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

D2.2

Basic models (initial)

Document Information	
Grant Agreement N°	101136314
Authors and institutions	Ayat Zaki-Hindi (LIST), Jean-Sébastien Sottet (LIST), Sébastien Faye (LIST), Ion Turcanu (LIST), Gaetano Volpe (Poliba), Wasim Ali (Poliba), Miguel Camelo (IMEC), Burkhard Hensel (TUD), Luis Santos (UBI), Inès El Korbi (UBOU)
Date	24 December 2024
Related WP	WP2 | Network Digital Twin Modelling
Dissemination level	PU | Public, fully open

Co-funded by the European Union

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

Document change history

Version	Date	Author	Description
V0.1	01/10/2024	Ayat Zaki Hindi (LIST)	First document with a table of content.
V0.2	15/10/2024	Ayat Zaki Hindi (LIST), Jean-Sébastien Sottet (LIST), Sébastien Faye (LIST), Ion Turcanu (LIST)	Initial content.
V0.3	03/11/2024	Wasim Ali (Poliba)	Added Section 4 content.
V0.4	26/11/2024	Burkhard Hensel (TUD), Luis Santos (UBI), Gaetano Volpe (Poliba), Inès El Korbi (UBOU)	Finalized sections content.
V0.5	28/11/2024	Miguel Camelo (IMEC)	Proposition for additional network domains.
V0.6	29/11/2024	All contributors	Content harmonization and completion.
V0.7	04/12/2024	Ayat Zaki Hindi (LIST)	Document ready for peer review
V0.8	16/12/2024	All contributors	Pre-final version after peer review ; document ready for second round of reviews
V1	24/12/2024	Ayat Zaki Hindi (LIST), Sébastien Faye (LIST)	Final submitted version

Quality control

	Name	Organisation	Date
Editor:	Ayat Zaki Hindi	LIST	24/12/2024
Peer review 1:	Luis Santos	UBI	11/12/2024
Peer review 2:	Sidi Mohammed Senouci	UBOU	11/12/2024
Peer review 3:	Georgy Myagkov	VIAVI	11/12/2024
Peer review 4:	Paola Soto Arenas	IMEC	19/12/2024
Authorised and submitted by (Project Coordinator):	Sébastien Faye	LIST	24/12/2024

Co-funded by the European Union

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

Legal disclaimer

Executive Summary

As the telecommunications industry progresses toward 6G, Network Digital Twins (NDTs) are emerging as a transformative tool for real-time monitoring, automation, and performance optimization in next-generation networks. This deliverable contributes to ongoing research within the framework of the 6G-TWIN project by establishing foundational definitions for NDT data models, specifically basic and functional models.

This document, **D2.2 - "Basic Models"**, specifically focuses on defining and developing basic models, which are essential for representing the physical network and its elements. This deliverable, produced in Month 12, will be complemented by D2.3, to be delivered in Month 24, which will focus on developing functional models. Additionally, a final version of D2.2 will be delivered at Month 36, incorporating refinements and updates based on project advancements and feedback. The document is structured methodically to build a strong basis for understanding and implementing basic models.

Chapter 2 provides a detailed overview of the NDT architecture and the role of data models within the NDT layer, distinguishing between basic and functional models and their respective evaluation indicators. **Chapter 3** delves into the representation of basic models, introducing hierarchical approaches for modelling network elements (NEs) and network systems. **Chapter 4** examines state-of-the-art methods for deriving network models from traffic data, focusing on both probabilistic and graph-based approaches. **Chapter 5** reviews graph-based tools for NDTs, emphasizing their applicability to basic models, and proposes GreyCat as the preferred tool for the 6G-TWIN framework. **Chapter 6** highlights the integration of basic models with the broader 6G-TWIN architecture, emphasizing their role in data collection and simulation framework. The document concludes in **Chapter 7** by summarizing key findings, addressing open challenges, and outlining future work.

Defining data models for 6G NDT

This document establishes formal definitions and boundaries between basic and functional models, which underpin the NDT framework. It identifies the main network domains to be modelled, including the Radio Access Network (RAN), Transport Network (TN), Core Network (CN), computing domains, and the User Equipment (UE) domain. Additionally, it outlines key evaluation indicators for basic and functional models, grounded in the ITU-T Y.3091 recommendations, to ensure their alignment with our studies.

Representation of basic models

This document investigates the structure and representation of basic models, highlighting their hierarchical organization: (1) basic models of network elements (NEs) and (2) basic models of network systems. Network elements are categorized based on the network domains, with detailed schema examples provided as foundational templates. These NE models are then synthesized into broader network basic models using graph-based techniques and standards-driven insights.

Deriving network models from traffic data

The deliverable also addresses the challenges of creating basic models, including the complexities of dynamic networks. For that, this study extends the exploration of graph-based

Co-funded by the European Union

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

6GSNS

modelling by presenting state-of-the-art methodologies for deriving network models from traffic data. It examines probabilistic and graph-based approaches, concluding to a proposed framework for implementing basic models within the 6G-TWIN architecture, using Graph Neural Networks (GNN).

Tools for graph-based NDT models

In addition to setting up theoretical foundations for basic models, this deliverable also reviews tools that support graph-based NDT models, offering a review of state-of-the-art platforms used across digital twin communities. By evaluating the features, strengths, and limitations of these platforms, GreyCat tool is identified as the most suitable tool for the 6G-TWIN framework, given its ability to handle complex data structures and support graph-based modelling.

Integrating basic models within the 6G-TWIN framework

Finally, this document establishes the connections between basic models and other project activities, particularly their integration into the data collection framework developed in WP1 and their role in supporting simulation tools in WP3. This integration ensures a cohesive alignment between the basic models and the overarching 6G-TWIN architecture.

Co-funded by the European Union

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

Abbreviations and acronyms

Abbreviations and acronyms	
AAS	Asset Administration Shell
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
API	Application Programming Interface
ASTGCN	Attribute-Augmented Spatiotemporal Graph Convolutional Network
CE	Computing Element
CN	Core Network
CNS	Complex Networked Systems
CPU	Central Processing Unit
CSI	Channel State Information
CU	Centralized Unit
Dx.x	Deliverable number “x.x”
DAGG	Data Adaptive Graph Generation
DCRNN	Diffusion Convolutional Recurrent Neural Networks
DT	Digital Twin
DT-C	Digital Twin Connector
DTC	Digital Twin Channel
DTDL	Digital Twin Definition Language
DU	Distributed Unit
ETSI	European Telecommunications Standards Institute
GA	Grant Agreement
GMAN	Graph Multi-Attention Network
GNN	Graph Neural Network
GUI	Graphical User Interface
GWN	Graph WaveNet
HDF	Hierarchical Data Formats
ITU-R	International Telecommunications Union – Radiocommunication Sector
ITU-T	International Telecommunications Union – Telecommunications Sector
KPI	Key Performance Indicator
MAC	Medium Access Control
MANO	Management and Orchestration
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NAPL	Node Adaptive Parameter Learning
NDT	Network Digital Twin
NE	Network Element
NF	Network Function
NFV	Network Function Virtualization
NGSI-LD	Next Generation Service Interfaces – Linked Data
NS	Network Service
NTN	Non-Terrestrial Network
OMG	Object Management Group
RU	Radio Unit
SDK	Software Development Kit
SINR	Signal-to-Noise and Interference Ratio
SMF	Session Management Function
SS	Synchronization Signal
STGCN	Spatio-Temporal Graph Convolutional Networks

Co-funded by the European Union

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

6GSNS

TN	Transport Network
UDM	Unified Data Models
UDR	Unified Data Repository
UE	User Equipment
UML	Unified Modelling Language
VAB	Virtual Automation Bus
VNF	Virtual Network Function
VPN	Virtual Private Network
WP	Work Package

Co-funded by the European Union

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

6GSNS

Table of Contents

Co-funded by the European Union

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

Co-funded by the European Union

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

Table of figures

Table of tables

Co-funded by the European Union

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

1. Introduction

The rapid digitization of industries necessitates advancements in network technologies, particularly as we transition towards 6G systems. The 6G-TWIN project addresses this need by proposing an AI-native reference architecture that incorporates Network Digital Twins (NDTs).

We first outline the objectives of 6G-TWIN, emphasizing its core mission to develop a sophisticated network framework, before specifically focusing on the key targets of this deliverable, presenting its structure and the contribution of the project's partners.

1.1. Aims and objectives

1.1.1. 6G-TWIN objectives

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

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

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

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

Co-funded by the European Union

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

1.1.2. Deliverable objectives

The objective of this deliverable is to define and investigate basic models, allowing the virtual representation of a network topology and its infrastructure elements. These models will establish a robust framework for the development of advanced functional models that are related to the application. The deliverable focuses on the following technical activities:

1. Reviewing state-of-the-art methods and standardisation efforts on basic models for NDTs, particularly those aligned with the EU 6G vision.
2. Constructing a multi-dimensional graph structure as a basic model, representing the physical network with essential elements such as nodes, links, and their attributes under space-time constraints.
3. Developing techniques to extract basic models from traffic data, enabling the initial population of the NDT with reliable and representative data.
4. Exploring low-TRL tools to manage and utilise the graph-based NDT, ensuring its integration with components in WP1 (6G architecture design) and WP3 (Open and secured management and simulation framework).

This deliverable builds on the harmonised data from Task 2.1, supports the refinement of basic models in Task 1.4, and contributes to the development of the simulation framework in WP3.

1.2. Relation to other activities in the project

As represented in Figure 1 below, this deliverable builds on the harmonised dataset and requirements provided by T2.1: NDT Data Requirements, Governance, and Harmonisation. The standardised data space and harmonisation mechanisms developed in T2.1 simplify the complexities of raw data, enabling the creation of consistent basic models for NDTs. These inputs ensure that the virtual representation aligns with privacy, storage, and access requirements while supporting integration across the project.

The outputs of this task - low-TRL prototypes - feed directly into T2.3: Network Dynamics Modelling (Functional Models) and are essential for the open simulation framework developed in WP3. These prototypes provide a foundation for functional models to predict or emulate network behaviour and for discrete-event simulations to test and optimise 6G systems under various scenarios.

Co-funded by the European Union

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

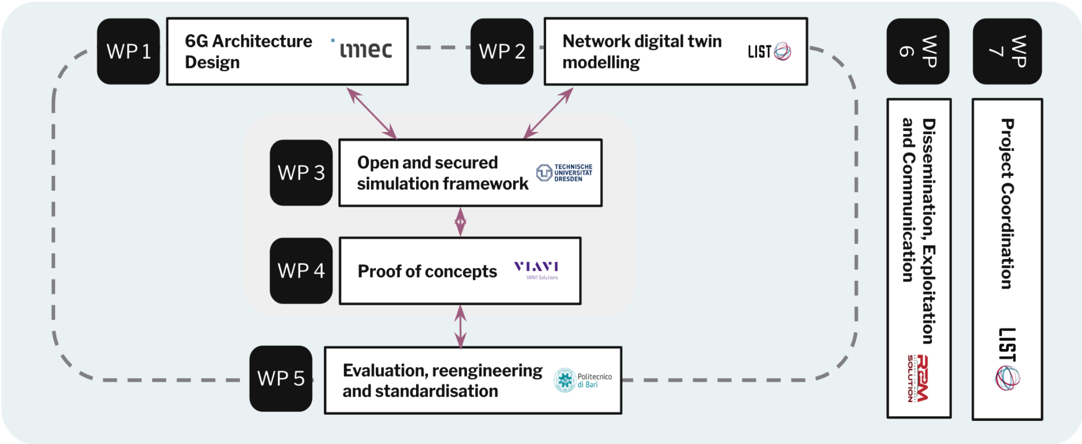

Figure 1 6G-TWIN PERT chart.

1.3. Report structure

This document investigates the NDT basic models, which are one of the two components of the unified data models, i.e., basic and functional models, within the NDT layer of the architecture outlined in Deliverable 1.1 (D1.1 - Architecture and Technical Foundations) of the 6G-TWIN project. The report structure is organized as follows.

- **Chapter 1** introduces the 6G-TWIN project by highlighting its overarching goals, including building a graph-based NDT framework within the NDT architecture. Additionally, the introduction connects the report to other activities within the project, particularly its reliance on harmonized data from T2.1 and its contributions to tasks in WP1 and WP3. This chapter also acknowledges the contributions of various partners, emphasizing the collaborative nature of the initiative.
- **Chapter 2** provides an overview of the NDT conceptual architecture, focusing on the placement of data models within the NDT layer. It defines the network domains to be modelled by the NDT, covering four important domains of network infrastructure (RAN, TN, CN, and computing domains) and the user equipment (UE) domain. Additionally, this chapter offers formal definitions for basic and functional models, delineating their boundaries to clarify their interrelationship. It also introduces key evaluation indicators for these data models based on the ITU-T Y.3091 recommendations.
- **Chapter 3** explores basic models in depth, emphasizing their hierarchical structure: (1) basic models of network elements (NEs) and (2) basic models of network systems. NEs are categorized by the network domains identified in Chapter 2, and schema examples for these NEs are presented as basic models. These NE models are then utilized to construct the broader network basic model, employing graph-based approaches and drawing insights from existing standards.
- **Chapter 4** extends the graph-based modelling concept, presenting a comprehensive review of state-of-the-art methodologies for deriving network models from traffic data. It distinguishes between probabilistic and graph-based methods, reaching an initial proposal for implementing basic models within the 6G-TWIN framework using graph-based techniques.

- **Chapter 5** complements the previous chapters by focusing on tools for graph-based NDT models. It reviews state-of-the-art platforms employed by different digital twin communities, evaluating their advantages and limitations. Building on this analysis, the chapter proposes GreyCat as a tool to be integrated into the 6G-TWIN framework.
- **Chapter 6** establishes connections between this deliverable and other 6G-TWIN activities. It emphasizes the relationships between basic models and the data collection framework developed in WP1, as well as the simulation framework outlined in WP3.
- **Chapter 7** concludes the report by summarizing the findings and contributions, reinforcing the significance of the proposed approaches for advancing the 6G-TWIN project.

1.4. Contribution of partners

The following table present the contributions from all the partners into the deliverable.

Table 1 Partners contributions to the D2.2 deliverable.

Partner	Section(s)	Contributions
LIST	**1**, **2**, **3**, **5**, **6**, 7	Lead editor on the document; responsible for the content in Section 1 (Introduction), Sections 2 and 3 (Basic models definitions and representation), Section 5 (Tools for graph-based NDT), and Section 6 (Conclusions).
IMEC	2, 3, **3.1.4**	IMEC supported sections 2 and 3 by adding transport network domain and computing domain to the network modelling, and proposed data schemas for computing domain in section 3.1.4. Document review.
POLIBA	**4**	POLIBA lead Section 4 on deriving network models from traffic data and describe the connection between basic models and network models.
UBOU	**2.2.2**	UBOU contributed on the functional model's definition.
TUD	**6.2**	TUD contributed to the relation of basic models and simulation models, described in section 6.2.
UBI	**6.1**	UBI contributed to the relation of basic models with the data collection framework and the overall architecture, described in section 6.1. Document review.
VIAVI	**-**	Document review.

***Bold** numbers represent section technical leaders*

1.5. Deviations from the GA

Not applicable.

Co-funded by the European Union

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

6GSNS

2. Defining Data Models for 6G NDT

Data models are one of the founding components of the NDT layer in the NDT architecture as presented in D1.1 of the 6G-TWIN project. These models rely on data collected from the physical network and other sources that affect the performance of the network. A preliminary data taxonomy based on existing standards has been developed in a 6G-TWIN deliverable, D2.1 (Data governance, privacy, and harmonization).

We focus in this chapter of this deliverable on defining basic and functional models as described in D1.1, which are aligned with the ITU-T recommendations on NDTs [1].

2.1. Overview of the NDT architecture

In alignment with the reference architecture defined by ITU-T [1] and 6G-TWIN D1.1, we present here a conceptual architecture structured around three layers: the physical network layer, the NDT (Network Digital Twin) layer, and the application layer. This layered structure, depicted in Figure 2, highlights the components within each layer and their interactions through an NDT management entity, allowing the seamless orchestration of data and models depending on the application needs. Here, we review the roles of each layer and elaborate on the specific building blocks of the physical network layer and the NDT layer, as relevant to this study.

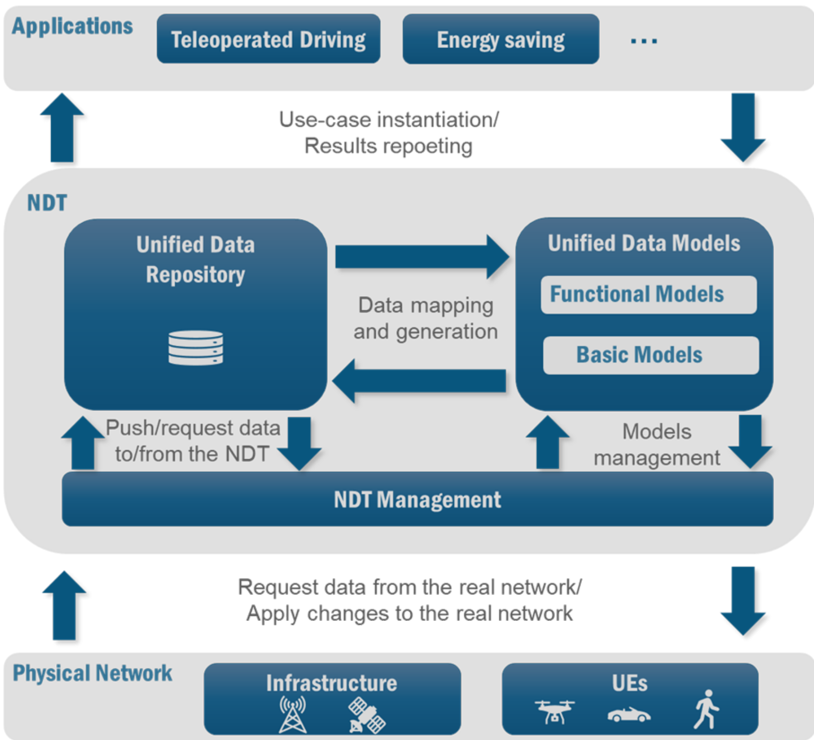

Figure 2 Conceptual NDT architecture.

Co-funded by the European Union

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

2.1.1. Physical Network Layer

Referred to as the infrastructure layer within the broader 6G end-to-end architecture as suggested by 5G PPP [2] , the physical network layer comprises both physical and virtualized resources. These include essential components like the Radio Access Network (RAN), Core Network (CN), edge and cloud computing infrastructure, and Transport Network (TN), all of which deliver services to User Equipment (UE).

To accurately reflect the **6G network infrastructure** within the NDT framework, we segment this layer into **four main domains** for twinning: **RAN, TN, CN, and edge/cloud computing**. The RAN serves as the connectivity bridge between UEs and the CN, represented in 5G by the Next Generation Node B (gNB). The CN includes Virtual Network Functions (VNFs) tasked with data collection, management, service continuity, and network access, supporting operator services, internet access, and third-party services. The primary function of TN is to provide reliable and efficient data transport across the different parts of the mobile network infrastructure such between the RAN and the CN, or inside the RAN itself connecting Radio Units (RU), Distributed Units (DU), and Centralized Units (CU) in disaggregated deployments. Finally, computing infrastructure includes edge and cloud servers used to host (virtual) network functions (NF) and network services (NS), e.g., a cloud-native RAN or CN, and vertical applications and services, e.g., a predictive maintenance model for smart industry running at the edge.

RAN, TN, CN domains are further organized into Network Elements (NEs), and computing domains into Computing Elements (CE), which may refer to intelligence, information, hardware, or software aspects. Categorizing each NE and CE enables a structured approach to managing these diverse components effectively.

Another important domain to be twinned by the NDT layer which does not necessarily belong to the network infrastructure is the **UE domain**. UEs refer to connected devices such as smartphones, UAVs, Vehicles, and other terminals that request services, including internet access and edge/cloud computing. It is essential to model UEs within the NDT along with the NEs in order to obtain a complete knowledge of the system.

2.1.2. NDT Layer

The NDT layer is designed to capture and mirror the real-time status and operational conditions of the physical network components. This layer not only creates and manages Digital Twin (DT) instances but also handles data management, aggregation and harmonization, inter-layer communication, and the lifecycle of NDT representations. The NDT layer is structured around the following key building blocks:

Unified Data Repository (UDR)

The UDR functions as a centralized storage and access point for historical and real-time data, supporting accurate and efficient NDT modelling. It aggregates data from varied sources such as network infrastructure, sensors, and contextual information, making data harmonization, collection, and storage essential.

Co-funded by the European Union

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

A taxonomy for data to be handled by the UDR is proposed in Deliverable D2.1. Furthermore, Deliverable D1.2 dives into the details of the data collection framework and data harmonization aspects of the UDR, which is referred to as the harmonization data layer.

Unified Data Models (UDM)

Within this layer, data models are categorized into two main types: (1) **basic models**, which describe the network's physical state in real-time—covering configuration, environment, and topology—to help validate and simulate control adjustments, and (2) **functional models**, which utilize insights from basic models to optimize and forecast network performance. Functional models commonly incorporate AI/ML algorithms aimed at specific goals, such as optimization or anomaly detection. The present deliverable focuses on basic models, while Deliverable D.2.3 will be dedicated to functional models.

NDT Management

The NDT management component ensures the lifecycle management of NDT instances, aligning them with application-layer needs. It coordinates UDR and UDM interactions, mapping data effectively and ensuring that models accurately reflect the current network state. This component also oversees data collection processes and applies relevant algorithms to the physical network.

2.1.3. Application Layer

The application layer directs the NDT management entity to create NDT instances based on specified Key Performance Indicators (KPIs). These KPIs guide the selection of basic and functional models that align with the application's needs. Additionally, this layer receives feedback from the NDT management entity, informing users of actions taken or intended for implementation within the physical network.

The project aims to achieve ambitious performance KPIs aligned with 6G use cases, addressing mobility and energy-efficiency challenges through two complementary use cases. These will be implemented in low-TRL laboratory demonstrators and will be further developed in WP4 of the project.

In the following section, we define data models within the UDM, i.e., basic and functional models.

2.2. Data models definition

In telecommunications and mobile networks, various models have been developed to support decision-making processes, starting with foundational approaches like Markov processes, statistical models, and queuing theory, and progressing to more recent data-driven models leveraging AI. However, **NDT modelling, an emerging approach that incorporates interactions from/to reality for real-time or predictive decision-making, remains relatively underexplored, revealing a clear gap in the literature**. In contrast, other fields, such as manufacturing [3], have developed more advanced modelling frameworks. These

Co-funded by the European Union

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

frameworks typically view physical entities in a **hierarchical structure** [4], commonly organized into unit level, system level, and System of systems (SoS) level. For example, in the DT modelling of a city, a "unit" could be a street or a building, while a "system" might represent an entire campus, and the "SoS" would represent the entire city. Applying this concept to 6G networks, we can similarly view Network Elements (NEs) as analogous to unit-level entities, while the entire network could be represented at the system level. For now, we do not differentiate between a "system" and a "system of systems" within the network context, as this distinction primarily reflects varying levels of detail, and it is sufficient at present to treat the network as a cohesive system.

In the context of 6G-TWIN, one interpretation regarding the classification of models is that behavioural models, specifically algorithms that help emulate basic network behaviour, should be categorized as basic models. This classification is not commonly addressed in the existing literature.

Another interpretation within the 6G-TWIN project is regarding **the classification of behavioural models as part of basic models**, specifically algorithms that help emulate basic network behaviour. This classification is not commonly addressed in existing literature.

For instance, a physical layer resource allocation algorithm can be viewed as a basic model since it outlines the essential behavioural dynamics of the network. This algorithm is responsible for managing how resources, such as bandwidth and power, are allocated to different users and services, thereby directly simulating the normal network's performance.

On the other hand, functional models are designed to enhance or optimize existing algorithms for specific purposes or scenarios. Taking the aforementioned resource allocation algorithm, a functional model can modify it to prioritize certain data flows over others. Therefore, we propose the following formal definitions of a basic and functional models from the 6G-TWIN perspective.

2.2.1. Basic models

Basic models provide real-time descriptions of the network's physical state, including configuration, environment, and topology, helping verify and emulate control changes before implementation. We distinguish between two categories of basic models.

---

*A **basic model of a network element** is the collection of data describing its properties, configurations, and operational status, along with any associated algorithms or protocols used to emulate its dynamics and evolution with time.*

The properties, configurations, and operational status refer to the attributes and measurements, as described in Deliverable D2.1. Each NE is associated with a set of data entries which are updated based on the UDR entries. Depending on the nature of the NE, it could be associated with a set of protocols or algorithms, that are executed in the physical

Co-funded by the European Union

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

network. These protocols in addition to the data entries help building the NE model and simulate its behaviour.

*A **basic model of a network** is the aggregation of basic models of network elements, including their physical and logical relationships and the interactions that occur between them.*

This second type of models considers the relationships among NEs in order to represent the network. This includes topological information and interfaces, which will be further investigated in Section 3.2.2 Static graph representation.

2.2.2. Functional models

Functional models leverage insights and data from basic models to optimize and predict the behaviour of network elements.

Depending on the scenario or use case being evaluated, functional models may require the utilization of lower-level network functionalities tied to the physical infrastructure, henceforth referred to as basic models. A functional model of a network leverages these basic models, applying advanced processing techniques—often through AI/ML algorithms—across diverse operational scenarios. These models are tailored to achieve specific objectives, such as performance optimization, anomaly detection, or predictive maintenance.

In summary, basic models pertain to the physical layer of the network, focusing on foundational elements, while functional models operate at the application layer, addressing higher-level functionalities and objectives.

***Functional models** refer to the set of models that, while making an abstraction of the real world, can predict the behaviour of the network within the digital world (the twin), then make the decision applicable within the real world*

Functional models are further developed in Deliverable D2.3 with a comprehensive model classification and implementation, in relation to the project defined use-cases.

2.3. Data models evaluation indicators

ITU-T's classification of NDT capability levels [5] serves as a reference for measuring the maturity of NDT implementations. These levels range from Level 1, where the NDT performs limited simulation tasks, to Level 5, where the NDT operates autonomously, capable of self-configuration and adjusting the physical network without human intervention.

Co-funded by the European Union

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

2.3.1. Basic models integrity level

The complexity of the data models and management strategies employed by the NDT layer are directly influenced by the chosen level of NDT capability. Therefore, basic models' integrity can be understood across four progressive levels, each enhancing the depth and scope of modelling capabilities.

- **Level 1** focuses on representing specific attributes of physical network equipment, including geometric details, physical specifications, and basic provisioning data.
- **Level 2** extends this capability to capture all details of physical network devices, covering aspects such as port status, link performance metrics, and protocol states.
- **Level 3** enables modelling of certain network topologies, such as those found in data centres, RAN, and campus networks. This level also begins to include limited logical relationships within the network.
- **Level 4** offers the most comprehensive modelling capability, including all types of network topologies and detailed representations of physical and logical structures, as well as advanced features like network slicing, cross-domain interactions, and end-to-end service mappings.

2.3.2. Functional models integrity level

A similar classification of the above integrity levels of basic models is extended to functional ones. These levels outline the progressive depth and capability of functional model implementation:

- **Level 1** supports partial implementation of low-risk functional models, such as network capacity planning and site planning.
- **Level 2** fully supports implementation of all low-risk functional models, providing reliable functionality for predictable and well-defined scenarios.
- **Level 3** supports partial implementation of high-risk functional models, including tasks like network resource allocation in complex environments.
- **Level 4** provides full support for all functional models, enabling advanced functionality and adaptability across diverse operational conditions.
- **Level 5** extends full support for all functional models and introduces self-optimization capabilities, leveraging AI/ML algorithms for real-time decision-making, anomaly detection, and adaptive network behaviour.

Co-funded by the European Union

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

3. Representing Basic Models

As described in Section 2.2.1 Basic models of this deliverable, basic models have two levels of hierarchy, modelling the NEs or the network system [6]. However, there is a substantial difference in modelling each one of these levels of hierarchy, which is addressed in this chapter. The classification of NEs is organised by network domain in this deliverable.

3.1. Basic Models of network elements (NEs)

As described in Section 2.1.1, the physical network infrastructure spans over four main domains: **RAN**, **TN**, **CN**, and **edge/cloud computing**, in addition to the "non-infrastructure" **UE domain** which remains essential for a complete NDT modelling.

For RAN, TN, and CN, the 3GPP standards have structured Network Elements (NEs) into defined categories, known as Information Object Classes (IOCs) [7]. Each class is linked to specific attributes that describe various management aspects of network resources, establishing a framework for exchanging information through management interfaces in a technology-agnostic way. In addition to these attributes, NEs may also be related to measurement data that reflects the network performance under the considered attributes.

We note that in Deliverable D2.1 we proposed a data taxonomy for RAN, CN, and UE domains, where we related each IOC to its attributes and measurements. We detail in this deliverable the basic models based on that data taxonomy and extend the study to include TN and computing domains. These schemas (metamodels) can be represented in terms of smart data models [8], as described in Deliverable D2.1. However, a more detailed analysis of the tools to be used to represent basic models is further investigated in Chapter 5 of this deliverable.

3.1.1. Basic models for RAN NEs

The RAN infrastructure is mainly characterised by the gNB in cellular networks. Future implementations and deployments include the 7.2 split in O-RAN, which decomposes the tasks of the gNB into three distinct components: the Radio Unit (RU), Distributed Unit (DU), and Central Unit (CU). The RU, which serves as the primary interface between the UE and the network, can be broken down further into sub-components like Beam, which represents the antenna's beam properties, and BWP (Bandwidth Part), which allocates and manages the gNB's spectral resources. An example of the *CU*, *DU*, and *Beam* metamodels is given in Figure 3.

This NE decomposition is derived from 3GPP TS 28.541 [9], where IOCs are related to their attributes, while the related measurements are mapped to the corresponding IOCs from 3GPP TS 28.552 [10].

Co-funded by the European Union

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

Class ID: NRCellCU

Parent: RAN

Attributes: cellLocalId; pLMNInfoList;

Measurements: RRC connection number; RRC connection establishment and Re-establishment; UE-associated logical NG-connection; DRB (Data Radio Bearer) related measurements; PDU Session Management; Measurements related to MRO (HO); 5QI x QoS Flow Duration

Class ID: NRCellDU

Parent: RAN

Attributes: cellLocalId; operationalState; administrativeState; cellState; pLMNInfoList; nRPCI; nRTAC; arfcnDL; arfcnUL; arfcnSUL; bSChannelBwDL; ssbFrequency; ssbPeriodicity; ssbSubCarrierSpacing; ssbOffset; ssbDuration; bSChannelBwUL; bSChannelBwSUL;

Measurements: Packet Delay; Radio resource utilization; UE throughput; Number of Active UEs; CQI-related measurements; Transmit power utilization measurements; Received Random Access Preambles; PHR (power headroom) Measurement

Class ID: Beam

Parent: RAN

Attributes: beamIndex; beamType; beamAzimuth; beamTilt; beamHorizWidth; beamVertWidth.

Measurements: Intra-NRCell SSB Beam switch Measurement; RSRP Measurement; SSB beam related measurement.

Figure 3 Example of RAN NE metamodels.

Cellular networks account for the biggest part in current 4G and 5G networks. However, future networks will rely on the coexistence of diverse heterogeneous technologies such as cellular networks, non-terrestrial networks (NTN), Reconfigurable Intelligent Surfaces (RIS), and Wi-Fi, in order to satisfy emerging applications requirements. We note that in this deliverable, we only give an example of NEs that are specified in the standards, so other NEs from other technologies should have a similar structure as the one we present here.

Another aspect of RAN is the transmission channel which does not necessarily fall into the “typical” NE definition, yet it is considered as a separate entity within the network.

Basic models of transmission channel

Due to the intricate nature of transmission channels, and especially wireless channels, basic models should be built differently for them. Here we consider channel as an element of the network which can be plugged into the overall network basic model as will be detailed in the following section (Section 0).

We focus in this study on wireless channel model as it represents most of the RAN technology in 5G and 6G networks, keeping in mind that other wired channels are possible to be twinned such as optical fibres and ethernet.

Because of the importance of channel in wireless networks, a Digital Twin Channel (DTC) concept has emerged, aiming to build a real-time, online replica of the channel in any given scenario. The work in [11] defines five levels of channel twins depending on the methods and outcomes of the models.

Basically, level 1 refers to the analytical channel models obtained through mathematical formulation, e.g. Rice and Rayleigh models. This level merely provides insights on the system capacity and performance but cannot reflect scenario-specific analysis.

Co-funded by the European Union

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

Level 2 profits from measurements to obtain the statistical characteristics and distributions of the channel in specific scenarios. Examples of these models are provided by the standardized 5G 3GPP 38.901 [12] and ITU-R M.2412 [13] models for several typical scenarios for micro and macro cellular networks deployment.

Level 3 relies on simulation and emulation tools such as raytracing, provided the 3D mapping of the environment. This can be combined with AI techniques to further predict the channel conditions. However, these techniques can only be executed offline and cannot capture real-time changes on the physical system.

In level 4, the DTC makes use of the data used by level 2 models in addition to other available environment and sensing data, which is then used by AI techniques to predict channel conditions in real time. Finally, the level 5 DTC adds the self-learning capability to level 4 models by using historical data and information available by multiple nodes in a centralized and cooperative structure. New data types can be used here such as historical data, network status information, and user preferences.

3.1.2. Basic models for CN NEs

In a similar modular approach presented for RAN NEs, the CN consists of various Virtual Network Functions (VNFs), such as the Access and Mobility Management Function (AMF) and Session Management Function (SMF). Each VNF operates independently and is characterized by its specific role in supporting tasks like mobility management, session management, and other critical network services. This modularity enhances network flexibility, scalability, and ease of management, aligning with the objectives of softwarisation and automation, essential for evolving 6G networks.

In Figure 4 Example of CN NE metamodels.Figure 4, we present an example of the CN NE metamodels, including their attributes and measurements. Similar to RAN NEs, the attributes and measurements of CN IOCs are derived from 3GPP standards [9] [10].

Co-funded by the European Union

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

Class ID: AMFFunction

Parent: CN

Attributes: pLMNIdList; aMFIdentifier; sBIFQDN; sNSSAIList; managedNFProfile; commModelList;

Measurements: Registered subscribers' measurement; Registration procedure related measurements; Mobility related measurements; UE Configuration Update procedure related measurements; Authentication procedure related measurements; Measurements related to registration via trusted/untrusted non-3GPP access; Measurements related to Service Requests via Trusted and Untrusted non-3GPP Access.

Class ID: SMFFunction

Parent: CN

Attributes: pLMNIdList; nRTAClist; sBIFQDN; sNSSAIList; managedNFProfile; commModelList.

Measurements: Session management; QoS flow monitoring; Performance measurement for N4 interface.

Class ID: UPFFunction

Parent: CN

Attributes: pLMNIdList; nRTAClist; sNSSAIList; managedNFProfile; supportedBMOList;

Measurements: Interfaces related measurements; GPRS Tunnelling Protocol (GTP) packets delay in UPF; One way packet delay between NG-RAN and PSA UPF; Round-trip packet delay between PSA UPF and NG-RAN; One way packet delay between PSA UPF and UE; QoS flow related measurements.

Figure 4 Example of CN NE metamodels.

3.1.3. Basic models for TN NEs

The Transport Network (TN) is a crucial part of telecommunication systems, providing high-capacity connectivity between different network domains, such as RAN, CN, and data network. It is responsible for the efficient, reliable, and secure transmission of data, voice, and video services across long distances, enabling seamless communication between users, data centers, and cloud services. The role of the TN in 5G systems is illustrated in Figure 5 [14], connecting RAN and CN.

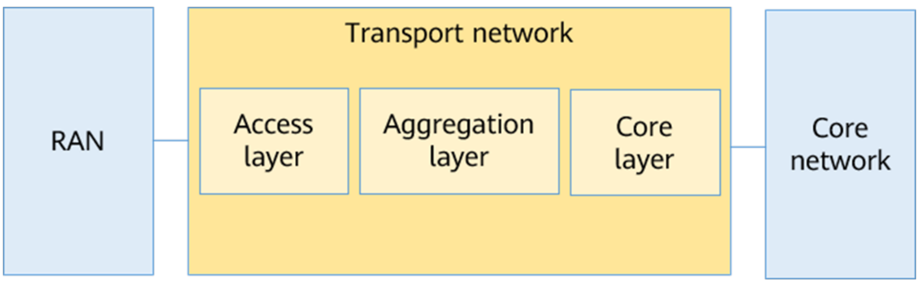

Figure 5 Transport network in 5G linking RAN and CN. [14].

Modern TNs use advanced technologies like optical fibre, packet switching, and synchronisation mechanisms to support the diverse and demanding requirements of today's applications, including 5G, IoT, and cloud computing. They form the backbone of telecommunications, ensuring the smooth operation of network services by managing data flow between various nodes and devices.

Co-funded by the European Union

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

TNs must balance speed, scalability, and reliability to meet the growing demand for bandwidth and real-time services. As such, they are evolving to include features like network programmability, automation, and multi-layered architectures to support the rapid expansion of digital services and future network innovations. This is why it is important to include the TN domain in the NDT modelling.

The 3GPP standards define the main TN IOCs in [15]. We illustrate in Figure 6 these IOCs following the same logic as for RAN and CN.

Class ID: TransportNetworkInterface

Parent: TN

Attributes: transportNetworkInterfaceId; userLabel; transportNetworkType.

Measurements: -

Class ID: ATMChannelTerminationPoint

Parent: TN

Attributes: aTMChannelTerminationPointid; usageChannel; virtualPathId; virtualChannelId; physicalPortId; physicalInterfaceType; serviceCategoryIn; ServiceCategoryEg; usedAAL; peakCellRateIn; peakCellRateEg; sustainableCellRateIn; sustainableCellRateEg; maximumBurstSizeIn; maximumBurstSizeEg; minimumDesiredCellRateIn; minimumDesiredCellRateEg; minimumCellRateIn; minimumCellRateEg; aTMChannelTerminationPoint – ATMPathTerminationPoint; aTMChannelTerminationPoint- IubLink

Measurements: -

Class ID: ATMPathTerminationPoint

Parent: TN

Attributes: aTMPathTerminationPointid; virtualPathId; physicalPortIdList; peakCellRateIn; peakCellRateEg; aTMPathTerminationPoint - ATMChannelTerminationPoint.

Measurements: -

Figure 6 Example of TN NE metamodels.

3.1.4. Basic models for edge/cloud computing CEs

As mentioned before in Section 2.1.1, the computing domain encompasses both edge and cloud computing, which are aligned with the Network Functions Virtualization Management and Orchestration (NFV-MANO) framework (further described in the Annex of Deliverable D1.2). This alignment ensures that the VNFs and their management are efficiently orchestrated across diverse computing environments. The NFV-MANO framework provides a standardized approach to manage and orchestrate these functions, facilitating seamless integration and operation within the NDT ecosystem.

In the context of edge and cloud computing, several elements can be twinned, each with its associated data. These elements include:

1. **(Virtual) Computing Elements (CE):** These are fundamental computation units in both edge and cloud environments such as Virtual Machines (VMs) and Containers. The data associated with these elements include their configurations, resource allocations (CPU, memory, storage), operational status, and performance metrics.
2. **Network Functions (NFs)**: These include both physical and virtual network functions such as routers, firewalls, and load balancers, which are specific to the computing

Co-funded by the European Union

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

domain. The associated data encompasses configuration settings, operational status, traffic statistics, and security policies.

3. **Network Services (NSs)**: An NS is a composition of network functions, which can also be both physical and virtual. An example of an NS is a Virtual Private Network (VPN) service, which typically includes VNFs such as the VPN gateway, the firewall, and the load balancer.
4. **Network:** Although elements beyond computing may be outside the scope of NDT for this domain, the interactions and relationships between these components, which collectively form the network, can also be considered. For example, an NS twin can include a relation between NFs in the chain in terms of bandwidth and latency.
5. **Storage Systems**: Both edge and cloud storage systems can be twinned. The data includes storage capacity, usage statistics, data replication status, and performance metrics.
6. **Orchestration and Management Systems**: These systems manage the deployment and operation of VNFs. The data includes orchestration policies, deployment configurations, and operational logs.

Notice that NFs and NSs may be described as a pure CE, e.g., number of CPU, memory, storage, bandwidth, or from a more high-level perspective containing properties associated with its function or domain, e.g., an AMF or a router. An example of a schema to represent CE are presented in Figure 7.

Class ID: Server

Parent: Data Center

Attributes: cpuCores; memorySize,storageCapacity; networkInterfaces; operatingSystem

Measurements: CPU Usage; Memory Usage; Disk I/O; Network Throughput; Uptime

Class ID: Container

Parent: Server

Attributes: containerID, imageName,allocatedCPU, allocatedMemory, ports

Measurements: CPU Usage; Memory Usage; Network I/O; Container Uptime; Restart Count

Class ID: SwitchVNF

Parent: NF

Attributes: switchID; switchType; portCount; vlanSupport; forwardingTable.

Measurements: Port Traffic; Packet Loss; Latency, Error Rate; Throughput

Figure 7 Schemas representing Computing Elements (CEs) in edge/cloud computing domains

3.1.5. Basic models for UE elements

While UEs are not part of the network infrastructure, they play a crucial role in scenario modelling and achieving application-level KPIs. Defining common properties for UEs is challenging due to their vendor-specific designs and diverse capabilities, such as sensors, mobile phones, autonomous vehicles. However, network data collected from UE feedback can be utilized to establish basic UE models within the NDT. A similar schema, as described previously for the network infrastructure domains, is presented for UEs in Figure 8, focusing

Co-funded by the European Union

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

exclusively on measurements received from the UE, and derived from 3GPP TS 38.215 [16]. Examples of measurements considered include Channel State Information (CSI) and localization information. UE attributes are excluded because the current NDT is designed to optimize network parameters rather than those of the UEs.

Class ID: Synchronization Signal (SS)

Parent: UE

Attributes: -

Measurements: SS reference signal received power and quality (SS-RSRP) and (SS-RSRQ); SS signal-to-noise and interference ratio (SS-SINR); SS reference signal received power per branch (SS-RSRPB); SS reference signal antenna relative phase (SS-RSARP).

Class ID: Channel State Information (CSI)

Parent: UE

Attributes: -

Measurements: CSI reference signal received power (CSI-RSRP); CSI reference signal received quality (CSI-RSRQ); CSI signal-to-noise and interference ratio (CSI-SINR).

Class ID: Global navigation satellite system (GNSS)

Parent: UE

Attributes: -

Measurements: UE GNSS Timing of Cell Frames for UE positioning for E-UTRA; UE GNSS code measurements; UE GNSS carrier phase measurements.

Figure 8 Example of UE element metamodels.

3.2. Basic models of networks

The backbone of the basic models for NEs is the taxonomy of network data defined in Deliverable D2.1. In addition to the proposed data taxonomy, these models also include all the functions that depict the standard behaviour of the network, such as the 3GPP protocol stacks and procedures [17], e.g., Medium Access Control (MAC) protocols.

However, basic models of the network require modelling the overall network topology and relationships among NEs. For that purpose, we focus in this section on describing basic models for networks as defined in Section 2.2.1 of this deliverable.

While NDT research is still in its early stages, we can draw valuable insights from more mature DT implementations in other domains, such as manufacturing, where standardized models, e.g., ISO [3], have been established. These serve as useful references for building comprehensive NDT basic models.

In the following, we review modelling approaches proposed by the DT literature, which motivates our proposed support for basic models.

3.2.1. Related work to DT system modelling

The comprehensive survey [18] offers a global perspective on how entities and interactions in Complex Networked Systems (CNS) can be represented, focusing on methods that preserve the complexity and heterogeneity of networked systems. The presented modelling paradigms include: Basic Graph-Based Models, Probabilistic Graph-Based Models, and Network Embedding-Based Models. Each of these approaches increases in complexity but also in capability, enabling richer and more accurate network representations as the complexity of the system grows.

Another influential work in [19] introduces a dynamic knowledge-graph approach to modelling DTs, leveraging semantic web technologies. This approach includes a "base world," maintained by real-time data updates, and "parallel worlds" for exploring alternative network designs without affecting the real-world model. Such knowledge-graph methods highlight how graphs can offer flexibility and depth in modelling complex DTs.

Several research efforts have applied graph-based models to networking. For example, [20] proposes an NDT reference architecture from a software engineering perspective, identifying key modelling elements from standard network architecture documentation (e.g., ETSI) and network simulators (e.g., ns-3 and OMNET++). This research highlights the need for robust metamodels that define NDT elements, attributes, and relationships.

In [21], a vendor-agnostic data model is used to simplify complex network management tasks. The system defines relationships between network entities, such as shared session attributes, and physical relationships, demonstrating the power of graph-based models in simplifying network operations.

3.2.2. Static graph representation

Given the high complexity and requisite processing power, constructing a single NDT encompassing all network aspects and scenarios is impractical. Therefore, the development of an NDT is **inherently use-case-driven**, with its data and models chosen precisely for the scenario it serves.

For the sake of simplicity, we propose the scenario illustrated in Figure 9, where we have a set of UEs connected to the network via the RU.

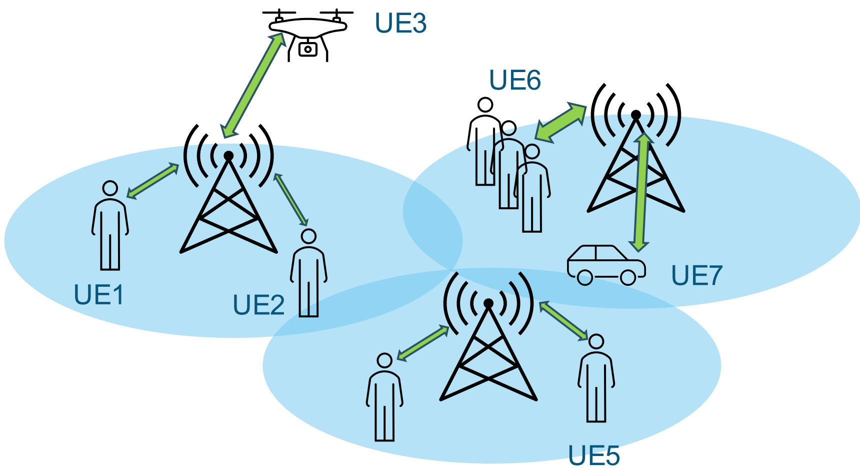

Figure 9 Scenario of a physical network to be twinned.

Building on the idea of graph-based modelling, we propose a preliminary graph representation of the scenario in Figure 10, illustrating the main entities and their logical or physical relationships. We treat the channel model as a distinct node due to its inherent complexity and stochastic nature. Complex nodes like the channel model can be captured through

Co-funded by the European Union

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

mathematical models, emulations, or data analysis as described in Section 3.1.1. Furthermore, the channel model node contains all the environmental and contextual factors not listed in the data taxonomy.

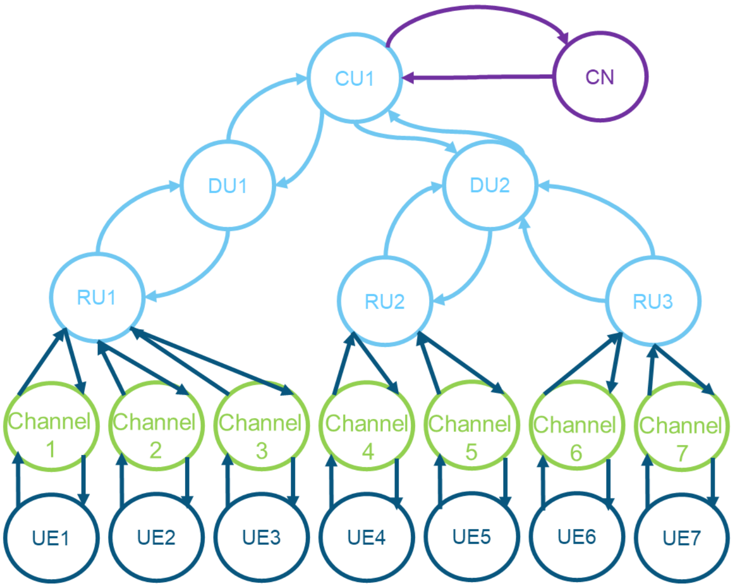

Figure 10 A graph representation of the scenario.

In fact, with a "higher granularity" level of representation, each node of the graph presented in Figure 10 represents a graph itself, following the decomposition into IOCs we propose in this document. Therefore, we show in Figure 11 an example of the different IOCs that represent the RAN part. These relations are mainly in line with the defined 3GPP standards [9].

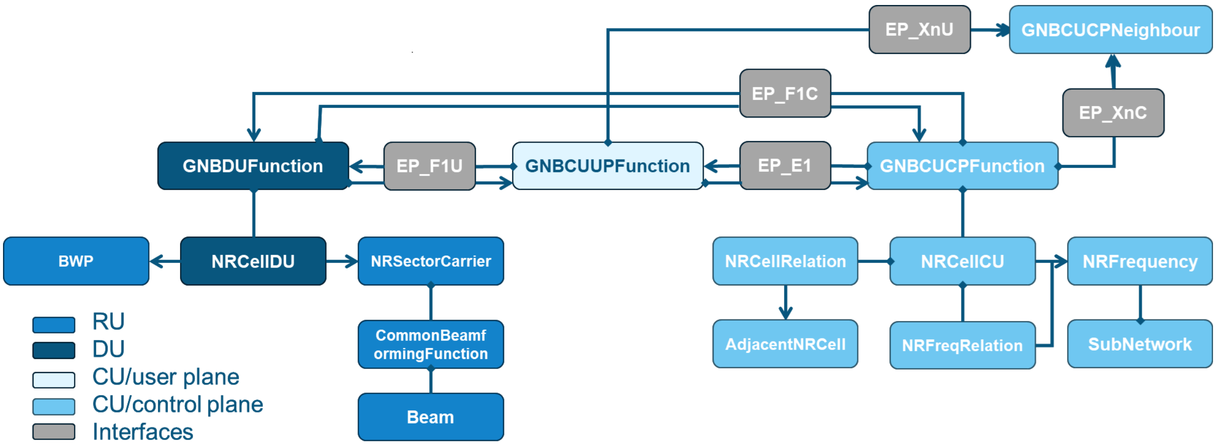

Figure 11 Several RAN IOCs and their relations.

A similar graph-based representation can be extended to the CN domain. For instance, Figure 12 shows the main VNFs and their relation to UE and RAN domains [22]. This modular graph-based approach allows for better flexibility in modelling the nodes, depending on the needed fidelity of the scenario [5].

Co-funded by the European Union

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

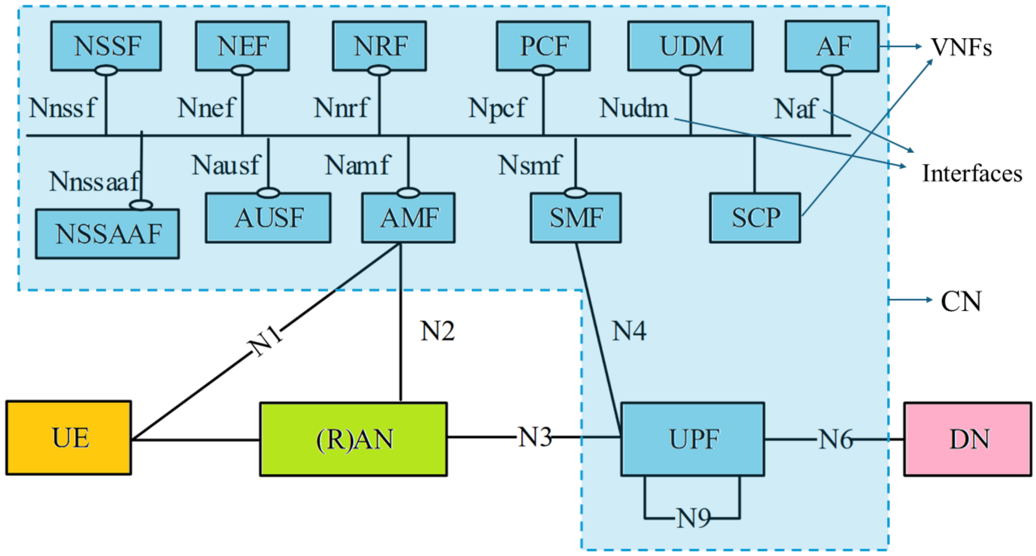

Figure 12 Main CN VNFs with their relations through interfaces [22].

3.2.3. Dynamic graph representation

While the straightforward graph-based approach offers a structured and intuitive way to model simple network scenarios, it faces significant challenges in terms of scalability and flexibility. A major limitation is that it provides only a static snapshot of the network at a specific moment, making it inadequate to capture the dynamic behaviour of NEs. This restricts its ability to model ongoing changes, predict future states, or implement real-time adjustments.

To address these limitations, more advanced tools are required that can incorporate both temporal dynamics and uncertainties. Probabilistic graph models, such as Markov chains and Bayesian networks, offer a way to represent the inherent uncertainty in the relationships between network nodes [18].

Graph embedding techniques provide a powerful alternative for handling high-dimensional graphs more efficiently. Graph embedding reduces the complexity of graph representations by embedding node information into a lower-dimensional space, which allows for better graph construction and inference. This technique combines an encoder to extract node features and a decoder to reconstruct the graph in its simplified form. Common methods for graph embedding include Principal Component Analysis (PCA), deep learning techniques, and more advanced approaches like Graph Neural Networks (GNNs).

The objective of the next section is to provide an extensive literature review of methods to derive network models based on traffic data and focus on the relation to 6G-TWIN basic models.

Co-funded by the European Union

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

4. Deriving network models from traffic data

Complex networks, such as those supporting future 6G systems, require precise models to forecast and optimise performance effectively. Traffic data plays a critical role in this process by creating dynamic basic models that reflect the true behavioural aspects of live networks, moving beyond traditional expert-driven methods derived in previous chapter.

This chapter provides a comprehensive review of existing methods for deriving network models from traffic data, with a specific focus on their relevance to the project. The techniques reviewed span from probabilistic models to graph-based frameworks, highlighting their strengths and limitations.

The chapter concludes by presenting the 6G-TWIN approach for constructing dynamic basic models from traffic data based on GNNs, which in fact extract complex relationships between network components, forming spatial and temporal graphs that accurately capture the network's operational dynamics. The insights and relationships derived from traffic data serve as the foundation for advanced tools, discussed in Chapter 5, that further refine these basic models for simulation and optimization purposes.

4.1. Probabilistic methods

Probabilistic modelling methods employ statistical correlations within data to forecast network behaviours under unknown circumstances. These models assist in managing the intricacies of changing traffic patterns by assessing the probability of different outcomes [23]. These statistical techniques are integral to DT frameworks, as they boost the twin's capability to predict network activity under varying conditions, enabling it to replicate and adjust to real-time dynamics proficiently. Many research works have introduced diverse approaches for network modelling, leveraging various techniques to enhance predictive accuracy and adaptability in complex environments, for instance in [24]. Current probabilistic methods, like Bayesian Networks and Hidden Markov Models, aim to capture dependencies and structures within network data, therefore improving predictive precision and adaptability in dynamic environments.

4.2. Graph-based methods

Graph-based methods are effective tools for modelling and evaluating network architectures by capturing the interactions and interdependence among nodes. These approaches use graph theory to represent NEs as nodes and their interactions as edges, which enable a structured and systematic perspective on complex network topologies. These techniques are also very useful for applications such as traffic forecasting, anomaly identification, and resource allocation in dynamic network settings. These methods are still being investigated and expanded in several studies to improve network modelling and prediction capabilities. For example, the paper [25] thoroughly examines network traffic analysis and prediction

Co-funded by the European Union

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

techniques. It explores techniques such as neural networks, data mining, and time series models, emphasizing their utilization in proactive network management for security, congestion mitigation, and resource distribution. Important models addressed include linear (e.g., ARMA) and nonlinear (e.g., neural networks and GARCH) methods, integrating hybrid models to enhance precision. The research underscores the need for standardized datasets and preprocessing methodologies to guarantee uniform and effective outcomes. It combines current developments in predictive traffic modelling to enhance network performance. Another review paper [26] studies current advancements in graph-based techniques and applications, highlighting the importance of GNNs in traffic networks, biology, and computer vision. The paper addressed scalability issues with graph algorithms, especially when handling large, sparse graphs and computational inefficiency. The main contribution is revising advanced GNN models for applications like action segmentation and anomaly detection and showcasing improvements in tasks like link prediction and time series analysis. Authors in [27] examine graph-based models as tools for communication network management and optimization. They focus on how these models might help with predictive tasks like traffic load forecasting and anomaly detection by capturing intricate topological structures and connections inside network data. To make these models suitable for dynamic network settings, the study addresses a variety of graph representations and presents techniques that improve data handling efficiency. In [28], the authors focused on the application of GNNs in traffic forecasting and network management. They introduced methods that integrate spatial and temporal features, which are crucial in understanding traffic flows. Techniques such as Diffusion Convolutional Recurrent Neural Networks (DCRNN) and Spatio-Temporal Graph Convolutional Networks (STGCN) are highlighted in this work and show their ability to process both static and time-variant network data for improved performance in real-time applications. aGNN approaches

GNNs offer an innovative approach to help manage data represented as networks, where relationships among data points (nodes) are as important as the individual data points. For example, in a mobile network, gNBs can be seen as nodes, and their relationships (direct link or interference) as edges connecting them. This makes GNNs useful for traffic forecasting, topological modelling, and recommendation systems. Traditional neural networks, such as Convolutional Neural Networks (CNNs), are limited when dealing with data in graph form because they work well with fixed-size inputs like images but struggle with graphs' irregular and interconnected structure. In contrast, GNNs capture these complex relationships by representing the data as a graph, where edges connect nodes. These traditional methods consider data points independently, whereas GNNs enable more comprehensive modelling of real-world data's interconnected nature by dynamically modifying each node's representation following the behaviour of its neighbours [26]. As shown in Figure 13, the input graph is passed through a series of neural networks that encode information about nodes, edges, and the overall graph structure. This process is also known as graph embedding, where each node's features are updated following its connections to other nodes, enabling GNNs to retain critical information about the relationships within the graph. For example, if point A is directly connected to point B, point A can gain information from point B’s characteristics, and vice versa. By repeating this process throughout the network, the model creates a detailed representation of connections, enabling tasks such as predicting traffic flow, analysing delivery routes in logistics systems, and understanding patterns in transportation networks.

Co-funded by the European Union

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

Recent advancements in GNNs have further improved their ability to efficiently propagate and aggregate information across large-scale networks. These capabilities have proven particularly valuable for predicting traffic patterns in interconnected road networks and analysing social behaviours influenced by community structures in social network analysis [29]. Despite their versatility, challenges remain in scaling GNNs for very large or dynamic graphs, as traditional deep-learning architectures struggle with irregular data and high computational demands. Innovations in hierarchical and pooling techniques address these limitations by allowing GNNs to learn representations at multiple graph levels.

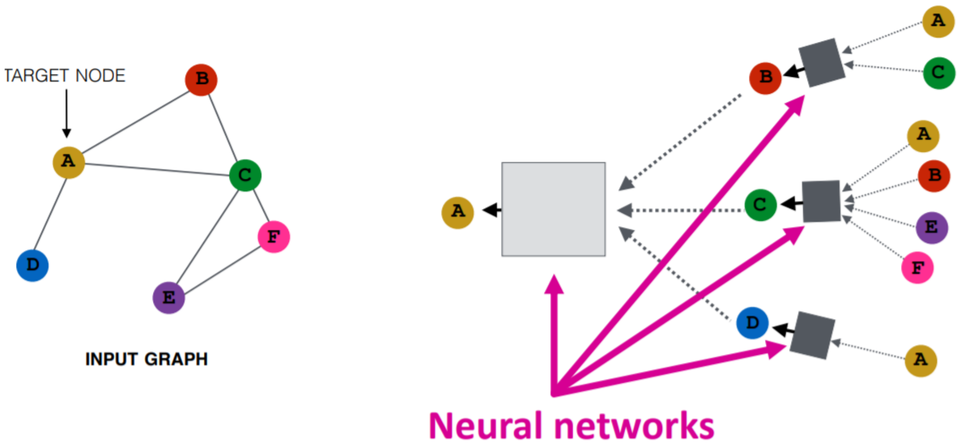

Figure 13 Graph Neural Networks.

As interest in GNNs grows, so does the development of open-source tools and benchmark datasets, fostering rapid progress in applications and research and positioning GNNs as essential tools for next-generation data-driven problem-solving [30].

4.2.1. DCRNN

The Diffusion Convolutional Recurrent Neural Network (DCRNN) is a specialized model designed to handle spatiotemporal forecasting challenges, especially in contexts where data is highly interconnected, such as telecom and transportation networks [31]. DCRNN models spatial relationships as a diffusion process on directed graphs, enabling it to capture non-Euclidean and directional dependencies that typical models overlook. DCRNN effectively models how information flows across a network using bidirectional random walks, while its recurrent neural structure allows it to handle temporal dependencies, such as fluctuating traffic patterns. With its encoder-decoder architecture, DCRNN maintains prediction accuracy over extended time horizons, proving effective for both short- and long-term forecasts [32]. Tests on real-world datasets have shown that DCRNN outperforms standard machine learning models such as ARIMA, SVM, Feedforward Neural Networks, and LSTM in accuracy, highlighting its potential for advanced network management and other spatio-temporal applications.

Co-funded by the European Union

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

4.2.2. ASTGCN

AST-GCN (Attribute-Augmented Spatiotemporal Graph Convolutional Network) model focuses on improving network traffic forecasting by integrating both dynamic and static external factors, such as network conditions and mobile traffic patterns. AST-GCN improves the spatiotemporal graph convolutional architecture by representing external variables as network nodes' attributes, capturing the complicated relationships between traffic flows and external impacts.

This model improves network traffic prediction by considering not only historical data but also contextual information, leading to more accurate forecasts. Integrating external parameters enhances the model's understanding of real-world scenarios, including variability in traffic resulting from different network conditions or mobile traffic load.

Experiments demonstrate that AST-GCN performs better than conventional forecasting models by achieving higher accuracy, especially in dynamic conditions.

The model's effectiveness lies in leveraging external influences and temporal dependencies, offering better interpretability and more robust mobile and network traffic predictions. This approach represents a significant advancement in traffic prediction models, with applications in optimizing network performance and traffic management [33], [34].

4.2.3. GWN

Graph WaveNet (GWN) is an advanced spatiotemporal GNN model designed for accurate traffic prediction in complex network environments. GWN builds on earlier models like DCRNN by combining graph convolutions for aggregating information from nearby sensors with dilated convolutions to efficiently capture both spatial and long-term temporal dependencies. Notably, the model addresses the challenge of overly emphasizing sensor proximity in the adjacency matrix by learning sensor embeddings for more precise correlations between sensors. The model performs better with a few important changes, such as hyperparameter tuning and adding connections for a bigger gradient flow. The integration of a pretraining strategy for short-term traffic forecasting further optimizes results GWN's model architecture also significantly reduces training and inference time compared to prior models, making it a powerful tool for real-time traffic forecasting in large-scale networks [35].

4.2.4. AGCRN

The Adaptive Graph Convolutional Recurrent Network (AGCRN) is designed specifically to address the complex challenges of traffic forecasting by capturing spatial and temporal dependencies without relying on predefined graph structures. Node Adaptive Parameter Learning (NAPL) and Data Adaptive Graph Generation (DAGG) are two essential modules used by AGCRN in network traffic modelling, where traffic data frequently reflects complexities between geographical nodes and fluctuates greatly over time. NAPL enables the model to learn unique node-specific patterns, allowing for more nuanced representations across different traffic nodes. Meanwhile, DAGG dynamically infers and updates the spatial dependencies, which lets the network respond more flexibly to traffic changes over time without needing pre-set spatial graph connections. The model's recurrent architecture

Co-funded by the European Union

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

(integrating both convolutional and recurrent layers) further enhances its ability to model long-term temporal trends, making it more accurate in both short- and long-term predictions. AGCRN exhibited substantial enhancements over previous models in real-world traffic datasets by accurately capturing the evolving spatial and temporal patterns inherent to traffic flow, thereby effectively handling the challenges of heterogeneity in urban traffic data [36].

4.2.5. GMAN

Graph Multi-Attention Network (GMAN) is a model built to improve the precision of long-term traffic prediction by addressing the intricate spatial and temporal factors that affect data traffic flow. Unlike simpler models, GMAN uses a series of multi-attention layers and a structure able to capture both short-term variations and long-term dependencies. It uses an encoder-decoder framework enhanced with spatio-temporal attention blocks, identifying the relationships within spatial regions and across time points. A special feature in its transform attention layer is the ability to connect historical data directly to future predictions, mitigating the issue of cumulative errors that can arise in sequential forecasting. In addition, GMAN's layered approach enables the model to process nonlinear changes and sudden traffic fluctuations, capturing interactions that vary by location and time [37].

4.3. Network models and Basic Models

The output of all the algorithms described in the previous section after training is normally unsuitable to incorporate in a Basic Model. The intrinsic mathematical nature of the neural networks embedded in most of the Graph-Based approaches, together with the encoder-decoder architecture, makes it difficult to represent the network model effectively.

Luckily, neural networks are usually trained using libraries such as *Tensorflow*, PyTorch, and the *Keras* high-level API. These tools provide a practical way to save and export the weights of the trained model in structured file formats like H5. H5 is one of the Hierarchical Data Formats (HDF) used to store large amounts of data. The format is widely used to store the weights of a neural network for its quick retrieval and analysis capabilities [38].

However, saving and storing weights may not be enough to fully recover the model after training, and other metadata, such as layers configuration, need to be saved. In this context, a JSON structure can be used to save the model metadata.

Having saved the weights of the embedded neural networks and the related metadata, along with the architecture of the approach in mind, it becomes possible to plug this information into a Basic Model. This representation can encompass not only the behaviour of individual network devices but also higher-level network dynamics, such as resource allocation or node interactions. This enriched Basic Model can then be passed to a simulator capable of programmatically implementing the algorithm and effectively reproducing the dynamic behaviour of a network. The simulator needs to have the general architecture of the used GNN approach embedded in that is parametrized once a specific model is imported. Figure 14 shows a typical pipeline representing the process described above. In the first stage, the GNN-based algorithm is designed and trained. Then, the resulting weights are saved in one or more

Co-funded by the European Union

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

H5 files, and the structure-related metadata are saved in a JSON file. These files are part of the representation of a Basic Model that reproduces the behaviour of a common network element. At runtime, the files are plugged into a simulator to parametrize a vanilla implementation of the same GNN algorithm and make it ready to be used in the simulation. The next chapter provides a selection of frameworks and platforms in which the dynamic basic models described here can be embedded.

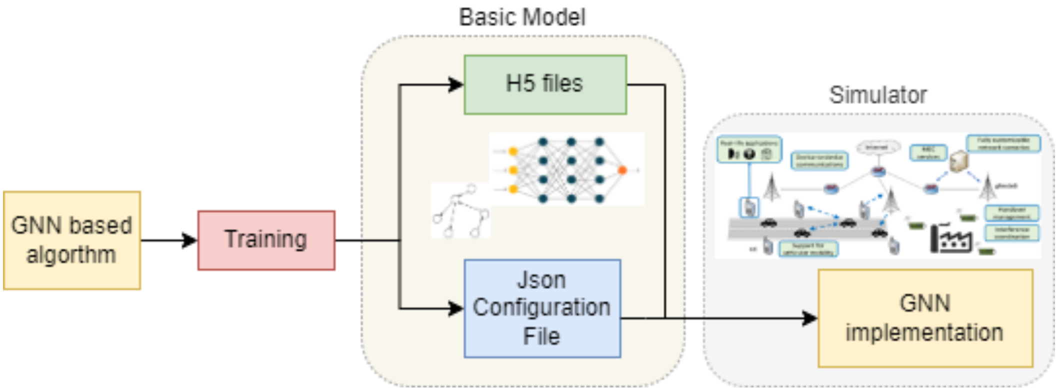

Figure 14 Embedding of a GNN model in a Basic Model.

The models described above will be further elaborated in the final version of this deliverable, as certain dependencies within the project, such as data availability, AI integration, and other technical considerations, need to be resolved first.

5. Tools for graph-based NDT

This chapter focuses on tools used to build graph-based network models. Firstly, we review the literature around tools used by the various DT communities, we then choose an adequate tool to be used in the framework of 6G-TWIN.

5.1. State-of-the-art review

Existing DT platforms and frameworks are mainly driven by data acquisition, such as the Internet of Things (IoT), big data analysis, and cloud computing. Therefore, they provide a solution to structure the data, which could cover the basic model in the NDT context. Those tools rely on description languages to describe the structure of the data and the data itself. Existing surveys, such as [39], propose some classification of DT platforms based on general DT concerns and definitions. These classifications serve as a foundation, which has been adapted here with a specific focus on the requirements for basic models and their integration within the 6G-TWIN architecture. These requirements are derived from the preliminary work on system architecture (D1.1) and data (D2.1) done in 6G-TWIN, ensuring alignment with the project goals and functionalities.

In the following, we focus on a selection of DT platforms and frameworks that could support our definition of basic models and have or allow third-party tools to define functional models. Notably, we want to have modelling capabilities that allow for the definition of a data model based on our definition (supporting modelling capabilities like UML class diagrams) and support, to some extent, the conformance checking of the data with this provided data model.

The focus of this project's NDT aligns with the capabilities and corresponding architectural components outlined for DTs by the Object Management Group (OMG), as depicted in the figure below. These capabilities are also referenced in the Local-Digital Twin toolbox [40]. This table encompasses many capabilities, supported by existing platforms such as Azure DT. Amongst them, we need domain-specific data management and DT model repository, which are at the heart of the graph-based Digital Twin.

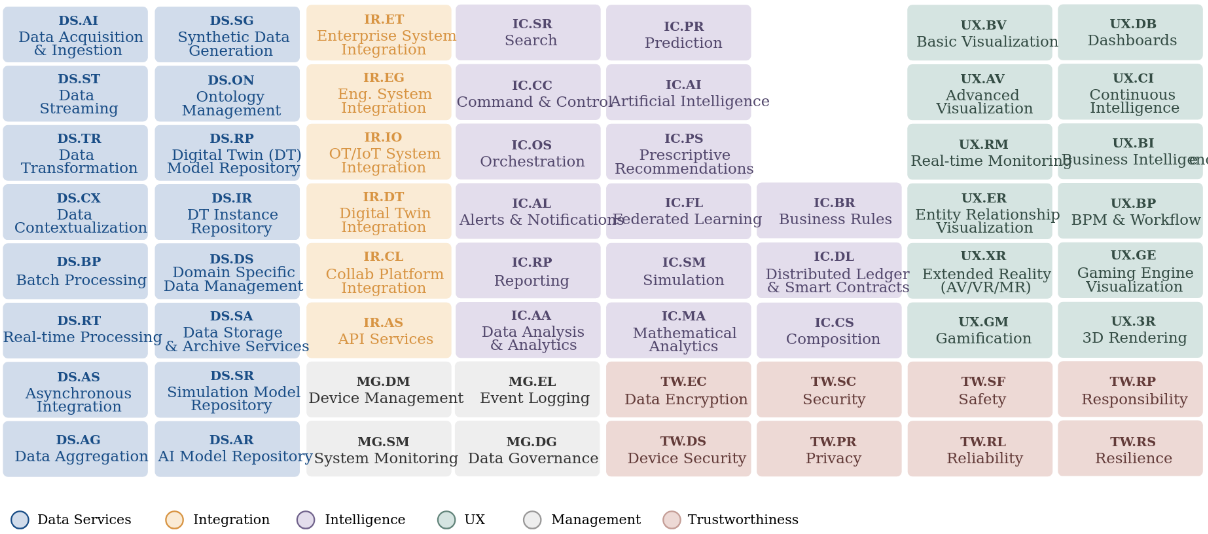

Figure 15 OMG DTC Periodic table of Digital Twin capabilities.

5.2. Description of Platforms and Framework

5.2.1. Azure Digital Twin

It is a comprehensive IoT framework that provides an environment to create DTs. A JSON-based language, i.e., Digital Twin Definition Language DTDL [41] is used to provide schema definition by describing the digital models of physical environments, systems, and devices, termed as DT. DTDL uses the concept of interfaces, consisting of: (i) identifier, (ii) type, (iii) context, (iv) comment, (v) contents (with inner embeddable schema telemetry, property, command, relationship, component), (vi) description, (vii) displayName and (viii) extends (for inherited interfaces). This framework offers the creation of static DTs, which require a back end to model an evolving environment; however, the back end, provided by Azure, is not free to use. A graph-based modelling approach represents the relationships and interactions between real-world entities through **model** and **twin graphs**. This platform has significant importance as it allows the definition of structure, behaviour, and relationship among entities in a standardized way. It is also flexible enough to integrate seamlessly with other services to enable data ingestion, event-driven workflow, flexibility with DTDL, and advanced data analytics [42].

5.2.2. Eclipse Basyx

This platform facilitates a Java/.NET implementation of the Asset Administration Shell (AAS) standard, offering infrastructure for both AAS and registry servers as services through docker container or JAR executable. At the core of the AAS specification [43] is an information meta-model designed to represent assets, including machines, their components, capabilities, and relationships. It organizes these elements using a hierarchy of assets, sub-models, and properties, commonly referred to as sub-model elements. It also offers procedures to facilitate the creation, reading, updating, modification, and deletion of AAS objects, sub-models, and elements, supporting AAS types.

This platform also provides a Graphical User Interface (GUI) client to manage AAS objects through OWL, RDF, JSON, or XML. It is designed as feature-rich Software Development Kit (SDK), enabling integration across various networks and protocols, i.e., (i) TCP, (ii) HTTP/REST (APIs), and (iii) MQTT, through Virtual Automation Bus (VAB), providing semantics for interacting with AAS components. The software supports secure HTTP connections and allows asset interfaces to trigger operations via HTTP/REST and VAB, though synchronization issues may arise. Comprehensive documentation, examples, and Eclipse supports are available, alongside limited SDKs for Python, C++, and RUST [44].

5.2.3. Open Twins (Eclipse Ditto + Ecosystem)

Open twins [45] is an open framework based on an ecosystem of components allowing for:

1. Digital twin schema definition.
2. Connection with IoT devices and collection of their information.
3. Storage of digital twin data in real-time series.
4. User-friendly visualization of data.

The central component is Eclipse Ditto used to align IoT information with a twin model, i.e., the definition of higher order entities which are aggregating different data coming from sensors. It allows for the definition of basic metamodels and models. Metamodels as digital twin entities (types) and its instances (models) under an entity-relationship form, which can be expressed as a graph. It also allows integration of third-party simulation tools, AI/ML models for prediction (i.e., functional models).

5.2.4. FIWARE (Context Broker)

European Telecommunications Standards Institute (ETSI) along with Industry Specification Group on Context Information Management (ISG CIM) developed a specification known as FIWARE Next Generation Service Interfaces – Linked Data (NGSI-LD) [46].

It is an extension of NGSI-V2 standard, adding linked data principles for supplementing the richness and interoperability of data models. The main purpose of NGSI-LD is to manage context information which can be significant for creating DT of any physical entities. NGSI-LD uses JSON-LD to represent context structure and data. JSON-LD is significant for enabling data across different entities and sources, making it easier to integrate heterogeneous datasets, from multiple systems, while maintaining semantics.

Scorpio [47] is an open-source context broker, which is offered by NGSI-LD, enabling the management, access, and discovery of context information. It models context information in a graph structure consisting of entities, attributes, and relationships. Scorpio also allows services and applications to request context structure and context data followed by modelling of DTs for real-world entities. NGSI-LD provides a wide variety of methods, i.e., create, read, update, delete, along with a sophisticated query and subscription system, to interact with DT.

Co-funded by the European Union

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

5.2.5. GreyCat

DataThings offers GreyCat [48] which is built on a dynamic, evolving graph structure where nodes store various properties, including large time series or geographical data. It features an executable language with imperative syntax for defining types, traversing, and modifying graph nodes. It combines graphs and time series data into a scalable storage system. It is also able to organize a huge amount of dynamic and unstructured data into multi-dimensional data models, known as Many-Worlds Graphs [49]. Hence, we summarize that it is a programmable data graph designed for advanced data management and processing, which emphasizes real-time data processing, scalability, and flexibility of different applications, i.e., real-time data analytics, live monitoring, and DT solutions. The platform allows the creation of DTs by manipulating historical data for prediction and simulation engines. Learning models are used to update and refine real-time DTs, ensuring high accuracy.

5.3. Support for Graph Based NDT: analysis

Based on the previously mentioned tools, this section proposes a deeper view on how those approaches could support graph-based NDT.

Eclipse Basyx proposes to rely on the AAS modelling standard. It offers some facilities to represent assets, but it lacks some of the property we need for a proper graph modelling (e.g., composition, generalisation, etc.). Indeed, through the concept of sub-models it will finally represent elements as trees.

FIWARE context broker implementation lacks schema verification; indeed each of the exchanged model could follow its own schema (each message contains the data schema it should conform to). It could hamper the harmonization part if, at least, two different providers are not consistent in their schema definition. If FIWARE offers a native integration of smart-data models, it is not providing explicit support for a graph representation, but rather a collection of model elements.

Azure DT offers a good graph-based representation of the DT's collected data, as well as offering facilities to express operation (basic model functions) that could help to represent the behaviour of the system. Basic functions are not part of the graph and should be defined separately. Nevertheless, bridges will have to be established from the data collection part (harmonized repository) by porting smart-data model specifications into Microsoft's DTDL standard. Moreover, advanced functionalities are not available on a free/open-source base mode.

GreyCat offers a graph-based approach that will allow to store basic models into the graph as nodes, including the basic functions that represent the behaviour of the system. A bridge between smart-data models and GreyCat offers a low memory footprint and quick response, providing APIs and facilities the capability to connect to any AI/ML functions (i.e., functional model) and handle its graph.

Co-funded by the European Union

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

In practice, all of these frameworks lack of the necessary flexibility to support the natural evolution of the NDT, as well as its continuous development. In fact, the NDT will face many interoperability challenges beyond data harmonization, we need to also be able to harmonize entity descriptions inside the system model. Different operators, network devices may describe similar elements following a different semantical viewpoint. The semantic misalignment and evolution are not well supported by current DT platforms [50].

5.4. 6G-TWIN's approach

As stated in the previous section, concluding on the tools, we need 1) support for the connection to smart-data models coming from telemetry and knowledge about the system (i.e., including the collected data into the basic model graph structure) and 2) providing support to the evolution and connection to other components, notably delivering dedicated views (illustration given on simulation environment) and a shared semantic (relying on smart data models). In this section, we will explore the case of 6G-TWIN and the approach we would like to explore and prioritise to implement our models.

5.4.1. Overview

We propose an open-source low-code approach for the implementation of support for basic-models in NDT; The definition of basic models is in practice framing the DT content (harmonising and integrating data and knowledge about the network) as well as defining the basic behaviour of the NDT. It comes from the need of building smart-data-models which will be progressively refined through practice, during the implementation of the different case studies. The first example of smart-data models we propose is only based on the standard definition. Our approach will thus allow to deal flexibly with the refinement of such data schema.

Furthermore, a low-code approach allows the expression of basic models without enforcing a technological proprietary choice and aiming at offering the possibility to bridge different implementations/tools, such as any of FIWARE, Eclipse Basyx, or Azure DT, for instance. It could also depend on the targeted case: Indeed, we could deploy one solution with low memory footprint requirements (e.g., GreyCat 4Mo solution) or existing technical infrastructure. The WP4 case studies, will provide concrete technical requirements that we will have to fulfil. Our low-code approach, remaining generative, can support a large variety of scenarios.

Beyond its goal, being a platform agnostic solution, it also aims at facilitating the connection to other component developed in the frame of the project, with a peculiar focus on simulators. Our envisioned approach will be developed using a python open-source framework called BESSER [51], to design low-code application maintained by LIST [52]. As shown in Figure 16, we present a low code approach to help to:

- Synchronize with smart data models and data schema, notably extending the pure data part with the system knowledge. It is possible to already import the provided data into Besser instances (i.e., to be reused as export to external tool).
- Provide a configuration for the targeted platform, initializing and instantiating basic models into the targeted platform.

Co-funded by the European Union

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

- Prepare its deployment and integration into the overall 6G-TWIN architecture based on the actual/envisioned component of the 6G-TWIN architecture (API generation)
- Provide a support to NDT management: configuring components insides the selected platform (depending on platforms capabilities).
- Provide transformation of NDT graph for other components. For instance, targeting simulation configuration / parameters.

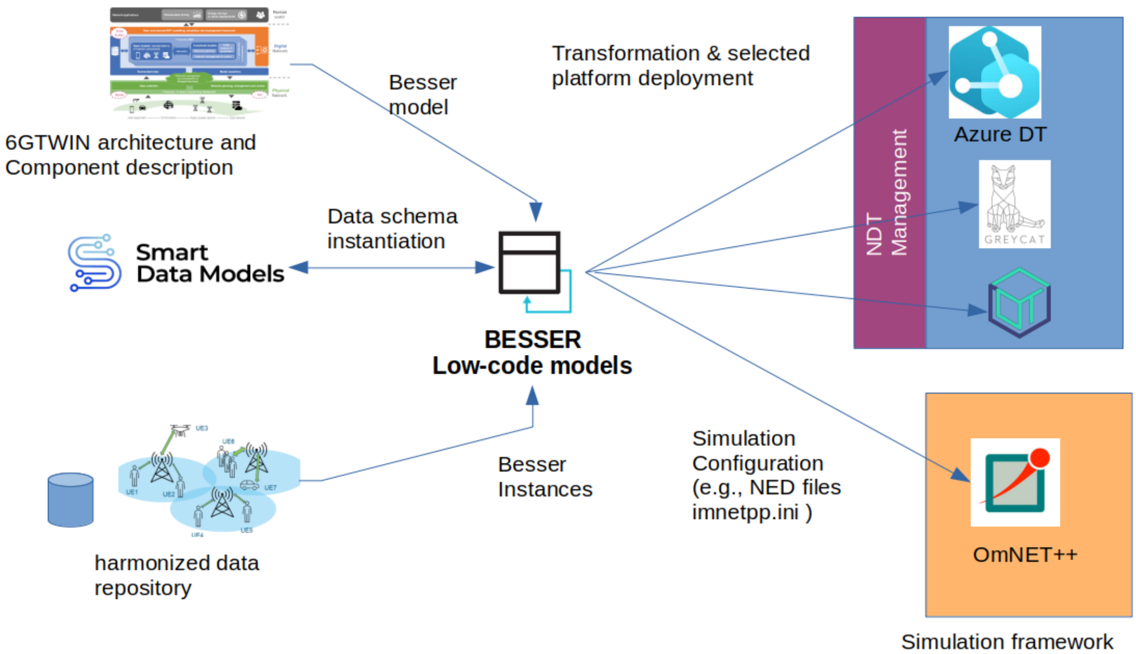

Figure 16 Low-code approach: targeting different platforms according to the case requirements.

5.4.2. Focus on a graph-based NDT tool

GreyCat provides supports for different part of the 6G-TWIN architecture related to core models of the NDT. It includes data (lake) storage (TC4 in Figure 17), model register (to expose all methods that affect the basic models including functional models). It provides a full support of basic models (TC5), and it offers some support for NDT management lifecycle (partial converge of TC3): managing versions and evolutions of the graph-based NDT.

Co-funded by the European Union

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

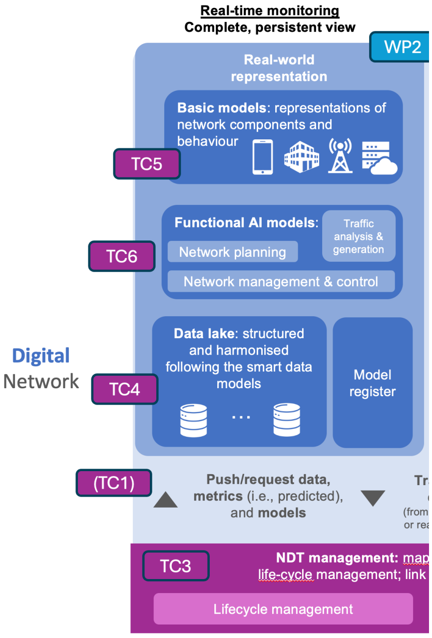

Figure 17 Part of the technical infrastructure supported by the GreyCat tool.

Greycat offers an API to facilitate the integration with any Python function (functional model) to process its internal graph (basic models). It supports storing and execution of functional models (TC6). Practically, Greycat proposes a full graph-oriented approach where functions and entities are graph nodes. Figure 18 provides an example of GreyCat GUI used to build graph-based model of an electrical grid.

Figure 18 GreyCat example on electrical grid, including analytical views.

Co-funded by the European Union

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

6. Integrating Basic Models in the 6G-TWIN Framework

This chapter relates basic models presented in this deliverable to the other activities in the 6G-TWIN project.

6.1. Relation to 6G-TWIN's data collection framework and overall architecture

One of the crucial steps of designing the NDT system architecture is building the Data Collection Framework. The work done on Deliverable D1.2 outlines the specific Data Pipelines that enable the flow of data across the NDT. It is necessary to build a specific pipeline for telemetry data collection, as this data is essential for the network to generate real-time insights into its performance.

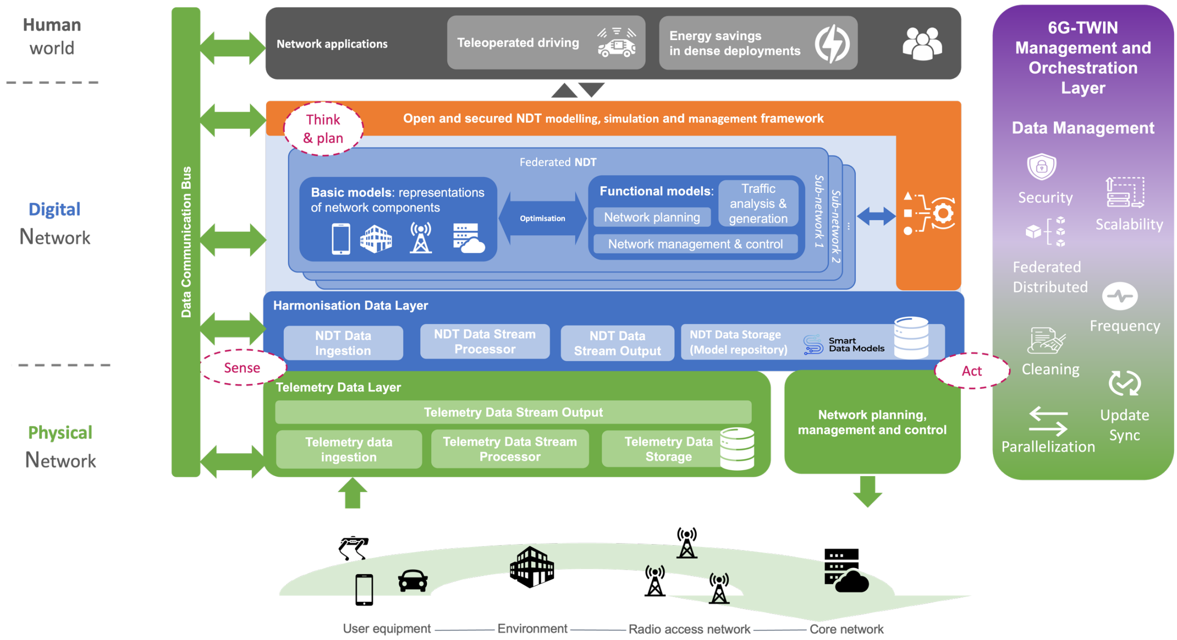

Figure 19 6G-TWIN data collection and exposure framework.

The representation of Basic Models outlined in this deliverable is considered by the design of these data pipelines, as data harmonisation and pre-processing mechanisms are required to ensure that the data collected from multiple sources is stored in a unified format. Basic models, as digital representations of real network entities, use information regarding network topology model of this NDT entity based on the basic configuration, environment information, operational state, link topology and other aspects to emulate changes in these network elements, to develop solutions that optimise the performance of this network entity.

For exposing the services of the NDT towards the network devices we have the Digital Twin Connector (DT-C). It interfaces between the physical and the digital layer in our NDT architecture, making sure the connected devices to our NDT can generate telemetry data, receive network configurations for optimised performance, and initiate relevant NDT applications that support lifecycle management functions. For the purposes of exchanging data between the NDT and the network devices, this DT-C has a Telemetry Data Injection module, in charge of extracting data from the physical network devices, handling different data types and different device communication protocols; a Telemetry Data Stream Processor, which handles the Data Harmonisation and Data Preprocessing mechanisms, that ensure data collected is consistent, to be used for the purposes of creating basic models, but also to be easily stored and accessed by the NDT. Finally, a Telemetry Data Stream Output supplies the digital layer with this collected data, through a Data Communication Bus, so that it can be used for the internal processes of the NDT, MANO applications and, of course, basic models.

6.2. Relation to 6G-TWIN's simulation framework

The aim of NDTs for 6G networks is the optimization of the real networks by running algorithms in the DT. Typically, such optimization algorithms contain simulations of what-if scenarios. Although these scenarios do not exactly match the current state of the real network, a simulation of the real network is needed to: (1) check that the simulation matches the real measurements sufficiently and (2) evaluate the outputs of that simulation as a baseline for comparing the outputs of the what-if scenarios with that baseline, for instance, by using the same KPIs.

Simulations require both *simulators* (i.e., simulation engines like OMNeT++, Matlab, or SUMO), perhaps extended by libraries (like Simu5G for OMNeT++), and *simulation models*. Simulation models are in that sense configuration files that define all details of simulation scenarios that are needed to be run by a simulator. The information that is needed for those simulation models comes mainly from basic models, because basic models contain the necessary information about network elements, networks and the basic algorithms running in them. Sometimes the basic models even contain parts of the simulation models themselves, for example algorithms of protocols or mathematical models of radio channels (see Section 3.1.1). Basic models also contain information about related measurement data (see Section 3.1). These measurement data are often not used to parametrize a model, instead it is used as input data of the simulation during runtime, as illustrated in Figure 20.

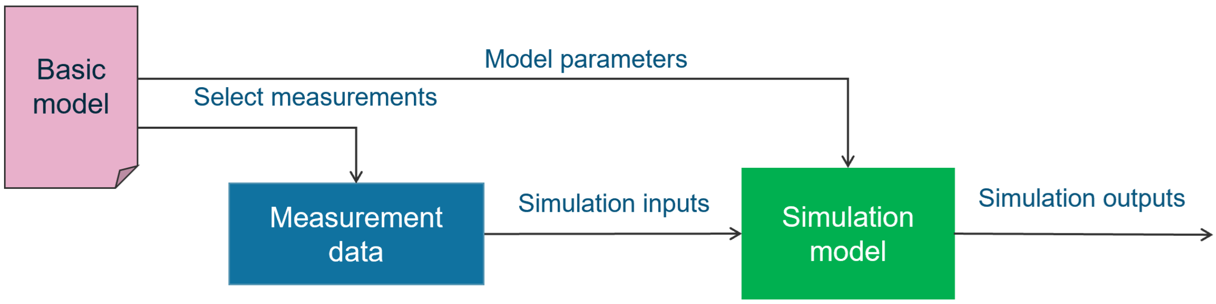

Figure 20 Usage of a basic model for creation of a simulation model

Simulations can consist of different simulators running together. One example is Veins [53], combining OMNeT++ and SUMO for simulating the network and the mobility, respectively. In the 6G-TWIN project, it is planned to develop a simulation framework that is a generalization of this approach (WP3). The simulation framework has the purpose to couple different simulators in a flexible way so that the selection of coupled simulators can fit to the purpose of the current simulation study and use case. For example, a teleoperated driving scenario might require (besides the RAN and core simulator) a mobility simulator like SUMO for modelling vehicle movements, while an energy-optimization scenario would instead require a detailed energy consumption model. Splitting the data contained in the basic model to the configuration files of different simulators is planned to be a joint work of the NDT management and the simulation framework, shown in Figure 21. The splitting must also consider which parts of the network are relevant to be simulated, because a simulation of the full network will often be too slow and contain details that have no effect on the aspects that are relevant for decision making.

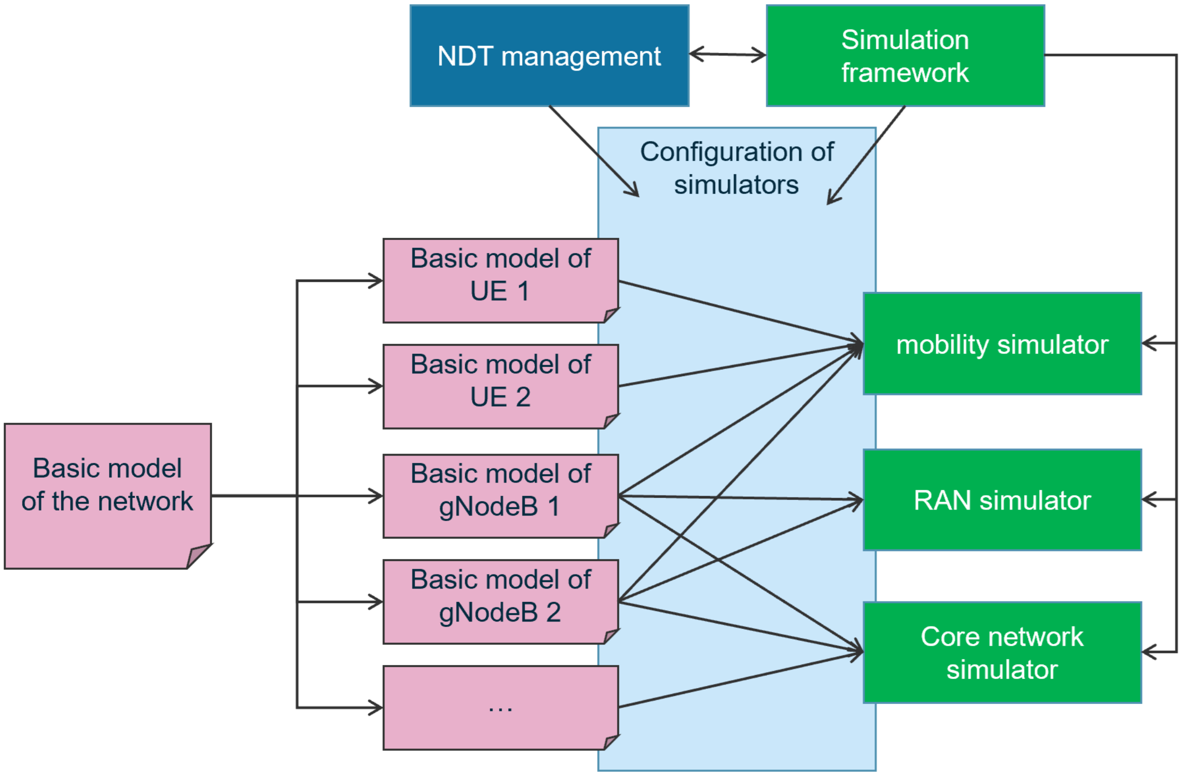

Figure 21 Configuration of simulators based on basic models

Besides basic models, also *functional models* can need the creation of a simulation model. This is the case if functional models describe special-purpose details of the network that are not part of the basic model but needed for the goal of the simulation study. However, in 6G-TWIN functional models are understood as tools for optimization and analysis of networks and in that sense, they *use* simulation models instead of being a basis for their generation. Examples for that could be evolutionary algorithms or machine learning using training of AI models, shown in Figure 22. Trained models can later also be used outside of the simulation framework, so a trained decision-making neural network may be used to directly affect the real network without needing simulations.

Co-funded by the European Union

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

Often it might also be useful to store outputs of simulations in the basic models. For example, this can be the case if simulations are used to get metrics of the network that are not measured in the real system but relevant for decision making. Then, an information path from the simulation outputs to the basic model is needed.

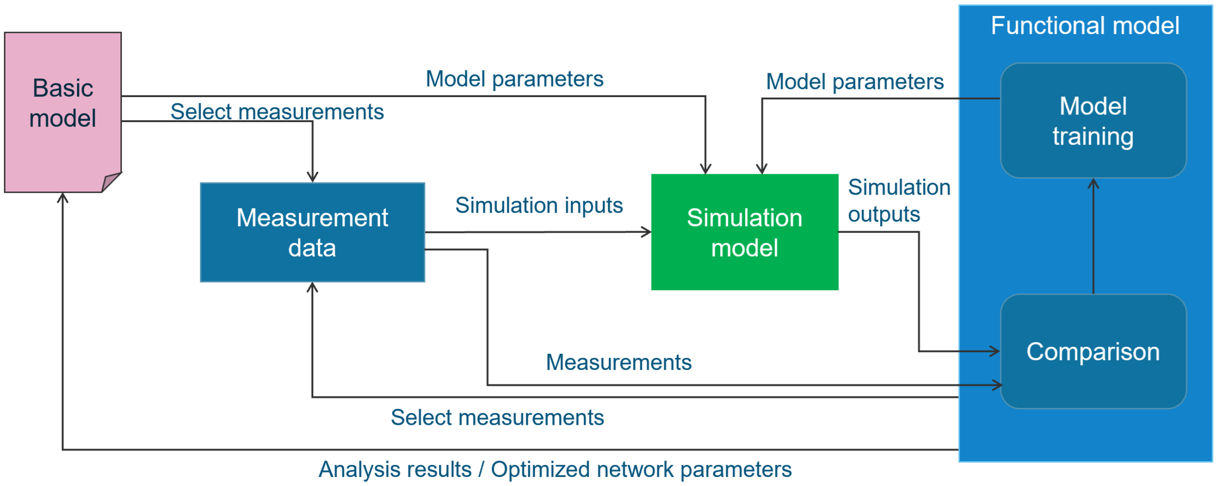

Figure 22 Basic and functional models in the context of simulation models

As basic models represent the state of a network, this state may also change over time, e.g. the number of connected UEs or their positions might change. In that case, it is not enough to use the basic model for creating a simulation configuration, but the changes of the basic models during a simulated scenario must be represented during runtime. Depending on the used simulator that might be realized by specifying these changes in a priori known trajectories (e.g. in VIAVI TeraVM RIC Test) or as time series inputs that are managed by the simulation framework (e.g. for Simu5G running in OMNeT++).

Co-funded by the European Union

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

7. Conclusions

This deliverable has detailed the foundational concepts, methodologies, and tools required to establish basic models as integral components of the 6G-TWIN Network Digital Twin (NDT) framework. These basic models enable the creation of virtual representations of network elements and network systems, laying the groundwork for advanced functional models and AI-driven network optimization.

Key contributions of this document include:

- **Formally defining basic and functional models:** The deliverable clarified the distinction between basic models, which reflect the state of the physical network, and functional models, which leverage insights for optimization and prediction based on the considered application.
- **Proposing basic model schemas for network elements:** Following the domain-based analysis of the physical network, this deliverable provided a harmonised schema for network elements modelling for each of the defined domains.
- **Developing graph-based representations:** A robust framework for modelling networks using graph-based approaches was proposed, enabling accurate and scalable representations of network elements and their relationships, based on traffic data analysis.
- **Reviewing state-of-the-art tools:** The document evaluated existing tools for graph-based NDT, recommending GreyCat for its ability to integrate real-time data processing, scalability, and compatibility with smart-data models.
- **Integrating basic models within 6G-TWIN:** The deliverable demonstrated the alignment of basic models with functional models (Task T2.3) and the data collection framework in WP1, in addition to their role in supporting simulation frameworks in WP3.

The insights and methodologies presented in this deliverable address the growing complexity of 6G networks and offer a pathway for managing dynamic, AI-enhanced systems. By emphasizing collaboration, standardization, and adaptability, this report contributes significantly to the 6G-TWIN project's goal of establishing an AI-native, federated network architecture capable of meeting the demands of future digital ecosystems.

Future efforts will build on this foundation by refining basic models, integrating them into functional models, and expanding their application to real-world use cases, ensuring a seamless transition from theoretical constructs to practical solutions in 6G network environments.

Co-funded by the European Union

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

8. References

[1] 'ITU-T REC-Y.3090 (2022/02), Digital twin network – Requirements and architecture'.

[2] M. K. Bahare *et al.*, 'The 6G Architecture Landscape - European perspective', Zenodo, Feb. 2023. doi: 10.5281/zenodo.7313232.

[3] 'ISO 23247-1:2021', ISO. Accessed: Nov. 21, 2024. [Online]. Available: https://www.iso.org/standard/75066.html

[4] F. Tao, B. Xiao, Q. Qi, J. Cheng, and P. Ji, 'Digital twin modeling', *J. Manuf. Syst.*, vol. 64, pp. 372–389, Jul. 2022, doi: 10.1016/j.jmsy.2022.06.015.

[5] 'ITU-T Y.3091 (12/2023) - Digital twin network – Capability levels and evaluation methods'.

[6] A. Zaki-Hindi, J.-S. Sottet, I. TURCANU, and S. Faye, 'Network Digital Twins for 6G: Defining Data Taxonomy and Data Models', Zenodo, Nov. 29, 2024. doi: 10.5281/zenodo.14244850.

[7] '3GPP TS 28.622 version 16.4.0 Release 16; Generic Network Resource Model (NRM); Integration Reference Point (IRP); Information Service (IS)'. Accessed: May 23, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/128600_128699/128622/16.04.00_60/ts_128622v160400p.pdf

[8] 'Smart Data Models'. [Online]. Available: https://smartdatamodels.org/

[9] '3GPP TS 28.541 V16.7.0 (2020-12) 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16)'.

[10] '3GPP TS 28.552 version 16.9.0 Release 16; 5G; Management and orchestration; 5G performance measurements'.

[11] H. Wang *et al.*, 'Digital Twin Channel for 6G: Concepts, Architectures and Potential Applications', Aug. 12, 2024, *arXiv*: arXiv:2403.12467. Accessed: Nov. 11, 2024. [Online]. Available: http://arxiv.org/abs/2403.12467

[12] 'TR 138 901 - V16.1.0 - 5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 16.1.0 Release 16)'.

[13] 'Guidelines for evaluation of radio interfacetechnologies for IMT-2020'.

[14] W. Zhuoran, 'What Is a 5G Transport Network?', Huawei. Accessed: Dec. 04, 2024. [Online]. Available: https://info.support.huawei.com/info-finder/encyclopedia/en/5G+Transport+Network.html

[15] 'TS 132 712 - V11.1.0 - Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Telecommunication management; Configuration Management (CM); Transport Network (TN) interface Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS) (3GPP TS 32.712 version 11.1.0 Release 11)'.

Co-funded by the European Union

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

[16]'3GPP TS 38.215 version 16.3.0 Release 16; 5G; NR; Physical layer measurements'. Accessed: May 29, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/138200_138299/138215/16.03.00_60/ts_138215v160300p.pdf

[17]'3GPP TS 24.501 version 16.5.1 Release 16; Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3'. Accessed: Jun. 27, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/124500_124599/124501/16.05.01_60/ts_124501v160501p.pdf

[18]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, doi: 10.1109/ACCESS.2022.3184801.

[19]J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, 'Universal Digital Twin - A Dynamic Knowledge Graph', *Data-Centric Eng.*, vol. 2, p. e14, 2021, doi: 10.1017/dce.2021.10.

[20]R. Verdecchia, L. Scommegna, E. Vicario, and T. Pecorella, 'Designing a Future-Proof Reference Architecture for Network Digital Twins', in *Software Architecture. ECSA 2023 Tracks, Workshops, and Doctoral Symposium*, vol. 14590, B. Tekinerdoğan, R. Spalazzese, H. Sözer, S. Bonfanti, and D. Weyns, Eds., in Lecture Notes in Computer Science, vol. 14590. , Cham: Springer Nature Switzerland, 2024, pp. 287–306. doi: 10.1007/978-3-031-66326-0_18.

[21]H. Hong *et al.*, 'NetGraph: An Intelligent Operated Digital Twin Platform for Data Center Networks', in *Proceedings of the ACM SIGCOMM 2021 Workshop on Network-Application Integration*, Virtual Event USA: ACM, Aug. 2021, pp. 26–32. doi: 10.1145/3472727.3472802.

[22]'3GPP TS 23.501 version 16.6.0 Release 16; 5G; System architecture for the 5G System (5GS)'. Accessed: Apr. 11, 2024. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/16.06.00_60/ts_123501v160600p.pdf

[23]M. Abbasi, A. Shahraki, and A. Taherkordi, 'Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey', *Comput. Commun.*, vol. 170, pp. 19–41, Mar. 2021, doi: 10.1016/j.comcom.2021.01.021.

[24]M. Conti, Q. Li, A. Maragno, and R. Spolaor, 'The Dark Side(-Channel) of Mobile Devices: A Survey on Network Traffic Analysis', Jun. 18, 2018, *arXiv*: arXiv:1708.03766. doi: 10.48550/arXiv.1708.03766.

[25]M. R. Joshi and T. H. Hadi, 'A Review of Network Traffic Analysis and Prediction Techniques'.

[26]A. D'Amelio, J. Lin, J.- Ramel, and R. Lanzarotti, 'Guest Editorial Emerging Trends and Advances in Graph-Based Methods and Applications', Jan. 2024, doi: 10.1109/TETC.2024.3374581.

[27]W. Jiang, 'Graph-based Deep Learning for Communication Networks: A Survey', *Comput. Commun.*, vol. 185, pp. 40–54, Mar. 2022, doi: 10.1016/j.comcom.2021.12.015.

Co-funded by the European Union

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

6GSNS

[28] T. Aittokallio and B. Schwikowski, ‘Graph-based methods for analysing networks in cell biology’, *Brief. Bioinform.*, vol. 7, no. 3, pp. 243–255, Sep. 2006, doi: 10.1093/bib/bbl022.

[29] W. Jiang and J. Luo, ‘Graph neural network for traffic forecasting: A survey’, *Expert Syst. Appl.*, vol. 207, p. 117921, Nov. 2022, doi: 10.1016/j.eswa.2022.117921.

[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, ‘A Comprehensive Survey on Graph Neural Networks’, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

[31] D. Andreoletti, S. Troia, F. Musumeci, S. Giordano, G. Maier, and M. Tornatore, ‘Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks’, in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 246–251. doi: 10.1109/INFCOMW.2019.8845132.

[32] Y. Li, R. Yu, C. Shahabi, and Y. Liu, ‘Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting’, Feb. 22, 2018, *arXiv*: arXiv:1707.01926. doi: 10.48550/arXiv.1707.01926.

[33] N. Sha, X. Wu, J. Wen, J. Li, and C. Li, ‘Adaptive spatio-temporal graph convolutional network with attention mechanism for mobile edge network traffic prediction’, *Clust. Comput.*, vol. 27, no. 9, pp. 13257–13272, Jun. 2024, doi: 10.1007/s10586-024-04577-2.

[34] J. Zhu *et al.*, ‘AST-GCN: Attribute-Augmented Spatiotemporal Graph Convolutional Network for Traffic Forecasting’, Nov. 22, 2020, *arXiv*: arXiv:2011.11004. doi: 10.48550/arXiv.2011.11004.

[35] S. Shleifer, C. McCreery, and V. Chitters, ‘Incrementally Improving Graph WaveNet Performance on Traffic Prediction’, Dec. 11, 2019, *arXiv*: arXiv:1912.07390. doi: 10.48550/arXiv.1912.07390.

[36] L. Bai, L. Yao, and C. Li, ‘Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting’.

[37] C. Zheng, X. Fan, C. Wang, and J. Qi, ‘GMAN: A Graph Multi-Attention Network for Traffic Prediction’, *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, Art. no. 01, Apr. 2020, doi: 10.1609/aaai.v34i01.5477.

[38] ‘The HDF5® Library & File Format’, The HDF Group - ensuring long-term access and usability of HDF data and supporting users of HDF technologies. Accessed: Dec. 03, 2024. [Online]. Available: https://www.hdfgroup.org/solutions/hdf5/

[39] S. Gil, P. H. Mikkelsen, C. Gomes, and P. G. Larsen, ‘Survey on open-source digital twin frameworks–A case study approach’, *Softw. Pract. Exp.*, vol. 54, no. 6, pp. 929–960, Jun. 2024, doi: 10.1002/spe.3305.

[40] ‘Co-design the Local Digital Twins (LDT) toolbox technical specifications for advancing the transformation of Smart Communities | Living in EU’. Accessed: Dec. 04, 2024. [Online]. Available: https://living-in.eu/news/co-design-local-digital-twins-ldt-toolbox-technical-specifications-advancing-transformation

Co-funded by the European Union

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

[41] 'Digital Twins Definition Language (DTDL)', opendigitaltwins-dtdl. Accessed: Dec. 04, 2024. [Online]. Available: https://azure.github.io/opendigitaltwins-dtdl/DTDL/v3/DTDL.v3.html

[42] A. Meijers, *Hands-On Azure Digital Twins: A practical guide to building distributed IoT solutions*. Packt Publishing Ltd, 2022.

[43] S. Karsten, 'Reference Architectural Model Industrie 4.0 (RAMI 4.0)'.

[44] 'Eclipse BaSyx | projects.eclipse.org'. Accessed: Nov. 21, 2024. [Online]. Available: https://projects.eclipse.org/projects/dt.basyx

[45] 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, p. 104007, Nov. 2023, doi: 10.1016/j.compind.2023.104007.

[46] U. Ahle and J. J. Hierro, 'FIWARE for Data Spaces', in *Designing Data Spaces : The Ecosystem Approach to Competitive Advantage*, B. Otto, M. ten Hompel, and S. Wrobel, Eds., Cham: Springer International Publishing, 2022, pp. 395–417. doi: 10.1007/978-3-030-93975-5_24.

[47] 'ScorpioBroker - Overview', GitHub. Accessed: Dec. 04, 2024. [Online]. Available: https://github.com/ScorpioBroker

[48] *GreyCat*. [Online]. Available: https://greycat.ai/

[49] T. Hartmann, F. Fouquet, A. Moawad, R. Rouvoy, and Y. L. Traon, 'GreyCat: Efficient What-If Analytics for Data in Motion at Scale', Mar. 23, 2018, *arXiv*: arXiv:1803.09627. doi: 10.48550/arXiv.1803.09627.

[50] F. Abbasi, P. Brimont, C. Pruski, and J.-S. Sottet, 'Understanding Semantic Drift in Model Driven Digital Twins', in *Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems*, in MODELS Companion '24. New York, NY, USA: Association for Computing Machinery, Oct. 2024, pp. 419–430. doi: 10.1145/3652620.3688256.

[51] 'GitHub - BESSER-PEARL/BESSER: A Python-based low-modeling low-code platform for smart software'. Accessed: Nov. 21, 2024. [Online]. Available: https://github.com/BESSER-PEARL/BESSER

[52] I. Alfonso *et al.*, 'Building BESSER: an open-source low-code platform', May 24, 2024, *arXiv*: arXiv:2405.13620. doi: 10.48550/arXiv.2405.13620.

[53] C. Sommer *et al.*, 'Veins: The open source vehicular network simulation framework', in *EAI/Springer Innovations in Communication and Computing*, Springer Science and Business Media B.V., 2019, pp. 215–252. doi: 10.1007/978-3-030-12842-5_6.

Co-funded by the European Union

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

6GSNS