, however, essential to distinguish between *simulation models* and *functional models*, as they are closely related but conceptually distinct. Both play a key role in enabling the operation and intelligence of the NDT, yet they differ in scope and level of abstraction. A simulation model refers to a machine-readable description of a specific scenario that can be interpreted and executed by a simulator. The simulator itself is the software responsible for loading and running this model, generating results that describe the system's behavior under the given conditions. The term *simulation* therefore denotes the actual execution of the simulation model within the simulator, producing synthetic data or performance indicators that reflect the system's response. In contrast, a *functional model* is designed to analyze, predict, or optimize network behavior. It typically combines an analytic, stochastic, or AI-based component with an optimization or training algorithm, while also embedding a simulation model that defines the operational scenario. In other words, the simulation model provides the environment and dynamic context in which the functional model operates. The functional model extends beyond mere simulation by incorporating intelligence and decision-making capabilities that enable continuous adaptation, learning, and performance optimization.

The interaction between both functional and simulation models forms the foundation of closed-loop experimentation and model refinement in the NDT. The process of training functional models through simulation will be detailed in Section V-C, while the execution and federation of simulations for model evaluation will be further discussed in Section VI-B.

C. AI TRAINING/OPTIMIZER

The AI training and optimization component within the NDT architecture is responsible for preparing functional models for deployment, refined for the application and scenario requirements. This module enables models to learn from data, predict network behavior, adapt to changing conditions, and continuously improve their performance through iterative training and validation cycles [115]. While all functional models are subject to optimization, their training requirements differ depending on their type. Analytical and deterministic models are expressed through explicit mathematical formulations or algorithmic rules and therefore do not require a dedicated training pipeline [116]. Once parameterized with suitable static values, these models are directly stored in the UDM and can be invoked by the NDT management layer when needed. In contrast, *AI-based* and *probabilistic models* require a structured training process. This involves computing model parameters, such as neural network weights or probability distributions, based on available data [99]. The following subsections outline the training pipelines applied to each class of models.

1) DEEP LEARNING AND PROBABILISTIC MODEL TRAINING

For DL and probabilistic models, the training process begins with the preparation of datasets that define the relationship between input and output variables [117]. These datasets may originate from the physical network or be synthetically generated through the simulation framework. In both cases, data preprocessing is critical in ensuring convergence and accuracy. This includes data cleaning to handle missing or outlier values, and feature selection or dimensionality reduction techniques such as ANOVA, Lasso, or principal component analysis to retain only the most informative features. Once preprocessed, the data are fed into the learning stage, where model parameters are iteratively optimized according to a defined loss function (e.g., mean square error for regression and cross-entropy for classification [118]). Gradient-based optimization adjusts the model weights until a target accuracy or convergence criterion is achieved. The trained model, along with its metadata (e.g., training conditions, performance metrics, version number), is then stored in the UDM to ensure traceability and future reuse. Functional models can be retrained when new data becomes available or as part of continuous learning procedures. In the case of probabilistic models, the pipeline follows a similar structure. After preprocessing, the algorithm learns the underlying probability distributions from the data using techniques such as Bayesian Networks, MDPs, or Monte Carlo simulations. The resulting probability functions and their associated metadata are also stored in the UDM for future inference or integration into hybrid models.

2) REINFORCEMENT LEARNING TRAINING

Reinforcement Learning (RL) follows a distinct paradigm in which the model, or agent, learns optimal decision-making strategies by interacting with an environment [119]. Unlike DL or probabilistic models, RL does not rely on

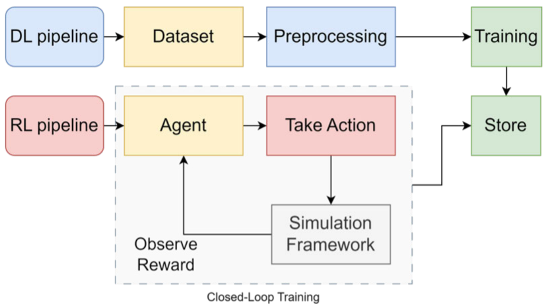

FIGURE 7. **Functional models training pipeline.**

a pre-existing dataset; instead, it generates training data dynamically through *exploration* and *exploitation*. The learning process is typically modeled as an MDP defined by a state space (environmental conditions), an action space (possible interventions), and a reward function (feedback signal after each action) [120].

For small state-action spaces, tabular Q-learning methods can store explicit values for each possible pair. However, as complexity increases, RL benefits from neural network approximations of the value function, resulting in DRL [121]. During training, the agent iteratively explores its environment, balancing exploration of new strategies with exploitation of known ones using techniques such as ϵ-greedy policies or Boltzmann exploration. The training can occur in real-time within an NDT instance or be conducted offline using simulated environments built from the simulation framework [122]. Once training converges, the learned policy or Q-function weights are stored in the UDM repository. These trained models can subsequently be deployed to monitor live network states and autonomously execute control actions based on learned strategies.

3) UNIFIED TRAINING AND OPTIMIZATION FRAMEWORK

Our proposed architecture provides a unified framework for the training and optimization of all functional model types identified in Section V-A.3.a. While AI-based models rely on supervised, unsupervised, or reinforcement learning paradigms, other models undergo calibration and parameter fitting to align theoretical assumptions with empirical observations. Hybrid models combine these approaches, requiring joint optimization procedures that bridge analytical and data-driven methods.

Fig. 7 illustrates the pipelines for DL and RL training. The outcome of each training or optimization process, whether it is a set of trained AI weights, a probabilistic distribution, or optimized analytical parameters, is systematically stored in the UDM. Each version is labeled with metadata corresponding to the model taxonomy defined in Section V-A.3, including its functionality, type, operational mode, and network domain. This structured metadata enables efficient orchestration, reuse, and adaptation of models within the NDT management layer.

In summary, the AI training and optimization pipeline ensures that every functional model within the NDT is continuously improved and ready for deployment. The interaction between the training component, simulation framework, and management layer creates a closed learning loop in which models evolve alongside the network itself. The detailed mechanisms of this interaction are further elaborated in Section VI-B.

VI. MANAGEMENT DOMAIN

The efficient operation of NDTs demands a unified management framework that coordinates activities across the physical network, simulation environments, data repositories, and network models. To this end, the proposed architecture introduces the *NDT MANO* framework (see Fig. 2), which serves as the central control entity governing the lifecycle and interactions of all NDT components. It synchronizes the physical and digital domains, oversees the instantiation and execution of NDT instances, and orchestrates data collection, simulation workflows, and AI-based model training. Through cross-domain orchestration, it enables adaptive resource management and continuous network optimization under dynamic conditions.

The NDT MANO framework comprises several functional subcomponents: NDT management, simulation management, AI workflow management, data management, federated management, and physical network MANO, each ensuring consistency, automation, and scalability within the overall NDT ecosystem. The following subsections describe their main functions and interrelations within the architecture.

A. NDT MANAGEMENT

According to ETSI ZSM [15], an NDT should be use case specific, where specific combining basic and functional models into a running NDT instance (cf. Fig. 2). An instance relies on key components, including an NDT Orchestrator that coordinates its internal processes, a Model Manager that supervises model performance and triggers corrective actions, a Resource Manager overseeing computing resources, and an NDT Monitor that tracks system behavior and alerts anomalies. A Translation Layer further enables interaction by interpreting user inputs and producing human-readable outputs.

Within the Management domain, the NDT Management block provides high-level coordination and lifecycle control (LCM) of all NDT instances and subcomponents [16], [123]. It acts as an intelligence layer, translating user intent and network context into operational configurations. Its main functions include (i) instantiating, configuring, and orchestrating NDT components; (ii) supervising resource utilization and performance; and (iii) managing the full lifecycle of NDT instances as network services evolve. Through these functions, it ensures synchronized operation among data, models, simulations, and AI mechanisms across digital and physical domains.

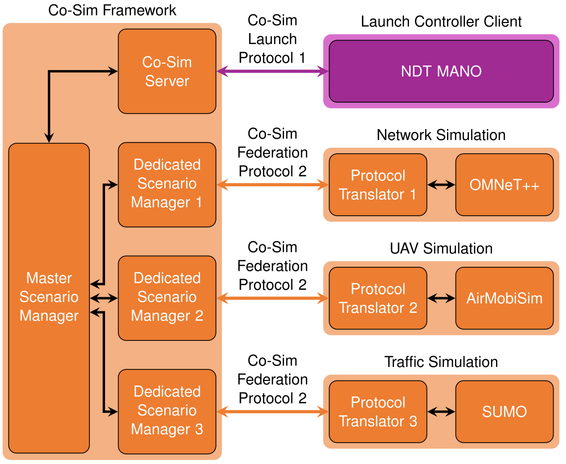

FIGURE 8. Co-simulation framework architecture.

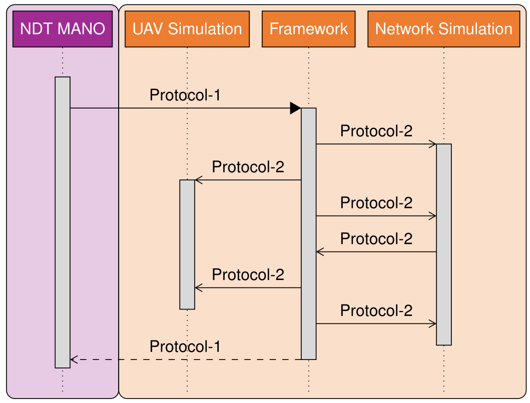

FIGURE 9. Co-simulation framework interaction.

The NDT management process follows several stages [29], [124], [125], [126]; thus, it is seen as the mastermind within the NDT management domain. It defines the use case and assets to be modeled, collects and harmonizes data, creates virtual representations, and integrates them via federation and interoperability standards. Once deployed across cloud, edge, or on-premises environments, the NDT enables simulation, analysis, and optimization, maintaining accuracy through real-time synchronization. Continuous monitoring and lifecycle management ensure its long-term fidelity and adaptability, while visualization tools facilitate human-in-the-loop decision-making. Section VII provides a detailed view of these lifecycle processes.

B. SIMULATION MANAGEMENT

Within the functional architecture, simulations primarily support *what-if* analyses—examining how network behavior changes under different parameters, without the need of having to deploy them in real-life. This process involves creating simulation models, parameterizing them, executing runs, and delivering results to other components, such as the AI training block. The *simulation framework* (Section V-B) coordinates these processes, synchronizing multiple simulators and facilitating data exchange during runtime.

Creating a simulation model relies on both basic and functional models, often using subsets of the full network for feasibility, in case the full network cannot be simulated. Information about basic and functional models might be needed when configuring simulators. For example, the network properties of a basic model are used while configuring an OMNeT++ simulation while UEs mobility traces might be used for configuring SUMO. In AI workflows (cf. Section VI-C), simulations are frequently reused and adapted across training loops. Once executed, simulation results are collected and passed to relevant entities, such as the AI module, dashboard, or the NDT MANO, for evaluation or real-world actuation.

A co-simulation framework is critical for analyzing complex, multi-domain network behavior within the NDT environment. It enables synthetic data generation, scenario testing, and what-if analyses to assess network performance under dynamic conditions. As illustrated in Fig. 8, it adopts a federated architecture [106] where multiple specialized simulators, or *federates*, represent distinct domains (e.g., road traffic, UAV traffic, or communication). The framework ensures synchronization and data exchange among federates through a modular, technology-agnostic design that supports simulator substitution and extensibility. Such composability has proven essential for accurately capturing effects across multiple domains in closed-loop optimization [127].

Communication between the NDT and the co-simulation framework occurs via a standardized google Remote Procedure Call (gRPC)-based interface, referred to as *Protocol-1*, which manages simulation control, configuration, and result exchange. Via this protocol, it interacts with a dedicated component in the framework, the *co-sim server* that sets up simulation runs. Simulation coordination is managed by a *master scenario manager* interacting with simulator-specific *scenario managers* through a shared API using *Protocol-2*, which implies simulator-specific *protocol translators*. This second protocol governs the runtime synchronization of federates through services such as *Start* and *Finish*, which start and terminate the execution of a federate; *ExecuteOneTimeStep*, *QueryRequest*, *SetAttribute*, and *GetAttribute*, which are used to interact with the federates; and entity management services via *InsertHost* and *DeleteHost* to dynamically control the number of simulation entities.

Upon completion, simulation outputs are transmitted back to the NDT through Protocol-1 for further use in AI-based optimization or visualization (Fig. 9). A persistent session between the NDT and the framework allows iterative experimentation, scenario refinement, and adaptive model calibration without reinitialization.

C. AI WORKFLOW MANAGEMENT

The AI workflow management component is vital to enabling data-driven and adaptive NDT operations. It manages the full lifecycle of AI/ML processes, including data ingestion, model training, validation, deployment, and inference, coordinating interactions among the physical network, NDT elements, simulation framework, and training mechanisms. Depending on the task, different learning paradigms are supported: DL for tasks such as traffic classification or radio prediction, probabilistic models like Gaussian Process Regression for spatial inference, and RL/DRL for closed-loop control tasks such as resource allocation or handover optimization.

The component operates over a unified data infrastructure, where telemetry is harmonized and stored in the UDR using standardized formats (e.g., Smart Data Models compliant with 3GPP/O-RAN). For RL-based functions, the AI block interacts directly with the simulation framework (e.g., OMNeT++, ns-3) to generate synthetic environments that enable safe policy learning before deployment in live networks.

Workflow automation and scalability are ensured through modern MLOps practices, such as orchestration tools like Perfect and Kubeflow in Kubernetes environments, that provide reproducibility, version control, and efficient resource usage. Model metadata, such as performance metrics, and versioning details are maintained in the UDM, enabling continuous monitoring and retraining triggered by performance degradation or data drift.

Aligned with emerging AI-native management paradigms, this framework supports distributed and federated learning by decoupling data ingestion from training and inference. Its modular design ensures low-latency, high-throughput processing of massive 6G telemetry streams while preserving responsiveness for mission-critical applications such as industrial automation and teleoperated driving [128].

D. 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 NDT 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.

Our architecture 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, as explained in Section IV-B. Based on the IETF RFC9232 [129], our approach includes the *generation*, *collection*, *processing*, and *consumption* data modules, mapped in the architecture via the telemetry data layer. These modules support data MANO services, such as security, scalability, and federated distributed parallelization. Additionally, the HDL 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, which serve as the foundation for defining and operating the entire architecture.

E. FEDERATED MANAGEMENT

Modern communication networks increasingly span multiple autonomous administrative domains, each enforcing independent policies and operating under distinct technological and governance constraints [130]. This decentralization enhances flexibility but complicates end-to-end coordination, especially when automation and cross-domain optimization are required.

From a physical-network perspective, federation enables multi-domain service delivery by decomposing services across administrative boundaries. Recent 3GPP releases (e.g., Release 16 [70]) define hierarchical management entities to support such cross-domain orchestration, consistent with frameworks like ITU-T Y.3061 and ETSI ZSM [47], [131]. However, integrating AI-based network functions introduces dynamics (e.g., adaptivity, continuous learning, and dependency on real-time data) that challenge the determinism of traditional hierarchical models and require more flexible coordination mechanisms.

Federation at the digital level, enabled by interconnected NDTs, offers a complementary approach by supporting privacy-preserving cross-domain collaboration. NDTs may operate at node-, domain-, or service-level granularity [15], and can be hierarchically composed so that lower-level twins feed insights to higher-level abstractions. This allows administrative domains to retain control over their data and models while exchanging standardized, anonymized information for joint optimization and pre-deployment validation of controllers or AI policies.

In large-scale deployments, multiple distributed NDT instances must remain synchronized. The federated management component ensures state consistency, regulates inter-twin communication, and provides policy-driven orchestration to avoid conflicting decisions across the federation. Moreover, federated setups may integrate external simulators or digital platforms [132]; in these cases, the simulation framework described in Section VI-B maintains semantic and temporal alignment across heterogeneous components.

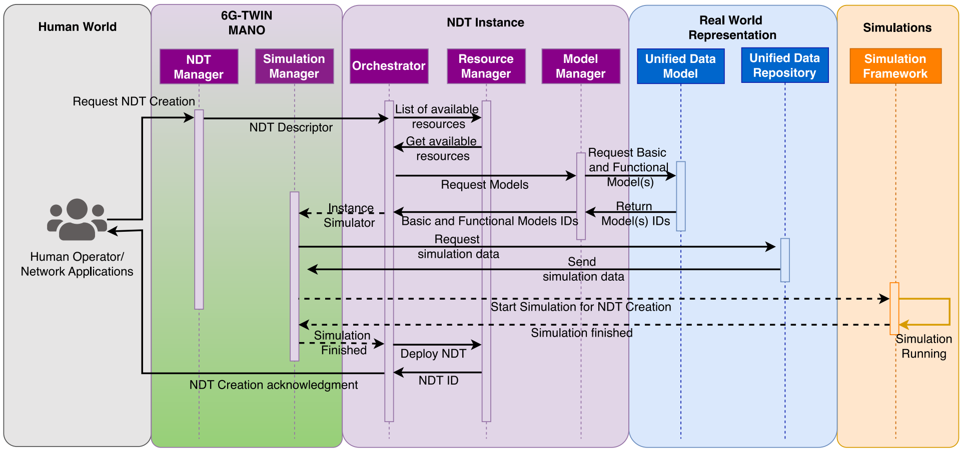

FIGURE 10. NDT creation sequence diagram.

F. PHYSICAL NETWORK MANAGEMENT

The proposed architecture introduces a *physical network management* block that serves as the orchestration and automation interface between the physical infrastructure and its NDT. This component integrates traditional LCM with AI/ML-driven decision-making and closed-loop control, enhancing network programmability and enabling efficient telemetry pipelines to preserve alignment between physical and digital layers. The physical domain also supports self-monitoring, analysis, and adaptation mechanisms, contributing to zero-touch operation in future 6G systems.

Two complementary closed-loop processes govern physical network behavior. The internal loop manages the creation and configuration of NDT instances through the NDT management system, coordinating interactions with simulation frameworks and AI training modules. The external loop uses NDT as a service, leveraging continuous telemetry to aid the physical network management to diagnose performance deviations, optimize configurations, and reapply decisions to the physical network in real time. Together, these loops shift network control from static rule-based methods toward an AI-native, self-optimizing paradigm.

The MANO/ZSM layer further enables automated device onboarding, intent translation, and secure exposure of network capabilities. Through these mechanisms, the physical network becomes a programmable and adaptive substrate that supports intelligent, closed-loop management across the 6G ecosystem.

VII. LIFECYCLE MANAGEMENT OF NDT INSTANCES

The lifecycle of an NDT involves coordinated interactions among the various building blocks and layers of the architecture depicted in Fig. 2. In this section, these interactions are captured through Unified Modeling Language (UML) sequence diagrams that describe how the system behaves over time. While the following workflows remain generic, they can be tailored for specific deployment scenarios.

A. CREATION

As shown in Fig. 10, NDT creation is triggered on demand due to the computational cost of running high-fidelity simulations and performing "what-if" analyses. Upon receiving a request, the NDT Management component interprets user and system requirements and generates an NDT descriptor [131]. The descriptor defines the initial scenario deployment, required basic and functional models, data sources, performance bounds, simulation settings, software dependencies, and expected outputs.

The NDT Manager initializes a generic instance template, which is refined in co-creation with the end user turning high-level requests into executable specifications. Then, the orchestrator analyzes the descriptor and coordinates with the resource manager, validating infrastructure availability. If sufficient resources are available, the model manager retrieves or generates missing models via the UDM. Additionally, the system may need to pull relevant input data, either historical or real-time, from the UDR to feed into the basic models. The simulation manager instantiates any simulation tools in case they are required for the NDT operation. After all components are prepared and configured, the orchestrator deploys the NDT instance and acknowledges readiness to the requester.

B. UPDATE

NDT instances evolve as models are refreshed with new physical network data. Basic models follow the data-collection update cycle, while functional models are updated either reactively, upon quality-metric degradation, or through periodic scheduling. We consider the reactive case, where the NDT continues operating under the previous configuration until an updated version is installed. Depending on the type of functional model(s) the NDT is based on, several scenarios may occur when updating it.

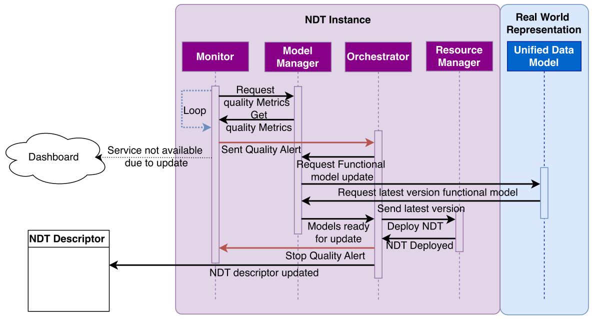

FIGURE 11. Update analytical functional models.

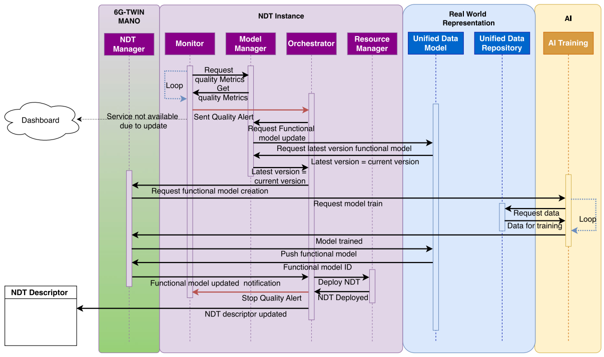

FIGURE 12. Update DL-based functional models.

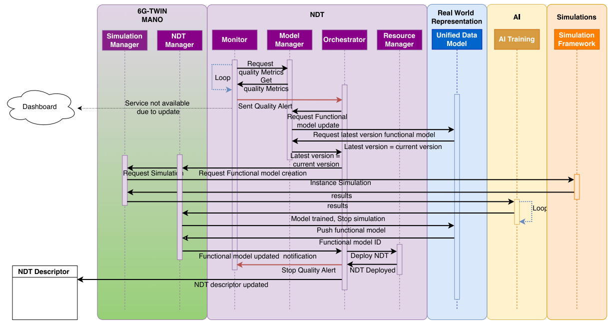

FIGURE 13. Update RL-based functional models.

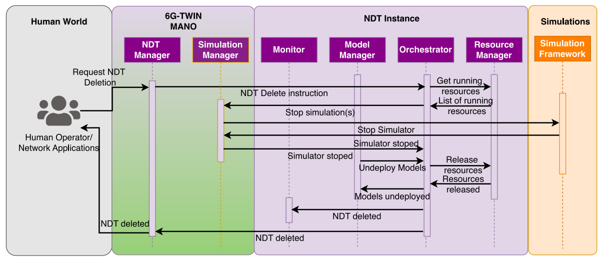

FIGURE 14. NDT deletion.

1) ANALYTICAL FUNCTIONAL MODELS

As illustrated in Fig. 11, the monitoring system triggers a quality alert that prompts the orchestrator to request a model update. The model manager checks the UDM; if no newer version exists, a new one is generated. The orchestrator then installs the updated model, amends the NDT descriptor, and clears the alert. Notice that such updates often respond to changes in physical-layer characteristics or system parameters.

2) AI-BASED FUNCTIONAL MODELS

As mentioned in Section V-C, if the functional model is based on AI/ML techniques, different pipelines are followed depending on the type of technique.

For DL-based models (Fig. 12), the update process also begins with a quality alert. If no updated version exists, the orchestrator triggers a new AI training pipeline via the NDT manager. The AI workflow block trains the model using data from the UDR, updates the UDM repository, and notifies the orchestrator to deploy the new version.

For RL-based models (Fig. 13), training additionally requires simulation–AI interaction. The NDT manager coordinates with the simulation framework to support time-step interactions until the RL policy converges. The new model is then stored in the UDM and deployed as above.

C. DELETION

Fig. 14 shows the deletion workflow. A deletion request may originate from an external entity or result from the expiration of the NDT descriptor. The NDT Manager issues the delete command, and the orchestrator retrieves resource usage information, terminates basic and functional models, releases all resources, and confirms deletion to the requester.

VIII. IMPLEMENTATION AND RESULTS

Building on the theoretical foundations outlined in earlier sections, where the architectural components enabling and creating NDT as a service were introduced, this section shifts focus to the practical realization of some of these concepts. It presents a series of implementations and examples that demonstrate the feasibility and effectiveness of the proposed NDT architecture.

A. TELEMETRY FRAMEWORK FOR LOCATION PREDICTION

The TDL was conceived as a foundational component of the *data collection framework* described in Section IV-B.2, enabling AI-native network management and NDT synchronization across heterogeneous domains. It provides a unified architecture for the collection, harmonization, and exposure of multi-domain data streams, supporting both O-RAN-compliant and non-compliant systems. As discussed previously, the *data collection framework* is organized in three principal layers: (i) the TDL, which ingests and preprocesses real-time network measurements from RUs, DUs, and Core components; (ii) the HDL, responsible for normalizing heterogeneous inputs into standardized data models (e.g., Yet Another Next Generation (YANG), JSON, or Protobuf formats); and (iii) the DCB, providing scalable publish/subscribe interfaces through Kafka, NATS, or Zenoh for northbound data exposure toward AI/ML functions and NDT engines.

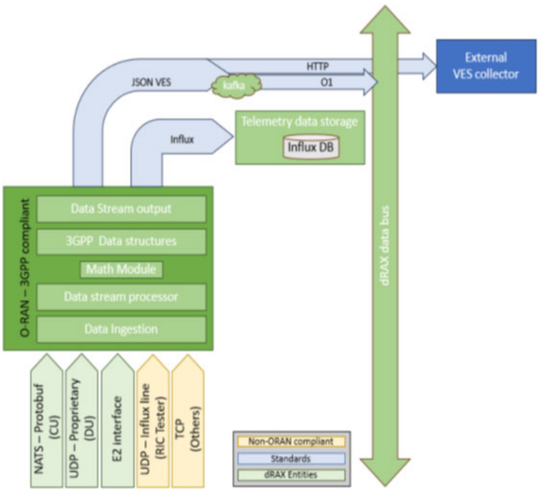

FIGURE 15. Telemetry Gateway implementation.

The functional realization of the TDL was implemented through the Telemetry Gateway (TGW), a modular software entity integrated within the Accelleran dRAX Near-Real-Time RIC environment. The TGW acts as an interoperability layer across E2, O1, and A1 interfaces, bridging O-RAN and Non-O-RAN compliant data, as well as legacy components.

Internally, the TGW comprises three main modules: a) the Input Interface Manager, supporting NETCONF, RESTCONF, User Datagram Protocol (UDP), and InfluxDB line protocols for ingesting RAN metrics and device telemetry. This input interface is mapped within the data ingestion and data stream process mechanisms described in the *data collection framework*. b) Translator Core, including Transformers and Mappers to convert raw messages into normalized telemetry structures compliant with O-RAN and 3GPP specifications; and c) the Output Publisher, which timestamps and streams processed metrics to Kafka and HTTP/Virtual Event Streaming (VES) collectors for consumption by xApps, rApps, and external AI pipelines, based on the 3GPP data structures and the data stream output. This output is then fed into the dRAX DCB for data storage or external VES collectors as described in Fig. 15.

For the implementation, telemetry streams were collected from a 5G O-RAN testbed provided by VIAVI's AI RAN Scenario Generator (RSG) tool. This tool emulates a realistic campus outdoor scenario with six cells and 20 users moving across the campus. The TGW continuously exported performance indicators such as Signal-To-Interference-Plus-Noise Ratio (SINR), Physical Resource Block utilization, and power consumption toward the dRAX DCB. These data were consumed by an energy-saving and traffic steering xApp to control the network's performance. Additionally, a location modeling rApp was created to model the radio environment, based on the telemetry ingested. Inside this rApp, a set of algorithms calculates the location of the UEs in relation to the cells and the error of its predictions. The entire pipeline is visualized in Grafana dashboards, showing

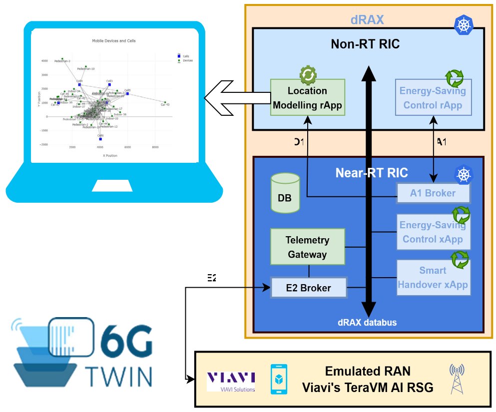

FIGURE 16. Internal demo architecture.

FIGURE 17. TGW dashboard.

near-instantaneous reflection of physical-network events in the NDT environment as in Fig. 16 and Fig. 17. The TDL implementation was showcased live at EuCNC 2025.

The latency between physical measurement and AI model update averaged below 250 ms, while the harmonization overhead remained below 3 % of total processing time. The system sustained a throughput of over 15000 telemetry messages per second without data loss across Kafka topics, confirming the scalability of the implementation. Finally, the calculated error is around 50 m, which highlights the necessity to involve further sensing information to increase accuracy, which is foreseen for future implementation of this demonstrator.

These results validate the TGW as a robust and scalable mechanism for unified telemetry in AI-driven network management. Its modular architecture ensures adaptability to diverse network technologies and provides an operational foundation for zero-touch automation and federated AI training. From a research perspective, the experiment demonstrates that telemetry-assisted NDT synchronization can be achieved with sub-second latency, enabling dynamic retraining of AI models and real-time policy enforcement in 6G-oriented RIC environments.

B. NDT INSTANTIATION OF BASIC AND FUNCTIONAL MODELS

After data collection and harmonization, the resulting datasets are stored in the UDR for future retrieval and analysis within the NDT instance. As an initial demonstration, we focus here on *radio coverage prediction*, which serves as

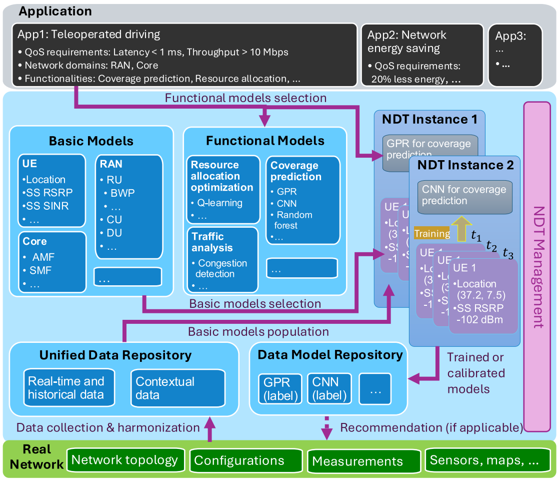

FIGURE 18. Example of NDT instantiation for radio coverage prediction.

a representative use case for illustrating the instantiation and interaction between basic and functional models. A detailed analysis of this use case is reported in [133]; while here, we emphasize the architectural aspects of model instantiation and management within the NDT framework. The instantiation process is illustrated in Fig. 18, where purple arrows indicate the NDT MANO workflow. While this process can ultimately be automated, the current implementation relies on expert input to guide model selection and configuration. During instantiation, the NDT MANO populates the basic models with data retrieved from the UDR and associates them with functional models relevant to the specific use case. These functional models are trained or calibrated using the basic model to produce parameterized versions which are subsequently stored in a dedicated repository for traceability, benchmarking, and offline validation.

1) BASIC MODEL AND FUNCTIONAL MODELS

The basic model represents the infrastructure of the network. For this experiment, we selected a real-world environment in an indoor laboratory setting and a 5G gNB using the OAIBOX platform. The measurements were collected across a single floor, where a UE was connected to the gNB, and logs containing Reference Signal Received Power (RSRP), as an indicator of signal strength, were collected at multiple static positions for approximately two minutes per location. The logs were processed and ingested into *GreyCat*, following the internal schema defined for basic models (see Section V-A.2). Each measurement was timestamped and spatially linked, creating a dynamic data structure capable of supporting time-aware queries and predictions. Fig. 19 shows an example of the *GreyCat* visualization for the UE measurements over the laboratory floor plan.

To demonstrate the integration of functional models in the NDT, we implemented two complementary approaches for radio coverage estimation: a probabilistic method, a kernel-based method using *Gaussian Process Regression (GPR)* and a data-driven *Convolutional Neural Network (CNN)*. These

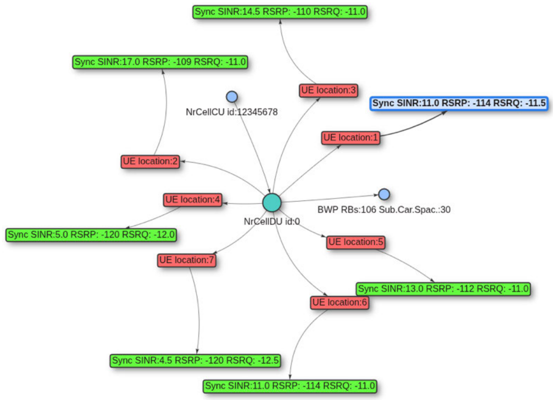

FIGURE 19. Visualization of the basic model within GreyCat.

models learn the mapping between spatial coordinates and RSRP from partial measurements, offering different trade-offs between interpretability, generalization, and sensitivity to data sparsity. The *GPR model* used location coordinates (x, y) as inputs and the averaged RSRP as the target variable, employing a Radial Basis Function kernel. Hyperparameters were optimized through likelihood maximization, and predictions were obtained through grid-based interpolation. The *CNN model* took a 2D grid of RSRP values, with missing data masked, as input. It consisted of two convolutional layers trained directly on known values using mean-squared error loss, allowing the inference of the full coverage map in a single pass.

2) EVALUATION AND COMPARATIVE ANALYSIS

The evaluation aimed to assess the predictive accuracy of both models using a shared dataset, enabling the NDT to identify the most suitable functional model for future deployment under similar conditions. The dataset was partitioned into training and validation subsets to ensure fair comparison. Each model, GPR and CNN, was trained on the same training samples, and predictions were compared against the withheld validation points.

Prediction errors were computed per location, distinguishing between Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions. The results, summarized in Fig. 20, show that the CNN achieved higher robustness in NLOS scenarios, while the GPR exhibited better accuracy in LOS. These findings underscore the importance of the NDT as an evaluation platform for comparing functional models under identical conditions and selecting the most appropriate one for deployment.

C. NDT UPDATE OF FUNCTIONAL MODEL BASED ON DEEP LEARNING

Once the NDT is instantiated, maintaining its fidelity and alignment with the physical network becomes critical. This responsibility is jointly managed by the NDT MANO layer and the monitoring system. To illustrate this, consider a scenario where a functional model within the NDT begins

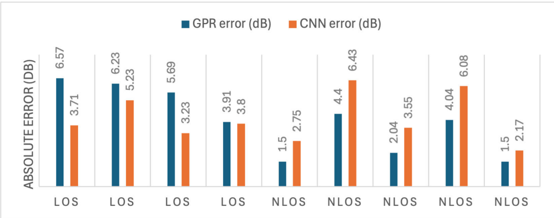

FIGURE 20. GPR and CNN prediction errors of RSRP for validation points.

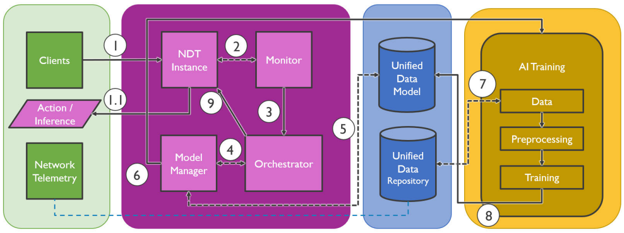

FIGURE 21. NDT Update general workflow.

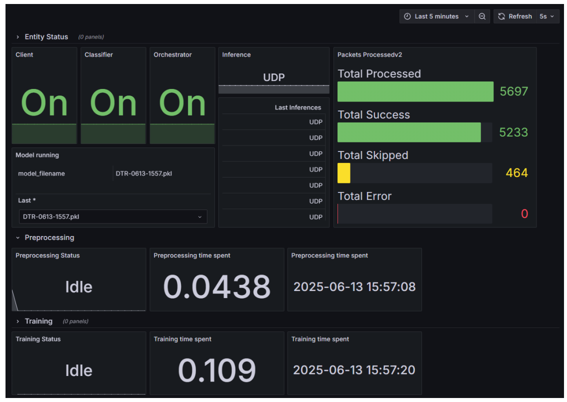

FIGURE 22. Grafana monitor acting as dashboard.

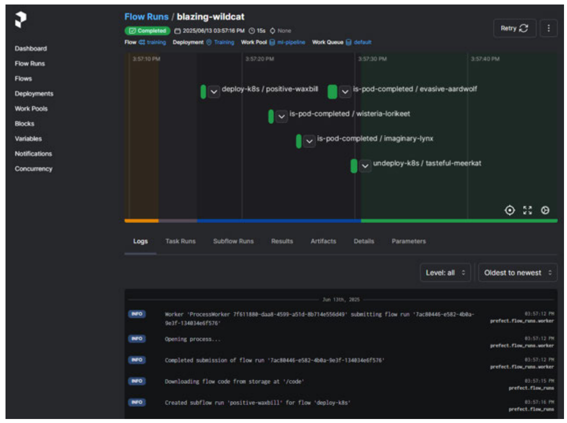

FIGURE 23. Prefect AI training pipeline.

to deviate from its expected performance bounds. In such cases, corrective action must be taken to preserve the NDT's integrity and utility.

As shown in Fig. 21, the system follows an automated workflow to 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 delivers accurate, real-time insights and decisions aligned with network conditions.

For this implementation, we consider an NDT instance that receives packets from the UE (1) and classifies each packet within the network (1.1) for posterior analysis, e.g., for anomaly detection. Meanwhile, the NDT monitor checks the response times, system load, and algorithm accuracy (2). Then, the procedure explained in Section VII-B.2 occurs. An alert is raised once a quality degradation is detected (3). Next, the Orchestrator requests the Model Manager to update the functional model to its latest version (5). The UDM reviews the last available update (5). If the newest version in the UDM is the same as the one already implemented in the NDT instance, the Model Manager requests training in a new version (6). If training is required, the AI Training module uses the data from the UDR to train a new model; this data is collected from network telemetry (7). When the new model is ready, it is stored in the UDM (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 then on, the NDT continues making decisions based on the new model version.

1) IMPLEMENTATION DETAILS

To illustrate this case, an NDT instance was created using DynamicSim [61], Python, Prefect, and Kubernetes. We used a Kaggle dataset [134] in pcap format containing 1,803,059 packet samples across three traffic classes: Transmission Control Protocol (TCP) (999765 samples), UDP (803087 samples), and Multipath TCP (MPTCP) (207 samples). 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, with Docker Desktop and Kubernetes v1.29.1 used for container management.

Figs. 22 and 23 show two screenshots of the implementation of this procedure. On Fig. 22, our Grafana monitor shows the client as a container sending a packet stream to the NDT (classifier on Fig. 22), which classifies the packets using a functional model based on DL. When the NDT monitoring system detects that the accuracy and reliability, measured in the amount of processed packets per second, are not met, it notifies the NDT orchestrator to trigger the pipeline that trains and deploys a new model to replace the current one. Additionally, the image shows the time required for preprocessing and training, based on the data collected up to that point.

Prefect manages the pipeline, an open-source orchestration engine that turns Python functions into production-grade data workflows. Fig. 23 shows a screenshot of the model training process, where the name "blazing-wildcat" is a random token created by Prefect to identify this pipeline. The timeline shows when the pod to perform data preprocessing is started. In Kubernetes jargon, a pod is the smallest deployable computing unit, similar to a container. When these pods finish their work, they are undeployed. Several functional models were used in this process, specifically a Linear Regressor, a Decision Tree Regression (DTR), and a DNN. These models were selected because they balance interpretability, performance, and scalability for different traffic classification scenarios. The models are evaluated in [135]. The NDT Orchestrator selects one model or another depending on the distinct trade-offs between accuracy and efficiency. The DNN achieves the highest accuracy, especially for less frequent classes like MPTCP, but at the cost of significantly higher computational time (up to 19× slower). On the other hand, the DTR provides a strong balance between performance and speed, making it ideal for real-time applications with unencrypted data. In contrast, the DNN becomes the preferred option in more complex or encrypted traffic scenarios, where its superior feature extraction justifies the added computational cost, particularly when hardware acceleration is available.

D. MULTI-OBJECTIVE ROUTE OPTIMIZATION FOR REMOTE DRIVING

In complex use cases, real-time representation of the physical network is not enough to perform network optimization. Such is the case in teleoperated driving, where, besides network-level representations, vehicular traffic representations are also needed [136]. For this application, the NDT should be able to integrate information from other domain-specific DTs or simulators through the Federated MANO (F-MANO) and the simulation framework.

Two complementary implementations demonstrate this capability. The first one represents a comprehensive validation use case of the proposed framework, combining *traffic and network-level representations* through the joint use of OMNeT++ [137], Simu5G [138], and SUMO [139], enabling a realistic *co-simulation of vehicular mobility and network dynamics* [140]. These simulators are coupled through APIs and TraCI interfaces, allowing real-time interaction between vehicle control, network conditions, and AI models. The second approach constructs the environment through systematic data integration, extracting real-world road networks from OpenStreetMap [141] and incorporating 5G base station deployments from the OpenCelliD dataset [142]. This data is processed to generate stochastic network parameters that reflect temporal and spatial variations in connectivity. Both architectures replicate the end-to-end behavior of a 6G-enabled teleoperated driving system, providing realistic and reproducible testbeds for validating intelligent routing algorithms under variable connectivity and latency conditions. This implementation aims to train and evaluate a multi-objective route optimization based on DRL. This algorithm jointly minimizes travel distance and network latency, while maximizing available bandwidth. The system is designed to satisfy the stringent requirements of latency below 30 ms and bandwidth above 20 Mb/s, thereby ensuring reliable, real-time communication essential for teleoperation [143].

The implementation operates across multiple layers of the architecture in Fig. 2, demonstrating how the proposed NDT can integrate mobility, network, and AI-based functional models within a single, coherent workflow. In the digital domain, a functional model processes real-time network metrics and generates optimized routing decisions using a DRL. This model is trained using the joint simulation environment to replicate the complex interactions between physical road infrastructure and telecommunications network performance. Although all the interactions within the joint simulation environment are handled manually or using custom implementations, ideally, the simulation framework should be able to synchronize and manage them effectively. The simulation environment emulates a real-world urban environment as a graph with stochastic network parameters based on a city map, capturing the dynamic nature of the network conditions as they vary temporally and spatially.

The functional model trained within the AI workflow management, then stored and managed within the UDM, demonstrates how RL creates intelligent, adaptive networking solutions that respond to the dynamic requirements of next-generation connected vehicle applications.

1) BASIC AND FUNCTIONAL MODELS

The proposed framework includes Basic models and Functional models and is organized into two key components:

- *Basic models*: representing the 6G mobile network infrastructure, including gNB entities for radio resource management, handover, connection establishment, and the vehicular infrastructure, including cars, roads and topography information. Normally, these basic models are provided by the different simulators for the first work. Additionally, constructing a clustered graph representation of another city based on [141] for road networks, and [142] for stochastic latency and bandwidth parameters modeled using normal distributions to capture temporal network variability in [143].
- *Functional models*: encompassing AI-based modules for both *coverage prediction* and *optimal vehicle routing*, while [143] focuses specifically on *optimal vehicle routing* through DRL.

The *optimal vehicle routing* functional model formulates the route optimization problem as a MDP, where *States* encode vehicle position, destination coordinates, associated network performance metrics (SINR, or alternatively bandwidth and latency), and accumulated path metrics. The *Action* space consists of basic navigation maneuvers (*turn left*, *turn right*, *go straight*) [144], or selecting adjacent intersections, with action masking applied to enforce graph connectivity constraints and prevent invalid moves. The routing policy

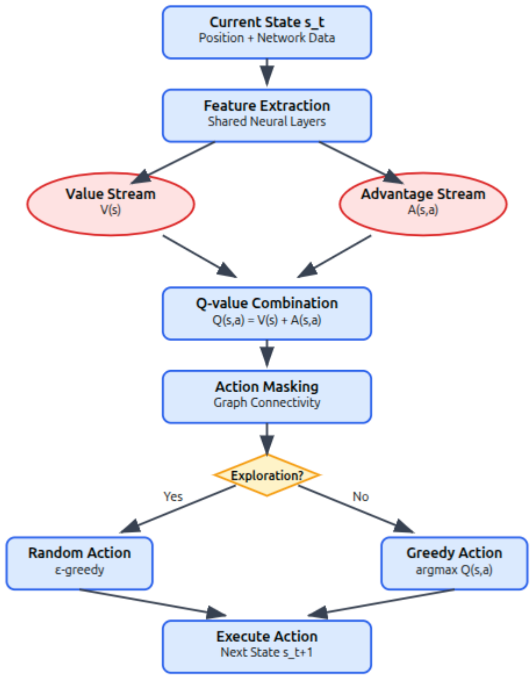

FIGURE 24. Dueling DQN algorithm flowchart showing dual-stream architecture and action selection process with exploration-exploitation balance.

minimizes path length while maintaining strong network coverage, ensuring safe teleoperation under dynamic conditions.

2) DRL-BASED FUNCTIONAL MODELS

We propose two approaches for the route optimization problem. The first approach employs a Dueling Deep Q-Network (DDQN) architecture where its *Reward* function balances competing objectives through a weighted linear combination:

$$R(s, a) = w_d \cdot R_d(s, a) + w_b \cdot R_b(s, a) + w_l \cdot R_l(s, a) \quad (1)$$

where the normalized components are defined as $R_d = -d(e)/d_{max}$ for the distance penalty, $R_b = (b(e, \tau) - b_{min})/(b_{max} - b_{min})$ for the bandwidth reward, and $R_l = -l(e, \tau)/l_{max}$ for the latency penalty. Through systematic experimentation, we determined the optimal weights $w_d = 0.5$, and $w_b = w_l = 0.25$. The workflow of the DDQN algorithm is illustrated in Fig. 24.

The second approach uses the Proximal Policy Optimization (PPO) [145], selected for its stability and efficiency in continuous control problems. The multi-objective reward function integrates several components:

$$r_{\text{tot}}(t) = \alpha r_{\text{shortest}}(t) + \beta r_{\text{coverage}}(t) + \gamma r_{\text{penalty}}(t) + r_{\text{objective}} + r_{\text{illegal}}$$

where r_{shortest} incentivizes progress toward the destination, r_{coverage} encourages higher network coverage, r_{penalty} penalizes insufficient coverage, $r_{\text{objective}}$ rewards successful task completion, and r_{illegal} penalizes unlawful actions. The weights α, β, and γ allow tuning the balance between

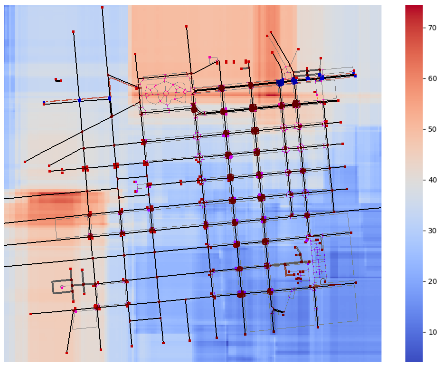

FIGURE 25. Heatmap of SINR values over the city of bari generated by the random forest coverage prediction model.

efficiency, connectivity, and safety. This formulation balances progress toward the destination, network coverage quality, and penalties for unsafe or disconnected states. The algorithm supports three operational scenarios: *Coverage Maximization*, which prioritizes maintaining high SINR values; *Shortest Path Maximization*, which minimizes travel distance while keeping coverage above the threshold; and *Balanced Optimization*, which achieves a trade-off between path length and connectivity, ensuring stable communication and efficient routing.

3) CASE STUDY AND RESULTS

To demonstrate the effectiveness of the proposed approach, a comprehensive case study was conducted, using a realistic city map where a teleoperated vehicle navigates between multiple gNBs. In the first implementation, the *Coverage Prediction* model is needed to map the vehicle positioning with network-related metrics such as SINR, delay, or bandwidth. Specifically, the coverage prediction model was developed using a Random Forest (RF) regression algorithm [146] trained on SINR data collected along simulated vehicle trajectories in the city center of Bari, Italy. This experiment provides a concrete validation of the proposed NDT-based AI workflow under realistic conditions. The trained RF model achieves $R^2 = 0.9993$ and RMSE = 0.3274, demonstrating outstanding predictive performance. The resulting SINR heatmaps (see Fig. 25) provide a spatially continuous estimation of coverage quality, which informs the routing agent with accurate connectivity information.

Regarding the *optimal vehicle routing* model, Fig. 26 depicts the learning outcomes of the PPO model, demonstrating convergence within a few hundred steps and an average cumulative reward exceeding 1400, indicating fast and stable convergence of the PPO agent in all configurations, validating its robustness for multi-objective optimization in stochastic urban network environments. More details of the evaluation and usage of this functional model can be found in Paparella et al. [144].

In addition to the PPO, we evaluated the DDQN model [143] implemented on the city center of Nevers, France. The urban environment was extracted from

FIGURE 26. Learning performance of the PPO agent in the balanced optimization scenario, showing convergence and stable reward accumulation.

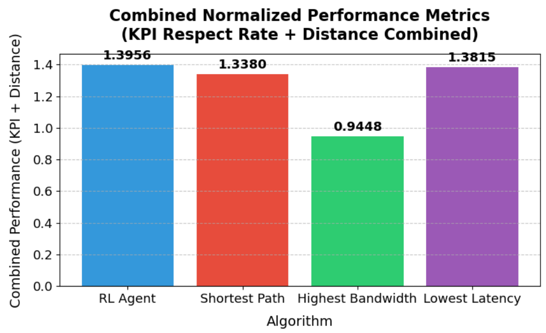

FIGURE 27. Combined performance metric demonstrating overall routing effectiveness across multiple objectives.

OpenStreetMap [141] and transformed into a directed graph through clustering. The network metrics used were latency and bandwidth assigned to each edge based on proximity to 5G base stations from the OpenCelliD dataset [142], with stochastic temporal variations modeled using normal distributions. The DDQN model was evaluated by comparing it against classical routing algorithms: *Shortest Path*, *Lowest Latency*, and *Highest Bandwidth*. To enable a holistic assessment of multi-objective performance, we designed a combined evaluation metric that normalizes and aggregates both the *operational requirement compliance rate* and the *route efficiency*. Under this unified metric, our method achieved the highest overall score, as illustrated in Fig. 27.

These results, obtained through the NDT framework, confirm that the proposed reinforcement learning approach effectively balances competing objectives, achieving a level of performance unattainable by any single-objective optimization strategy. This demonstrates the NDT's capability to serve as a comprehensive testbed for validating functional models before deployment in physical networks.

E. CO-SIMULATION FRAMEWORK

As mentioned in the previous section, several simulators can be coupled to enhance the capabilities of the NDT in predictive modeling. The co-simulation framework of the architecture (cf. Fig. 2) is responsible for the coordination and sequential execution of separate simulators and supports data exchange among the simulators, as mentioned in Section V-B and Section VI-B. For this purpose, protocol translators are used to map the generic *protocol-2* services to specific simulators. This has already been implemented for some of the most well-known simulators in the field of mobile communication. This section presents a reference implementation of the co-simulation framework, with some of the first features already tested successfully.

1) IMPLEMENTED COMPONENTS

One of these simulators is OMNeT++, a widely used network simulator in communication studies. Our protocol translator implementation for OMNET++ supports starting and stopping OMNET++ as well as telling OMNET++ the current position of UEs in the simulation. Additionally, the protocol translator is able to make OMNET++ continue its simulation until a given point in (simulated) time. Inserting and deleting UEs to and from the simulation are also supported at runtime. OMNET++ is started with simulation configuration files that contain further information about the simulated tasks, e.g., the simulation time and transmission power.

A second supported simulator is SUMO, the most well-known mobility simulator for vehicles. Our protocol translator prototype allows starting and stopping SUMO based on given configuration files, processing until a given time, as well as calling the position of all vehicles at runtime. Also, added and removed vehicles are communicated to the co-simulation framework via the protocol translator of SUMO.

Third, a protocol translator prototype for AirMobiSim has been implemented. AirMobiSim computes the position of UAVs based on path information that is given to the simulator at the start of the simulation. The protocol translator allows starting and stopping AirMobiSim and retrieval of the current position of UAVs. Information about newly added and deleted UAVs is sent to the co-simulation framework as well.

In order to make a fully coupled simulation system, also a prototype of the co-simulation framework itself has been implemented. This coordinates the creation and deletion of all necessary simulator instances based on a configuration file. It tells the simulators which configuration files they shall use and starts their execution. Then, it coordinates the execution of each simulator by telling them to simulate to a certain time instance. When the simulators reach this synchronization point, data from the simulators is collected and sent to the relevant other simulators. For example, the number and position of all vehicles is collected from SUMO via its protocol translator and sent to OMNET++ via its protocol translator. Similarly, the UAV position data from AirMobiSim is collected by the co-simulation framework at each synchronization point and sent to OMNET++. Furthermore, during run-time the co-simulation framework also provides a mechanism such that one simulator may influence the behavior of another simulator by sending commands like

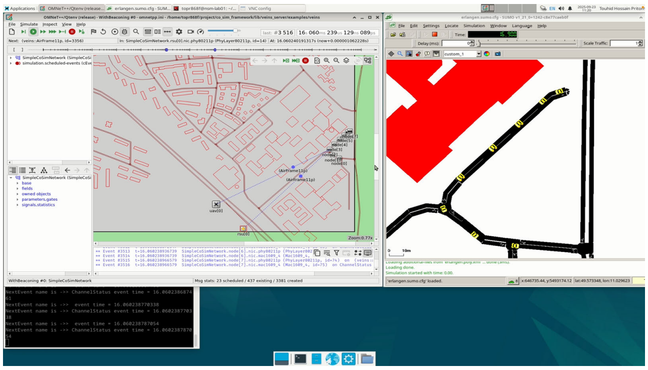

FIGURE 28. Screenshot of the running co-simulation framework.

changing the route of a vehicle or changing its speed at certain points in time. This is done at the synchronization phase by pro-actively asking each simulation to submit any requests it wishes to perform before moving to the next step of the simulation. At the end of the simulation, all simulators are stopped and their instances are disposed.

2) SIMULATION RESULTS

The implemented components have been tested in several configurations. It has been shown that running the co-simulation framework with OMNET++ and SUMO is working, as well as an extended simulation containing OMNET++, SUMO, and AirMobiSim. This can simply be realized by activating AirMobiSim in the co-simulations' configuration file, provided that each simulator has a fitting configuration file that is loaded when starting each simulator. The performance of the co-simulation framework has been measured in order to make sure that the overhead due to the introduction of co-simulation middleware is reasonable in comparison to the execution time of the tightly coupled simulation framework. More information on the performance results can be found in [106]. A screenshot of the running simulation is presented in Fig. 28, where on the left, OMNET++ is shown, and on the right SUMO. The positions of the vehicles in OMNET++ are taken from SUMO via the co-simulation framework. The position of the UAV is taken from AirMobiSim via the co-simulation framework. AirMobiSim has no own GUI. The blue dotted lines represent packet transmissions, showing that OMNET++ correctly uses the vehicle data it gets from SUMO and AirMobiSim, respectively. There is no direct communication between OMNET++ and the other simulators; all interactions are realized via *protocol-2*.

IX. NDT CONSIDERATIONS

Beyond its architectural and functional design, the deployment of an NDT must address several cross-cutting considerations that ensure its reliability, trustworthiness, and interoperability. These include the security and privacy mechanisms necessary to protect sensitive network data, the interfacing and data collection processes that enable seamless integration with real and virtualized network components, and the governance principles that regulate data sharing and model management. Finally, comprehensive evaluation methodologies are required to assess the NDT's performance, accuracy, and operational impact across different lifecycle stages.

A. SECURITY AND PRIVACY CONSIDERATIONS FOR NDTS

An NDT is not only a visualization or analytical tool for network management but a digital copy of the real networks interacting with it bidirectionally. For this reason, NDT is planned to be positioned at the center of the real network's decision-making cycle. Consequently, a breach in the security of an NDT or its data sources/synchronization mechanisms can directly impact the decision-making mechanisms and systemic risks.

An NDT requires collection, consolidation, and processing of large-scale data from heterogeneous network environments including different devices and components. This might create new, use case-specific attack surfaces and privacy threats. The operation of an NDT relies on sensitive information such as real-time telemetry data, performance indicators, and control feedback; therefore, the confidentiality, integrity, and availability of this data must be protected throughout all stages of its life cycle. In addition to that, the integration of data governance, privacy, and compliance mechanisms into the NDT architecture should be considered from the initial design phase. Within this framework, NDTs generate predictive and prescriptive decisions for live networks, so security-by-design must be a guiding principle for NDTs. Security and privacy analysis to be performed on NDT-based network management should identify threat vectors, attack surfaces, assess vulnerabilities and risks to define countermeasures and mitigation scenarios.

The overall architecture of NDT includes physical and digital components and assets. These assets might be vulnerable to various threats. A summary of the threats for these assets is provided in Table 2 [147]. All identified threats must be addressed and mitigated according to the specific use case. While there are established mitigation techniques, solutions, and security controls available to protect each asset, the most appropriate measures should be selected based on the asset's risk profile and operational context. Additionally, the applicability and effectiveness of any chosen controls must be evaluated against real-world operational constraints to ensure they provide the intended protection with the required complexity and cost [148].

B. NDT GOVERNANCE

NDT Governance plays a critical role in ensuring the secure, fair, and transparent management of data, models, and interactions among multiple stakeholders. Effective governance establishes the principles, policies, and technical mechanisms required to maintain trust, accountability, and regulatory compliance across a distributed NDT ecosystem. It defines

TABLE 2. Threat categories in NDT systems.

Threat category	Brief description
Data storage and NDT repository	Unauthorized access to the NDT and associated network functions. Critical or confidential data, basic/functional models may be stolen or copied.
Sensors or software in the data collection process	An adversary may spoof or manipulate data traffic between the NDT and the physical network, leading to corrupted model states or false decisions.
Physical network component	An adversary can damage or interfere with physical assets
Data flow between physical assets and NDT	An attacker can disrupt synchronization, listen to or alter data flow. They can also launch a denial-of-service (DoS) attack
Network management models and functions	An insider attacker or an attacker with unauthorized access can change configurations, security policies, or steal confidential data
Application layer interfaces	Application layer data can be altered and intercepted; NDT can be affected by DoS attacks

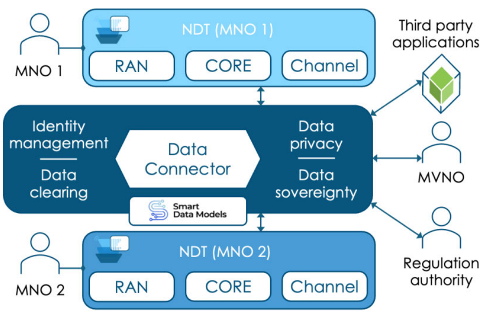

FIGURE 29. NDT governance through the data space.

how data is accessed, shared, and monetized while safeguarding privacy and sovereignty, key aspects for the operational viability of large-scale NDT deployments.

Unlike other sectors such as the European Network of Transmission System Operators for Electricity in the energy domain, or even certain wired communication infrastructures, the radio access network lacks a centralized information hub. Consequently, the governance of NDTs for 6G networks must adopt a decentralized approach. In such a framework, each MNO manages its own NDT instance while selectively sharing data with other MNOs' NDTs when cooperation or federation is required. Additionally, MNOs are responsible for providing relevant information to regulatory authorities, Virtual MNOs, and application providers. These exchanges take place within the framework of a regulated data space [149], as introduced in Section V-A.1, which specifies the technical and legal rules governing interactions between MNOs and external stakeholders (see Fig. 29).

Nevertheless, providing the national network regulator with a partial but relevant view of operations is essential for ensuring compliance with regulations and legal requirements. They should take into account the different governance:

- *Transparency & accountability.* Clear policies must define who can access, modify, or share data within the NDT, ensuring traceability and accountability for all actions.
- *Interoperability & standards.* Adopting common data models, interfaces, and protocols enables seamless interaction between different NDTs and stakeholders, fostering collaboration and reducing fragmentation.
- *Data sovereignty & privacy.* Governance should respect data ownership rights and enforce privacy protections, especially when sharing sensitive information across MNOs, regulators, and third parties.
- *Role-based access control.* Access to NDT data and functionalities should be granted based on predefined roles, ensuring that each actor, whether MNO, Virtual MNO, or regulator, has appropriate permissions aligned with their responsibilities.
- *Regulatory alignment.* The framework must align with national and international regulations, facilitating audits and ensuring that data exchanges meet legal/industry standards.
- *Dynamic adaptability.* As technologies and requirements evolve, the governance model should be flexible enough to accommodate new use cases, stakeholders, and regulatory changes without disrupting existing operations.
- *Trust & security.* Robust security measures, such as encryption and authentication, are critical to protect data integrity and build trust among all participants in the NDT ecosystem.
- *Federated data sharing.* Mechanisms for secure, consent-based data sharing should be established, allowing stakeholders to contribute and consume data while maintaining control over their own assets.

Additionally, related domains, such as building information and transportation, can influence the NDT, potentially forming a federation of diverse NDTs. The key challenge lies in establishing a tailored governance framework to manage the flow of data into and out of this NDT federation, taking into account the distinct roles of both the NDTs and the actors involved.

C. DATA COLLECTION REQUIREMENTS AND INTERFACES FOR NDTS

Data collection is performed across the physical network, which is trending to scale up exponentially, with more and more devices connected to the internet, turning the IoT paradigm into the Internet of Everything, a term coined by Cisco [150] to describe how more people are connected through new device types, generating new types of information. This creates challenges for the NDT of the future 6G network, necessitating a new level of scalability, dealing with both the exponential increase of connected devices and the wide range of data types generated by these heterogeneous devices. All of this needs to be done in a lightweight and efficient manner, to make sure the NDT delivers its enhanced decision-making capabilities towards network and service management.

As discussed in Section IV-B.2, the data collection framework seeks to ensure that data is gathered, processed and stored in a way that guarantees interoperability, accuracy and efficiency across the modeling and simulation tasks of the NDT. Some recommendations have been highlighted during the development of the architecture, based on both research initiatives and by standardization body activities, such as the IETF [48]; this led to the creation of a set of requirements that must be fulfilled by the NDT to make sure that these challenges are addressed.

Data collection performed by the NDT must be target driven and context-aware, focusing on specific operational needs rather than monitoring all available data; this can be done by operators, defining key parameters that ensure only pertinent data is collected (such as operational status and configuration of devices, events, logs, lifecycle operation data, user data and service data) rather than indiscriminately gathering all available data, making sure resource utilization of the network is optimized. Extending data collection across multiple data domains, going beyond the traditional service-based network architecture to the Cloud-Edge-IoT continuum, promotes the integration of multiple devices and enhances the functionality of the NDT. Effective naming and caching of data can also contribute to improve access speeds and reduce redundant entries within the data repository.

The NDT should also be equipped with multiple tools that can handle data collection in a heterogeneous-data-sources environment, integrating mechanisms that harmonize different data formats in a lightweight and efficient manner, standardizing data flows to maintain consistency and accuracy; this contributes to a more efficient data processing framework, minimizing the cost of computing, storage and communication bandwidth, as well as the likelihood of data processing and utilization errors across applications and services. Data from various sources must also be easily integrated into the data repository, allowing for the NDT to have, at all times, a comprehensive and up-to-date view of the network for effective analysis and decision-making processes.

Interfaces between the data collection framework and the NDT should also be open and standardized, avoiding any hardware/software dependencies and ensuring interoperability between components, allowing for the management and configuration of different timescales across devices and applications, through a secure and reliable channel that exchanges information in a federated manner, and future-proofed for extensibility and backwards compatibility. This framework is further enhanced by supporting multiple communication protocols, enabling both compatibility with legacy systems and future device interfaces.

Finally, beyond multi-device data collection, the NDT must also be capable of handling multi-destination delivery, as data collected from the same source could be requested by multiple instances of the NDT in the federation. Data sharing is managed by a robust set of access policies, based on the aforementioned federation paradigm, enhancing fault tolerance, access speeds, and scalability, while ensuring privacy and security of any sensitive information. These requests must be fulfilled as efficiently as possible, to maintain the scalability requirements, and ensure that the systems receive the data in a timely manner, through multicast communication and intelligent routing systems, for the prerequisites of the applications and services.

D. NDT EVALUATION CONSIDERATIONS

The evaluation process plays a crucial role in the NDT framework. To this purpose, ITU-T Y.3091 [41] Recommendation highlights three main requirements for an NDT:

- *Fidelity:* how accurate the metrics from performance models are with the physical network;
- *Efficiency:* has two sub-requirements: (i) performance models should be faster than real-time physical network; (ii) models should be easy to deploy and consume rational resources;
- *Flexibility:* means that the NDT should be flexible to provide service on-demand according to various network applications by selecting a variety of cross-domain resources on demand, flexibly collecting and storing data, combining different data models and interacting with other DTs.

In addition to the requirements, the ITU-T Y.3091 recommendation provides a framework for assessing the maturity and functionality of an NDT in a hierarchical and recursive fashion. More specifically, five NDT capability levels are defined:

- *L1 Representation level:* NDT can realize one-way mapping from the physical network to the virtual twin.
- *L2 Interaction level:* Based on L1, the control channel is added from the virtual twin to the physical network.
- *L3 Prediction level:* Based on L2, the virtual twin analyzes the characteristics and trends of the data collected from the physical network and uses strategies to infer indicators.
- *L4 Optimization level:* Based on L3, the virtual twin uses AI algorithms, expert knowledge, big data analysis, and other intelligent technologies for optimization.
- *L5 Autonomy level:* As the ideal goal of NDT, the virtual twin and the physical network live in symbiosis with each other.

Based on the above levels, the maturity of the NDT can be assessed for the following along the following set of dimensions: Data service; Digital Twin modeling; Interactive mapping; Intelligence user experience; and Trustworthiness. Each dimension is associated with a set of evaluation indicators for each of which a specific level, as detailed in [41], must be assessed. The indicators are designed following the SMART principles, i.e. Specific, Measurable, Assignable, Realistic and Time-related, as detailed in [151]. Moreover, evaluations at different levels are strongly interrelated and the synthetic capability level of the NDT is calculated in a recursive fashion. The evaluation framework defines a set of dimensions and indicators, each specifying the minimum conditions required to attain the targeted NDT capability.

In order to assess the level of each evaluation indicator, a set of KPIs should be established, and each KPI needs to be linked to one or more indicators. Such KPIs can be both qualitative and quantitative. Qualitative KPIs are related to aspects that cannot be measured, such as the ability to replicate a real network, the coverage of the basic and functional models as opposed to a specific use case, the flexibility of the NDT in the integration with other DTs, the conformance of methods adopted to ensure security and privacy for data in transit and at rest and the contribution to the standardization efforts. On the contrary, quantitative KPIs are mostly related to network performance metrics that can be numerically determined during specific simulation campaigns, such as data rate, latency, packet loss, service reliability, spectral efficiency, edge node utilization, and energy consumption.

In order to compute the KPIs, a data collection must be performed during the simulation campaign. To this purpose, consolidated methodologies, such as FESTA-V [152] can be used. More specifically, FESTA-V provides a clear framework made of three steps: (i) PREPARE, in which the simulation scenarios and the data to be collected are defined, (ii) USAGE, in which the simulation is performed and the data are collected and (iii) EVALUATION and IMPACT, in which the data are processed and the KPIs are computed. In this context, each KPI should be linked to its corresponding Evaluation Indicator. Finally, the individual capability level at each relevant Evaluation Indicator and Dimension level and, then, the overall maturity of the NDT, can be computed by using the recursive formula detailed in [41].

X. CONCLUSION AND FUTURE WORK

This paper proposes an NDT framework integrating AI, simulation, and adaptive management for future 6G systems, treating the twin as a lifecycle-managed capability rather than a monolithic artifact. It brings together a harmonized data pipeline, reusable models, and an orchestration layer that instantiates scoped NDT instances on demand, enabling closed-loop control from cloud to far edge. In a field where NDT work across the literature and industry is fragmented, this interoperable, AI-native architecture offers a coherent way forward by harmonizing data and unifying model management. In doing so, it supports seamless transitions between data-driven and simulation-driven reasoning, enables closed-loop analysis and control from cloud to far-edge, and keeps scalability, interoperability, and governance at the forefront. Early prototyping across telemetry ingestion, model training and re-training, network and mobility simulation, suggests that the approach is feasible and that it offers a practical path for safe "what-if" experimentation and repeatable actuation.

The framework is designed to be transferable across domains: its modular decomposition, emphasis on model catalogs and reusable pipelines, and alignment with contemporary orchestration practices make it suitable for scenarios where fidelity, responsiveness, reliability, and energy efficiency must be balanced. By elevating the NDT concept to a managed, on-demand capability, the proposal provides a concrete blueprint that can be replicated and extended, accelerating the move from concept to operation in real systems.

Future work will concentrate on end-to-end validation through concrete and tangible use-cases that exercise the full lifecycle: instrumenting real networks, harmonizing data flows, instantiating targeted NDT instances, training and adapting basic and functional models with auditable KPIs. As part of 6G-TWIN, this will be pursued through two use-cases, which will serve as proving grounds to showcase repeatable results. Beyond demonstrating performance and feasibility, these demonstrators are intended to deliver reusable assets—data schemas, model and experiment catalogs, and MLOps and co-simulation recipes—that facilitate uptake beyond the project and help translate the framework into deployable practice.

REFERENCES

[1] P. Jonsson, P. Cerwall, A. Lundvall, D. von Koch, and S. Davis. (Jun. 2025). *Ericsson Mobility Report*. [Online]. Available: https://elements.visualcapitalist.com/wp-content/uploads/2025/06/1750752177192.pdf

[2] D.-H. Tran et al., "Network digital twin for 6G and beyond: An end-to-end view across multi-domain network ecosystems," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 6866–6911, 2025.

[3] A. Zaki–Hindi, J.-S. Sottet, I. Turcanu, and S. Faye, "Network digital twins for 6G: Defining data taxonomy and data models," in *Proc. IEEE Conf. Standards for Commun. Netw. (CSCN)*, Nov. 2024, pp. 124–128.

[4] S. Faye et al., "A functional framework for network digital twins," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2025, pp. 1–6.

[5] (Dec. 2025). *D1.1–6G-TWIN Architecture and Technical Foundations*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D1.1-Architecture-and-Technical-Foundations-final.pdf

[6] (Dec. 2025). *D2.1–Data Governance, Privacy and Harmonization*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D2.1-Deliverable-Data-governance-privacy-and-harmonization-final.pdf

[7] (Dec. 2025). *D1.Secured, Scalable and Distributed Data Exposure and Collection Framework*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D1.2-Secured-scalable-and-distributed-data-exposure-and-collection-framework-final.pdf

[8] (Dec. 2025). *D2–Basic Models*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D2.2-Basic-Models-final.pdf

[9] (Dec. 2025). *D5.1–Evaluation Methodology, Planning and Coordination*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D5.1-Deliverable-Evaluation-methodology-planning-and-coordination-final.pdf

[10] (Dec. 2025). *D2.3–Functional Models*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D2.3-Functional-Models-final.pdf

[11] (Dec. 2025). *D1.3–Frameworks for Zero-Touch Service and Network Management and the Orchestration of Its AI-Based NF and NS*. [Online]. Available: https://6g-twin.eu/wp-content/uploads/2025/12/6G-TWIN-D1.3-Frameworks-for-zero-touch-service-network-management-and-the-orchestration-of-its-AI-based-NF-and-NS-initial-final.pdf

[12] (Feb. 2022). *Digital Twin Network-Requirements and Architecture*. [Online]. Available: https://handle.itu.int/11.1002/1000/14852
[13] *Study on Management Aspects of Network Digital Twin*, document TR 28.915, 3GPP, 2024.
[14] K. Sun et al., "Research report on digital twin ran use cases," O-RAN Next Generation Research Group (nGRG), Germany, Tech. Rep. RR-2024-07, Jul. 2024.
[15] *Zero-touch Network and Service Management (ZSM); Network Digital Twin*, ETSI, Sophia Antipolis, France, Feb. 2024.
[16] C. Zhou et al., "Network digital twin: Concepts and reference architecture," Internet Research Task Force (IRTF), Tech. Rep., 2025.
[17] *Simplifying 5G With a Network Digital Twin*, Spirent, Crawley, U.K., 2022.
[18] *Wireless Network Digital Twin As A Service*, ZTE, Shenzhen, China, 2023.
[19] *6G Networks*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.ericsson.com/en/6g
[20] *Deutsche Telekom, Google Cloud, and Ericsson Demonstrate Network Transformation Milestone With 5G Cloud-Native Network Pilot*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.telekom.com/en/media/media-information/archive/5g-cloud-native-pilot-shows-efficiency-1026992
[21] J. Hoydis, F. A. Aoudia, A. Valcarce, and H. Viswanathan, "Toward a 6G AI-native air interface," *IEEE Commun. Mag.*, vol. 59, no. 5, pp. 76–81, May 2021.
[22] *Dynamic Digital Twin: Enabling Real-Time Monitoring of Rapidly Changing Environments*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.nokia.com/bell-labs/research/air-lab/modelling-optimization/dynamic-digital-twin/
[23] Y. Chen et al., "NetGPT: An AI-native network architecture for provisioning beyond personalized generative services," *IEEE Netw.*, vol. 38, no. 6, pp. 404–413, Nov. 2024.
[24] *Huawei Releases the Intelligent World 2030 Report to Explore Trends in the Next Decade*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.huawei.com/en/news/2021/9/huawei-releases-intelligent-world-report-2030
[25] *The Digital Twin Technology Applied to 6G Communication*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.keysight.com/us/en/assets/3124-1789/white-papers/The-Digital-Twin-Technology-Applied-to-6G-Communication.pdf
[26] X. Wang, X. Li, and V. C. M. Leung, "Artificial intelligence-based techniques for emerging heterogeneous network: State of the arts, opportunities, and challenges," *IEEE Access*, vol. 3, pp. 1379–1391, 2015.
[27] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.
[28] Y. Zhou, R. Zhang, J. Liu, T. Huang, Q. Tang, and F. R. Yu, "A hierarchical digital twin network for satellite communication networks," *IEEE Commun. Mag.*, vol. 61, no. 11, pp. 104–110, Nov. 2023.
[29] I. Vila, 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.
[30] N. P. Kuruvatti, M. A. Habibi, S. Partani, B. Han, A. Fellan, and H. D. Schotten, "Empowering 6G communication systems with digital twin technology: A comprehensive survey," *IEEE Access*, vol. 10, pp. 112158–112186, 2022.
[31] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.
[32] D. Lehner et al., "Digital twin platforms: Requirements, capabilities, and future prospects," *IEEE Softw.*, vol. 39, no. 2, pp. 53–61, Mar. 2022.
[33] N. Bicocchi, M. Fogli, C. Giannelli, M. Picone, and A. Virdis, "Requirements and design architecture for digital twin end-to-end trustworthiness," *IEEE Internet Comput.*, vol. 28, no. 4, pp. 31–39, Jul. 2024.
[34] P. Bellavista, N. Bicocchi, M. Fogli, C. Giannelli, M. Mamei, and M. Picone, "ODTE: A metric for digital twin entanglement," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2377–2390, 2024.
[35] M. Polese et al., "Colosseum: The open RAN digital twin," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 5452–5466, 2024.
[36] H. Gao, P. Kyösti, X. Zhang, and W. Fan, "Digital twin enabled 6G radio testing: Concepts, challenges and solutions," *IEEE Commun. Mag.*, vol. 61, no. 12, pp. 88–94, Dec. 2023.
[37] P. Soto et al., "Designing the network intelligence stratum for 6G networks," *Comput. Netw.*, vol. 254, Dec. 2024, Art. no. 110780.
[38] Y. Wang, J. Sun, X. Sun, and S. Chen, "End to end AI architecture for next generation network," *IEEE Wireless Commun.*, vol. 31, no. 1, pp. 86–92, Feb. 2024.
[39] S. Zeb, M. A. Rathore, S. A. Hassan, S. Raza, K. Dev, and G. Fortino, "Toward AI-enabled NextG networks with edge intelligence-assisted microservice orchestration," *IEEE Wireless Commun.*, vol. 30, no. 3, pp. 148–156, Jun. 2023.
[40] S. Faye et al., "Integrating network digital twinning into future AI-based 6G systems: The 6G-TWIN vision," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit*, Jun. 2024, pp. 883–888.
[41] *Digital Twin Network-Capability Levels and Evaluation Methods*, document ITU-T Y.3091, Dec. 2023.
[42] S. Lin, J. Zhou, and M. Yu, "An LLM-based agentic framework for accessible NetworkControl," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 53, no. 2, pp. 15–20, Aug. 2025.
[43] N. Alliance, "Network architecture evolution towards 6G," NGMN Alliance, Dusseldorf, Germany, Tech. Rep., 2023.
[44] Y.-B. Lin, C.-C. Tseng, and M.-H. Wang, "Effects of transport network slicing on 5G applications," *Future Internet*, vol. 13, no. 3, p. 69, Mar. 2021.
[45] 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. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.
[46] 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, Jul. 2022, Art. no. 103362.
[47] *Autonomous Networks–Architecture Framework*, document ITU Y.3061, Dec. 2023.
[48] C. Zhou, D. Chen, P. Martinez-Julia, and Q. Ma. (Jul. 2023). *Data Collection Requirements and Technologies for Digital Twin Network*. [Online]. Available: https://datatracker.ietf.org/doc/draft-zcz-nmrg-digitaltwin-data-collection/03/
[49] *Smart Data Models*. Accessed: Dec. 12, 2025. [Online]. Available: https://smartdatamodels.org/
[50] M. Fedor, M. L. Schoffstall, J. R. Davin, and D. J. D. Case, *Simple Network Management Protocol (SNMP)*, document RFC 1157, May 1990. [Online]. Available: https://www.rfc-editor.org/info/rfc1157
[51] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, *Network Configuration Protocol (NETCONF)*, document RFC 6241, Jun. 2011.
[52] *Traffic Monitoring Using Sflow*. Accessed: Dec. 12, 2025. [Online]. Available: https://sflow.org/sFlowOverview.pdf
[53] S. Panchen, N. McKee, and P. Phaal, *InMon Corporation's SFlow: A Method for Monitoring Traffic in Switched and Routed Networks*, document RFC 3176, Sep. 2001.
[54] J. Scudder, R. Fernando, and S. Stuart, *BGP Monitoring Protocol (BMP)*, document RFC 7854, Jun. 2016.
[55] G. Fioccola, M. Cociglio, G. Mirsky, T. Mizrahi, and T. Zhou, *Alternate-Marking Method*, document RFC 9341, Dec. 2022.
[56] F. Brockners, S. Bhandari, and T. Mizrahi, *Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)*, document RFC 9197, May 2022.
[57] B. Claise, A. Johnson, and J. Quittek, *Packet Sampling (PSAMP) Protocol Specifications*, document RFC 5476, Mar. 2009.
[58] *O-RAN Architecture Description*, O-RAN Alliance, Buschkauler, Germany, Oct. 2025.
[59] *Network Functions Virtualisation (NFV); Management and Orchestration*, ETSI, Sophia Antipolis, France, 2014.
[60] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.
[61] P. Soto et al., "Network intelligence for NFV scaling in closed-loop architectures," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 66–72, Jun. 2023.
[62] *Open Source Mano (OSM)*. Accessed: Dec. 12, 2025. [Online]. Available: https://osm.etsi.org/
[63] *Open Network Automation Platform*. Accessed: Dec. 12, 2025. [Online]. Available: https://onap.org/
[64] *Cloudify*. Accessed: Dec. 12, 2025. [Online]. Available: https://docs.cloudify.co/

[65] *Network Functions Virtualisation (NFV); Report on Network Slicing Support With ETSI NFV Architecture Framework*, document ETSI GR NFV-EVE 012 V3.1.1, Dec. 2017.

[66] T. Usländer et al., "Symbiotic evolution of digital twin systems and dataspaces," *Automation*, vol. 3, no. 3, pp. 378–399, Aug. 2022.

[67] (2020). *Eu Data Strategy 2020*. [Online]. Available: https://dataeconomy.eu/eu-data-strategy-2020/

[68] (2022). *IDS Reference Architecture Model 4.0*. [Online]. Available: https://internationaldataspaces.org/

[69] *Management and Orchestration; 5G Performance Measurements*, document TS 28.552, 3GPP, 2022.

[70] *5G;NG-RAN; Architecture Description*, document TS 38.401, 3GPP, 2020.

[71] *Machine Learning Marketplace Integration in Future Networks Including IMT-2020*, document Y.3176, Feb. 2020.

[72] *5G Network Resource Model (NRM); Stage 2 and Stage 3*, document TS 28.541, 3GPP, Dec. 2020.

[73] *Telecommunication Management; Performance Management (PM); Concept and Requirements*, document TS 32.401, 3GPP, 2011.

[74] *Generic Network Resource Model (NRM) Integration Reference Point (IRP) (Release 18)*, document TS 28.622, 3GPP, Mar. 2024.

[75] *Automation Systems and Integration-Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles*, Standard ISO TC 184/SC 4, 2021.

[76] F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji, "Digital twin modeling," *J. Manuf. Syst.*, vol. 64, pp. 372–389, Jul. 2022.

[77] J. Wen, B. Gabrys, and K. Musial, "Toward digital twin oriented modeling of complex networked systems and their dynamics: A comprehensive survey," *IEEE Access*, vol. 10, pp. 66886–66923, 2022.

[78] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, "Universal digital twin—A dynamic knowledge graph," *Data-Centric Eng.*, vol. 2, p. e14, Jul. 2021.

[79] R. Verdecchia, L. Scommegna, E. Vicario, and T. Pecorella, "Designing a future-proof reference architecture for network digital twins," in *Proc. Eur. Conf. Softw. Archit.* Cham, Switzerland: Springer, 2024, pp. 287–306.

[80] H. Hong et al., "NetGraph: An intelligent operated digital twin platform for data center networks," in *Proc. ACM SIGCOMM Workshop Netw.-Appl. Integr.*, Aug. 2021, pp. 26–32.

[81] S. Gil, P. H. Mikkelsen, C. Gomes, and P. G. Larsen, "Survey on open-source digital twin frameworks—A case study approach," *Software: Pract. Exper.*, vol. 54, no. 6, pp. 929–960, Jan. 2024.

[82] *Digital Twins Definition Language (dtdl)*. Accessed: Dec. 12, 2025. [Online]. Available: https://azure.github.io/opendigitaltwins-dtdl/DTDL/v3/DTDL.v3.html

[83] A. Meijers, *Hands-On Azure Digital Twins: A Practical Guide to Building Distributed IoT Solutions*. Birmingham, U.K.: Packt Publishing, 2022.

[84] K. Schweichhart, "Reference architectural model industrie 4.0 (rami 4.0)—An Introduction," Plattform Industrie 4.0, Berlin, Germany, Tech. Rep., 2016.

[85] *Eclipse Basyx*. Accessed: Dec. 12, 2025. [Online]. Available: https://projects.eclipse.org/projects/dt.basyx

[86] J. Robles, C. Martín, and M. Díaz, "OpenTwins: An open-source framework for the development of next-gen compositional digital twins," *Comput. Ind.*, vol. 152, Nov. 2023, Art. no. 104007.

[87] *Scorpio NGSI-LD Broker*. [Online]. Available: https://github.com/ScorpioBroker

[88] U. Ahle and J. J. Hierro, "FIWARE for Data Spaces," in *Designing Data Spaces*. Cham, Switzerland: Springer, 2022, pp. 395–417.

[89] *Bring Your Data To Live*. Accessed: Dec. 12, 2025. [Online]. Available: https://greycat.io/

[90] T. Hartmann, F. Fouquet, A. Moawad, R. Rouvoy, and Y. Le Traon, "GreyCat: Efficient what-if analytics for data in motion at scale," *Inf. Syst.*, vol. 83, pp. 101–117, Jul. 2019.

[91] *A Python-Based Low-Modeling Low-Code Open-Source Platform for Smart and AI-Enhanced Software*. Accessed: Dec. 12, 2025. [Online]. Available: https://github.com/BESSER-PEARL/BESSER

[92] I. Alfonso et al., "Building BESSER: An open-source low-code platform," in *Enterprise, Business-Process and Information Systems Modeling*. Cham, Switzerland: Springer, 2024, pp. 203–212.

[93] 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.

[94] P. Almasan et al., "Network digital twin: Context, enabling technologies, and opportunities," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 22–27, Nov. 2022.

[95] L. Hui, M. Wang, L. Zhang, L. Lu, and Y. Cui, "Digital twin for networking: A data-driven performance modeling perspective," *IEEE Netw.*, vol. 37, no. 3, pp. 202–209, May 2023.

[96] N. Chukhno et al., "Models, methods, and solutions for multicasting in 5G/6G mmWave and sub-THz systems," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 119–159, 1st Quart., 2024.

[97] A. Boudi, M. Bagaa, P. Pöyhönen, T. Taleb, and H. Flinck, "AI-based resource management in beyond 5G cloud native environment," *IEEE Netw.*, vol. 35, no. 2, pp. 128–135, Mar. 2021.

[98] A. Micu, L. Connolly, A. O'Connor, E. O'Brien, and C. de Paor, "Modelling probabilistic digital twins of complex inland waterway transportation systems using Bayesian networks," in *Proc. Transport Res. Arena Conf.* Cham, Switzerland: Springer, Oct. 2025, pp. 874–879.

[99] T. Tripura, A. S. Desai, S. Adhikari, and S. Chakraborty, "Probabilistic machine learning based predictive and interpretable digital twin for dynamical systems," *Comput. Struct.*, vol. 281, Jun. 2023, Art. no. 107008.

[100] M. Segovia and J. Garcia-Alfaro, "Design, modeling and implementation of digital twins," *Sensors*, vol. 22, no. 14, p. 5396, Jul. 2022.

[101] R. F. Nielsen, "Hybrid modelling strategies for development of digital twins," Ph.D. dissertation, Technical Univ. Denmark, Lyngby, Denmark, 2021.

[102] E. Evgenieva, A. Vlahov, A. Ivanov, V. Poulkov, and A. Manolova, "A comprehensive survey of 6G simulators: Comparison, integration, and future directions," *Electronics*, vol. 14, no. 16, p. 3313, Aug. 2025.

[103] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Trans. Mobile Comput.*, vol. 10, no. 1, pp. 3–15, Jan. 2011.

[104] D. S. Buse, M. Schettler, N. Kothe, P. Reinold, C. Sommer, and F. Dressler, "Bridging worlds: Integrating hardware-in-the-loop testing with large-scale VANET simulation," in *Proc. 14th Annu. Conf. Wireless Demand Netw. Syst. Services (WONS)*, Feb. 2018, pp. 33–36.

[105] T. Queck, B. Schunemann, I. Radusch, and C. Meinel, "Realistic simulation of V2X communication scenarios," in *Proc. IEEE Asia–Pacific Services Comput. Conf.*, Dec. 2008, pp. 1623–1627.

[106] T. H. Pritom, B. Hensel, M. Franke, and C. Sommer, "Towards a generic middleware-based interface for mobile communication simulation," in *Proc. 29th Int. Symp. Distrib. Simul. Real Time Appl. (DS-RT)*, Sep. 2025, pp. 1–2.

[107] T. Hardes, D. Logan, T. H. Pritom, and C. Sommer, "Towards an open source fully modular multi unmanned aerial vehicle simulation framework," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jul. 2022, pp. 284–289.

[108] *Driving Profitability Through Data*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.teoco.com/

[109] *Atoll Allows Mobile Operators To Plan and Optimise 5G NR Networks Deployed in Both Nsa (non Standalone) and SA (standalone) Modes, Together With 4G and Legacy 2G and 3G Layers*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.forsk.com/4g5g

[110] *The Industry Standard for Designing In-Building Wireless Networks*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.ibwave.com/ibwave-design-enterprise/

[111] *NS-3 Network Simulator*. [Online]. Available: https://www.nsnam.org/

[112] *Lte User Plane Simulation Model for Inet and OMNET++*. Accessed: Dec. 12, 2025. [Online]. Available: https://simulte.omnetpp.org/

[113] F. Kaltenberger, "5G new radio in OpenAirInterface," in *Proc. 5th OAI Workshop*, Beijing, China, Jun. 2018, p. 18.

[114] M. K. Müller et al., "Flexible multi-node simulation of cellular mobile communications: The Vienna 5G system level simulator," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–14, Sep. 2018.

[115] X. He et al., "Data-driven hybrid modeling for digital twin of large-scale structures with local nonlinearities," *J. Mech. Design*, vol. 147, no. 12, Dec. 2025, Art. no. 121703.

[116] S. Robitzsch et al., "Standardisation assessment of a digital twin-based multi-domain deterministic communications system," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2024, pp. 1–7.

[117] L. Das, B. Gjorgiev, and G. Sansavini, "Uncertainty-aware deep learning for digital twin-driven monitoring: Application to fault detection in power lines," 2023, *arXiv:2303.10954*.

[118] B. Qin et al., "Machine and deep learning for digital twin networks: A survey," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34694–34716, Nov. 2024.
[119] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Found. Trends Mach. Learn.*, vol. 16, no. 1, pp. 1–118, Jan. 2023.
[120] L. Schena, P. A. Marques, R. Poletti, S. Ahizi, J. Van den Berghe, and M. A. Mendez, "Reinforcement twinning: From digital twins to model-based reinforcement learning," *J. Comput. Sci.*, vol. 82, Oct. 2024, Art. no. 102421.
[121] S. Malhotra, F. Yashu, M. Saqib, D. Mehta, J. Jangid, and S. Dixit, "Deep reinforcement learning for dynamic resource allocation in wireless networks," 2025, *arXiv:2502.01129*.
[122] A. Seipolt et al., "Reinforcement learning and digital twin-driven optimization of production scheduling with the digital model playground," *Discover Internet Things*, vol. 4, no. 1, p. 34, Dec. 2024.
[123] *Digital Twin for Management and Orchestrationin IMT-2020 Networks and Beyond*, document ITU Y.309, Dec. 2024.
[124] *Autonomous Networks: Empowering Digital Transformation For Smart Societies and Industries*, TMForum, Morristown, NJ, USA, 2020.
[125] S. Yoon, "Building digital twinning: Data, information, and models," *J. Building Eng.*, vol. 76, Oct. 2023, Art. no. 107021.
[126] 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 *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, Oct. 2021, pp. 225–228.
[127] P. Natzschka, B. Hensel, and C. Sommer, "Simopticon: Automated optimization of vehicular platooning controllers," *Ad Hoc Netw.*, vol. 170, Apr. 2025, Art. no. 103781.
[128] M. A. Zaharia et al., "Accelerating the machine learning lifecycle with mlflow," *IEEE Data Eng. Bull.*, vol. 41, pp. 39–45, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:83459546
[129] H. Song, F. Qin, P. Martinez-Julia, L. Ciavaglia, and A. Wang, *Network Telemetry Framework*, document RFC 9232, May 2022.
[130] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management toward 5G radio access networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1656–1686, 3rd Quart., 2016.
[131] *Zero-touch Network and Service Management (ZSM); Reference Architecture*, ETSI, Sophia Antipolis, France, 2019.
[132] 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.
[133] A. Zaki-Hindi, J.-S. Sottet, S. Kumar, I. Turcanu, and S. Faye, "A modular network digital twin for radio coverage prediction: From theory to practice," in *Proc. IEEE Global Commun. Conf.*, Dec. 2025, pp. 1–16.
[134] A. Yar. (May 2023). *Internet Traffic Data Set*. [Online]. Available: https://www.kaggle.com/datasets/asfandyar250/network
[135] J. M. Jimenez, A. Gavrielides, N. Slamnik–Krijestorac, S. Latré, J. M. Márquez-Barja, and M. Camelo, "Integrating AI orchestration and lifecycle management in 6G networks: A pipeline approach," in *Proc. IEEE Latin-American Conf. Commun. (LATINCOM)*, Nov. 2024, pp. 1–7.
[136] W. A. Ali, M. P. Fanti, M. Roccotelli, and L. Ranieri, "A review of digital twin technology for electric and autonomous vehicles," *Appl. Sci.*, vol. 13, no. 10, p. 5871, May 2023.
[137] A. Virdis and M. Kirsche, *Recent Advances in Network Simulation: The OMNET++ Environment and Its Ecosystem*. Cham, Switzerland: Springer, 2019.
[138] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G—An OMNeT++ library for end-to-end performance evaluation of 5G networks," *IEEE Access*, vol. 8, pp. 181176–181191, 2020.
[139] A. Kusari et al., "Enhancing SUMO simulator for simulation based testing and validation of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2022, pp. 829–835.
[140] W. A. Ali, A. M. Mangini, J. Júlvez, C. Mahulea, and M. P. Fanti, "Toward enhancing security in intelligent transportation: A simulation-based approach," *IFAC-PapersOnLine*, vol. 58, no. 4, pp. 156–161, 2024.
[141] *OpenStreetMap*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.openstreetmap.org
[142] *OpenCelliD France Cell Tower Dataset*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.opencellid.org/downloads
[143] Z. Iskander, S. S. Mohammed, E. K. Ines, and H. Shajjad, "Multi-objective route optimization via dueling DQN for teleoperated driving," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2025, pp. 1–16.
[144] F. Paparella, G. Olivieri, G. Volpe, A. M. Mangini, and M. P. Fanti, "A deep reinforcement learning approach for route planning of autonomous Vehicles*," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2024, pp. 2047–2052.
[145] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[146] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[147] A. Raghuramu et al., "Network digital twins: A threat analysis," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 733–739.
[148] A. San, M. Babaei, R. Fuladi, and U. Gulen, "Secured and privacy preserving data management system for 6G network digital twin," in *Proc. Intell. Comput. Computing Conf.*, 2025, pp. 603–621.
[149] J.-S. Sottet, A. Zaki–Hindi, I. Turcanu, and S. Faye, "A data space for 6G network digital twins: Challenges and opportunities," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Sep. 2025, p. 1.
[150] *The Internet of Everything How More Relevant and Valuable Connections Will Change the World*. Accessed: Dec. 12, 2025. [Online]. Available: https://www.cisco.com/c/dam/global/enmy/assets/ciscoinnovate/pdfs
[151] G. T. Doran, "There's a SMART way to write management's goals and objectives," *Manage. Rev.*, vol. 70, no. 11, pp. 35–36, 1981.
[152] (Sep. 2021). *FESTA Handbook*. [Online]. Available: https://www.connectedautomateddriving.eu/wp-content/uploads/2024/07/FESTA-Handbook-Version-8-FINAL-Version-20-09.pdf