Grant Agreement No.: 101192650

Type of action: HORIZON JU Research and Innovation Actions

Topic: HORIZON-JU-SNS-2024-STREAM-B-01-01

Call: HORIZON-JU-SNS-2024

D5.1 DEFINITION OF THE PRODUCT/SERVICE DEVELOPMENT LIFECYCLE MANAGEMENT IN THE INTEGRATED, HETEROGENOUS SBA-BASED NETWORKS AND IMPLEMENTATION OF THE DEDICATED SERVICES

Work package	WP 5
Task	Task Number
Due date	30/06/2026
Submission date	30/06/2026
Deliverable lead	FDI
Version	1.0
Dissemination Level	PU
Editor	Sotiris Spantideas (FDI)
Authors	Jorge Baranda (CTTC), Engin Zeydan (CTTC), Albert Bel (CTTC), Farhana Javed (CTTC), Luis Blanco (CTTC), Husnain Shahid (CTTC), Łukasz Kułacz (PUT), Adrian Kliks (PUT), Laura Finarelli (HESSO), Anastasios Giannopoulos (FDI), Marilina Bartsioka (FDI), Balaji Kirubakaran (DLR), Muhamad Luthfi (DLR). Harsh Manoj Shah (HOLO), Jetmir Haxhibeqiri (IMEC), Vasilis Maglogiannis (IMEC), Dries Naudts (IMEC), Jeroen Hoebeke (IMEC), Javier Velázquez Martínez (TID), Luis M. Contreras Murillo (TID), Maria A. Serrano (NBC), Efi Dvir (CER)

Reviewers	Gil Kedar (CER), Hua Wang (KEY), Gianluca Fontanesi (Nokia), Tomaso de Cola (DLR) Nikolaos Kanakaris (IPTO)

Abstract	It reports how the development lifecycle of products and services is managed in the context of SBA-based networks, detailing the interaction among the architectural components and includes information on the functionality and performance of dedicated intelligent applications such as xApps, rApps, dApps and how their integration into SBA can be utilised for resource, energy optimization, semantic-oriented communication, and Digital Twin (DT) services.
Keywords	Integrated 6G system, Service-Based Architecture (SBA), Service Based Management Architecture (SBMA), Service Development Lifecycle (SDLC), O-RAN, xHaul, Satellite, RIC, non-3GPP RAN, Integrated Services, Intelligent Applications, xApps, rApps, dApps, AI-driven models.

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	15/01/2026	1st version of the template for comments	FDI
V0.2	30/04/2026	1st round of draft contributions from partners	CTTC, PUT, IMEC, TID
V0.3	25/05/2026	2nd round of contributions	CTTC, PUT, DLR, TID, IMEC, HOLO, FDI, HESSO
V0.4	03/06/2026	Formatting and contribution consolidation	FDI, CTTC, PUT, DLR, TID, IMEC, HOLO, FDI, HESSO
V0.5	24/06/2026	Reviewers' comments	NOKIA, IPTO, DLR, ORA, KEY, CER
V0.6	29/06/2026	Addressed reviewers' comments	FDI, PUT, CER, CTTC, TID, IMEC, HOLO
V0.7	30/06/2026	Final version	FDI

DISCLAIMER

Co-funded by the European Union

6GSNS

Project funded by
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra
Swiss Confederation
Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

The **unity-6G** project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No *101192650*. This work has received funding from the Swiss State Secretariat for Education, Research, and Innovation (SERI).

COPYRIGHT NOTICE

EXECUTIVE SUMMARY

The present deliverable addresses the challenges arising from the highly heterogeneous 6G network ecosystem by defining the foundations for product and service development lifecycle management in integrated Service-Based Architecture (SBA)-based networks and by investigating the implementation of dedicated services for future 6G environments. The work builds upon the architectural foundations established in the project and focuses on service-oriented aspects, including service creation, deployment, operation, optimization, evolution, and decommissioning across integrated Open Radio Access Network (RAN), transport, Non-Public Network (NPN), and satellite/Non-Terrestrial Network (NTN) domains.

A key contribution of this deliverable is the definition of a Service Development Lifecycle (SDLC) framework tailored to cloud-native integrated 6G systems consisting of multiple networking domains. The proposed approach adopts microservice principles together with typical CI/CD and DevOps methodologies to enable automated and scalable management of network products and services. The lifecycle framework considers all service states and transitions, from design and validation to deployment, operation, suspension, and deletion, while supporting interoperability and sustainability objectives.

This deliverable further presents a Service Based Management Architecture (SBMA) framework for integrated heterogeneous network domains that facilitates communication between service consumers and service producers through common service exposure mechanisms, service registries, discovery functions relying on a dedicated communication bus. Particular attention is given to the integration of multiple technology domains and the realization of domain-specific architectural components and their required interaction with the inter-domain management entity to enable cross-domain, end-to-end business services that can leverage resources distributed across Open RAN, transport, mobile core, Non-Public Networks (NPN), and NTN infrastructures distributed across edge/cloud locations.

Finally, the document investigates a set of dedicated intelligent services and applications that can exploit the capabilities of the proposed SBA framework. These include advanced service concepts such as semantic-oriented communications and Digital Twin as a Service (DTaaS) that may utilise intelligent applications in the form of xApps, rApps, and potentially dApps for resource management, energy efficiency, and network optimization. The integration of these services demonstrates how intelligent, data-driven, and sustainable network operation can be achieved in future 6G systems.

In addition, the project KPIs that are associated with WP5 are identified and reported herein. Overall, the outcomes of this deliverable establish the initial framework for service lifecycle management and service orchestration within the UNITY-6G architecture, providing the basis for the implementation, integration, and evaluation of advanced 6G services in the subsequent phases of the project.

TABLE OF CONTENTS

1 INTRODUCTION 11
1.1 Background and motivation 11
1.2 Deliverable structure 12
1.3 Relation to other Work Packages 12
2 SERVICE DEVELOPMENT LIFECYCLE MANAGEMENT 14
2.1 Product/service development lifecycle management 14
2.1.1 Microservice approach 14
2.1.2 Continuous Integration / Continuous Deployment 15
2.1.3 Software Development Lifecycle Management solutions 16
2.2 Service definition in the unity-6g architecture 17
2.2.1 UNITY-6G architecture 17
2.2.2 Service Definition 18
2.3 Product/service development lifecycle and management 21
3 SERVICE BASED ARCHITECTURE (SBA) 28
3.1 Service Based Architecture (SBA) for integrated heterogeneous networks 28
3.2 Motivation for service-based architecture 29
3.2.1 The need for Service-Based Architecture in beyond 5G networks 29
3.2.2 SBA evolution paths 30
3.3 Components of the service based architecture in UNITY-6G 34
3.3.1 Cross-Domain Service Based Management Bus 34
3.3.2 Inter-Domain Management Orchestrator (IDMO) 35
3.3.3 Domain Components 37
4 INTEGRATED SERVICES 40
4.1 Definition of an Integrated service 40
4.2 List of Integrated services 41
4.2.1 Emergency Network Service 41
4.2.2 XR Semantic Service 43
4.2.3 Digital Twin for Integrated 6G Network 46
4.2.4 Multi-RAT Time-Sensitive Service 48
4.3 Integration of intelligent applications 49
4.3.1 App-01: Change Scheduler 49
4.3.2 App-02: Failure Prediction 55
4.3.3 App-03: Wireless xHaul Reconfiguration 57
4.3.4 App-04: Energy-aware load balancing 60
4.3.5 App-05: XR Application 61

4.3.6 App-06: Multi-RAT TSN optimization 63
4.3.7 App-07: Traffic engineering for TN-NTN switching 64
4.3.8 App-08: Semantic Domain Switching for XR Streaming over TN-NTN Networks 66
4.3.9 App-09: Probabilistic forecasting app in TN-NTN 69
4.3.10 App-10: Conflict Management between Open RAN and IDMO 70
4.3.11 App-11: Trust Modelling to enable trustworthy interactions between IDMO/DMO and intra DMO components using DLT 70
4.3.12 App-12: LLM based monitoring app for O-RAN 70
4.3.13 App-13: xApp for KPI monitoring 72
4.3.14 App-14: Traffic steering and policy-based load balancing 74
4.3.15 App-15: Radio Access Technology Monitoring 75
4.4 Integration and testing of components for services and applications 76
5 CONCLUSIONS 78

LIST OF FIGURES

FIGURE 2-1: CONCEPTUAL ILLUSTRATION OF THE CI/CD/CS PIPELINES IN UNITY-6G .. 15
FIGURE 2-2: SOFTWARE AND SERVICE DEVELOPMENT LIFECYCLE 17
FIGURE 2-3: UNITY-6G ARCHITECTURE, AS PRESENTED IN [3]. 17
FIGURE 2-4: EXAMPLES OF DOMAIN-SPECIFIC SERVICES 19
FIGURE 2-5: EXAMPLES OF BUSINESS END-TO-END SERVICES 20
FIGURE 2-6: EXAMPLE OF SERVICE LAYERS AND COMPLEX SERVICES IN THE UNITY6G ARCHITECTURE. 21
FIGURE 2-7: SERVICE DEVELOPMENT LIFECYCLE..... 22
FIGURE 2-8: SECOND STAGE VERIFICATION DURING A SERVICE DEVELOPMENT USING THE DIGITAL TWIN. 23
FIGURE 2-9: BUSINESS END-TO-END SERVICE DEPLOYMENT..... 25
FIGURE 2-10: STATE TRANSITIONS OF A SERVICE IN UNITY-6G. 27
FIGURE 3-1: 6G-CLOUD ARCHITECTURE PROPOSAL RAN-CN CONVERGENCE OF SBA (EXTRACTED FROM [20]) 31
FIGURE 3-2: OVERALL ARCHITECTURE ENVISIONED BY ORIGAMI (EXTRACTED FROM [23])..... 32
FIGURE 3-3: THE 6GREEN SBA ARCHITECTURE AND ITS COMPONENTS (NFS) FRAMEWORK (EXTRACTED FROM [25])..... 33
FIGURE 3-4: HEXA-II CONSIDERED RAN-CN CONTROL PLANE OPTIONS FOR 6G (OPTION A-C)..... 33
FIGURE 3-5: IDMO ARCHITECTURE 36
FIGURE 4-1: INTEGRATED SERVICES AND ASSOCIATED INTELLIGENT APPLICATIONS IN UNITY-6G 41
FIGURE 4-2: USE CASE WITH A TWO-OPERATOR CONFLICT 51
FIGURE 4-3: CHANGE SCHEDULER ARCHITECTURE 52
FIGURE 4-4: FAILURE PREDICTION 56
FIGURE 4-5: SGW CONCEPT 65
FIGURE 4-6: SEMANTIC-AWARE O-RAN UNITY-6G ARCHITECTURE FOR TN-NTN..... 67
FIGURE 4-7: LLM-BASED MONITORING FOR O-RAN..... 71
FIGURE 4-8: IMPLEMENTATION CONSIDERATIONS OF THE LLM-BASED MONITORING APP 72
FIGURE 4-9: KPI MONITORING XAPP IN AN INTEGRATED O-RAN NTN-TN ARCHITECTURE 73

LIST OF TABLES

TABLE 1: SEMANTIC DOMAIN SWITCHING STATES AND HANDOVER TRIGGERS. 68
TABLE 2: KPIS LINKED WITH WP5. 82

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
AMF	Access and Mobility Management Function
AoI	Age of Information
CFS	Customer Facing Service
CI/CD	Continuous Integration/Continuous Deployment
CI/CD/CT	CI/CD /Continuous Testing
CI/CD/CS	CI/CD/Continuous Supervision
CNF	Cloud-Native Network Functions
conMLO	Continuous Multi Link Operation
CQI	Channel Quality Indicator
CP	Control Plane
CSI	channel state information
CUPS	Control and User-Plane Separation
DevOps	Development Operations
DLT	Distributed Ledger Technology
DMO	Domain Manager and Orchestrator
DT	Digital Twin
ENI	Experiential Network Intelligence
ENIF	ENI Function
GSBA	Global Service Based Architecture
IDM	Infrastructure Domain Manager
IDMO	Inter-Domain Management Orchestrator
IP	Internet Protocol
ISR	Intent State Registry
KPI	Key Performance Indicator
KVI	Key Value Indicator
LLM	Large Language Model
MANO	Management and Orchestration
MLOps	Machine Learning Operations
MnS	Management Services
MCS	Modulation and Coding Scheme
MQTT	Message Queuing Telemetry Transport
NF	Network Functions
NFC	Network Function Chain
NFV	Network Functions Virtualisation
NFVO	NFV Orchestrator
NoN	Network of Networks
NRF	Network Repository Function
NTN	Non-Terrestrial Networks
OAM	Operation and Management
ODA	Open Digital Architecture
O-RAN	Open Radio Access Network
PoP	Point of Presence
PRB	Physical Resource Block
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RIC	RAN Intelligent Controller
RFS	Resource Facing Service
SA	Service Adapter
SBA	Service Based Architecture

SBI	Service Based Interface
SBMA	Service-based Management Architecture
SDLC	Service Development Lifecycle
SGW	Smart Gateway
SLA	Service Level Agreement
SLO	Service Level Objectives
SMB	Service Management Bus
SMF	Session Management Function
SMO	Service Management and Orchestration
SNR	Signal to Noise Ratio
SI	Semantic Information
SO	Service Orchestrator
SSR	Service State Registry
TB	Transport Block
TCP	Transmission Control Protocol
TN	Terrestrial Networks
TS	Technical Specification
UDR	Unified Data Repository
UP	User Plane
UPF	User Plane Function
VPN	Virtual Private Networks
ZMQ	Zero Message Queueing

1 INTRODUCTION

The evolution towards Sixth Generation (6G) networks is characterized by an unprecedented level of integration among heterogeneous communication domains, including Open Radio Access Networks (O-RAN), transport networks, Non-Public Networks (NPNs), satellite and Non-Terrestrial Networks (NTNs), IEEE 802.11 Wi-Fi, edge/cloud infrastructures, and AI-native management systems. Unlike previous generations of mobile networks, where network functions and services were often deployed in relatively isolated environments, 6G systems are expected to provide a unified service platform capable of supporting diverse vertical sectors, stringent performance requirements, and highly dynamic operational conditions. This evolution requires a fundamental shift from infrastructure-centric management towards service-centric and software-driven network operation based on Service-Based Architecture (SBA) principles.

Within the UNITY-6G framework, WP5 complements the architectural and technological developments conducted in the project by focusing on services and applications that exploit the capabilities of the integrated 6G infrastructure. The main objective is to establish the foundations for service creation, deployment, operation, optimization, and deletion in a multi-domain environment where network functions, applications, and intelligent agents are dynamically combined to support end-to-end business services. In this context, the concepts of cloud-native design, microservice decomposition, continuous integration and continuous deployment (CI/CD), DevOps/MLOps methodologies, and AI-assisted orchestration become essential enablers for the efficient management of products and services throughout their lifecycle.

1.1 BACKGROUND AND MOTIVATION

Towards this direction, the present deliverable presents the methodology and mechanisms required to manage the complete development and operational lifecycle of products and services in future SBA-based 6G environments. Particular emphasis is placed on the Service Development Lifecycle (SDLC), covering all phases from service conception and design to implementation, testing, validation, deployment, operation, scaling, suspension, upgrade, and decommissioning. The deliverable investigates how lifecycle management principles can be applied consistently across multiple domains while ensuring interoperability, automation, resilience, sustainability, and efficient resource utilization.

In addition, the document introduces a service-based architectural framework designed to support the seamless integration of heterogeneous network domains and service components. The proposed SBA relies on standardized service exposure mechanisms, service discovery procedures, service registries, dedicated service buses, and interoperable communication interfaces that enable efficient interactions among service producers, consumers, and orchestration platforms. The architecture is designed to support the dynamic creation and composition of domain-specific, cross-domain, and end-to-end business services while accommodating the flexibility and scalability requirements expected from future 6G ecosystems.

Building upon the lifecycle management framework and the SBA principles, the deliverable further investigates the implementation of dedicated services and applications that exploit the capabilities of integrated networks. Particular attention is given to intelligent applications such as xApps, rApps, and potentially dApps, as well as intelligent applications deployed in the transport, Wi-Fi and NTN domains that provide advanced functionalities for resource allocation, energy efficiency, network optimization, and autonomous management. The intelligent applications named cApps are recently extended to the Core domain, targeting AI-native core with self-configuration, self-optimization, self-healing, and self-protection functionalities [1]. Furthermore, innovative service concepts including semantic-oriented communications and Digital Twin as a Service (DTaaS) are explored, both of which are considered key enablers for sustainable and intelligent network operation. These services leverage distributed intelligence, data-driven decision making, and real-time network awareness to improve network efficiency while supporting the diverse requirements of future applications and vertical industries.

Concluding, the outcomes presented in this deliverable contribute to the realization of a flexible, programmable, and sustainable 6G service ecosystem capable of supporting integrated Open RAN, transport, NPN, and satellite environments. By defining the service development lifecycle, establishing the underlying service-based architecture, and demonstrating the implementation of dedicated intelligent services, this work provides the foundation for future service innovation and operational automation within the UNITY-6G architecture

1.2 DELIVERABLE STRUCTURE

The structure of this deliverable is as follows: Section 2 provides a state-of-the-art (SOTA) analysis of the service development lifecycle, focusing on the definition of the product/service in the context of the UNITY-6G system and how its lifecycle management can be realized in the integrated architecture. Section 3 presents SOTA considerations related to the SBA architecture originating from 5G system, as well relevant work that has been conducted by other SNS-JU projects. In addition, the interaction and communication principles between the different service components (e.g., Open RAN, transport, NTN) in order to provide end-to-end networking solutions is detailed via a dedicated cross-domain service-based management bus. Furthermore, Section 4 defines the relevant integrated services for the project (Emergency network service, XR semantics service, etc.) and outlines how several intelligent applications developed in other WPs can be integrated in the UNITY-6G system. Finally, Section 5 concludes the deliverable, providing closing remarks for the upcoming WP5 work.

1.3 RELATION TO OTHER WORK PACKAGES

The work presented herein, as well as the overall WP5 takes inputs from several other WPs. In specific, the identification of Key Performance Indicators and Key Value Indicators (KPIs/KVIs) measurement methodology presented in Deliverable D2.1 [1] as well as the first version of the UNITY-6G architecture illustrated in Deliverable D2.3 [3] are utilized for the definition of the integrated services and the service-based architecture. In addition, WP5 runs in parallel with WP3 that provides key implementation of the networking components (O-RAN,

NTN, IEEE 802.11, etc.) and the digital twin through D3.1 [4]. Moreover, WP4 provides the majority of the intelligent applications that are presented in this deliverable from the “service” perspective, while their algorithmic fundamentals (optimization problems, ML models, etc.) are discussed in D4.1 [5]. Finally, concrete implementation of the services and applications that are developed in WP5 will be showcased in the PoCs of WP6, providing a roadmap between development of the integrated services and demonstration in use case scenarios.

2 SERVICE DEVELOPMENT LIFECYCLE MANAGEMENT

In SBA-based systems, such as the UNITY-6G architecture, there is a need to precisely define what kind of services are considered in the network architecture, as well as the SDLC. To this end, this section defines the different variants of UNITY-6G services and then describes how the activation of the new product/service can be performed, along with additional management actions, i.e., deployment, pausing, suspending, or closing.

2.1 PRODUCT/SERVICE DEVELOPMENT LIFECYCLE MANAGEMENT

The current state-of-the-art for managing complex network services is characterized by a transition from hardware-dependent configurations to fully virtualized, intent-driven architectures. While 5G successfully introduced the SBA within the Core Network, the 6G landscape, particularly within the scope of the UNITY-6G project demands an extension of these principles across highly heterogeneous domains, including O-RAN, NTN, and Transport (xHaul).

Traditional management frameworks, such as the initial ETSI NFV-MANO (Network Function Virtualisation – Management and Orchestration), were designed for the virtualization of individual network functions (VNFs) rather than end-to-end service lifecycles [6]. While ETSI has evolved through initiatives like Experiential Network Intelligence (ENI) to incorporate AI-driven automation, these systems often lack the native cross-domain coordination required for 6G. Similarly, 3GPP has standardized 5G network slicing and resource management [7], yet existing solutions frequently encounter scalability limitations when attempting to handle the massive influx of devices and data characteristic of integrated 6G environments.

Contemporary research increasingly focuses on Intent-Based Management, a concept championed by TM Forum through its Open Digital Architecture (ODA) [8]. This paradigm aims to replace manual technical configurations with high-level service requests that may be expressed in natural language. However, a significant gap remains in the literature regarding a comprehensive end-to-end (E2E) framework that reliably translates these intents into technical parameters across multiple administrative domains. Furthermore, the emergence of Digital Twin (DT) technology is becoming a cornerstone of future network evaluation. State-of-the-art DT solutions, such as those discussed in recent academic literature (e.g., Colosseum as a Digital Twin), provide isolated testbeds for wireless emulation but are not yet fully integrated into the standard service lifecycle to proactively evaluate performance before actual service instantiation [9].

2.1.1 Microservice approach

In the context of the UNITY-6G project, the microservices approach is the fundamental architectural pillar that enables a shift from rigid, monolithic network management to a flexible SBA. This paradigm is primarily inspired by the 5G Core design but is extended in UNITY-6G to encompass the entire integrated network, including O-RAN, NTN, and WiFi domains. The architecture is built upon several key microservice-oriented design principles.

In this context, each of the 6G domains include Network Functions (NFs) that are designed as independent microservices that communicate in most cases via standardized, web-based RESTful APIs. Following the 5G Core paradigm, each NF represents a discrete functionality, such as session management, policy control, or data storage, that can be independently consumed by authorized Network Function Consumers or even 3rd-party applications. This avoids traditional point-to-point or monolithic interdependencies, allowing Network Services that are composed of multiple NFs or Network Function Chains (NFCs) to be modular and interoperable. Therefore, as new services or domains (like a specific O-RAN instantiation) are activated, they register their interfaces with a Service Registry through a Service Bus, allowing the system to automatically discover and incorporate them into end-to-end (E2E) service chains. Because NFs are decoupled into microservices, the Service Orchestrator can scale specific components of the Network Service (e.g., a User Plane Function (UPF) in a specific cluster) without affecting the rest of the network. In addition, the use of open APIs and microservice patterns facilitate the integration of components from multiple operators, avoiding vendor-lock services. Finally, the microservice approach directly supports the SDLC, where individual services can be updated, restarted, or paused independently, enabling "canary deployments" and rapid updates without total network downtime.

2.1.2 Continuous Integration / Continuous Deployment

In the framework of UNITY-6G, the application of Continuous Integration and Continuous Deployment (CI/CD) represents a paradigm shift from traditional software release cycles to a highly automated, network-aware orchestration process. While the state-of-the-art in standard cloud-native environments focuses on the velocity of code shipment, 6G requirements demand that CI/CD pipelines be intrinsically linked to the underlying network state, physical constraints (especially in O-RAN and NTN), and strict Service Level Agreements (SLAs). Current research and industry standards, particularly within the O-RAN Alliance and ETSI ZSM (Zero-touch network and Service Management), advocate for a "DevOps-for-Telco" approach. This involves the automation of the entire software delivery pipeline, from the initial commit of a Radio Intelligence Controller (RIC) rApp, xApp or dApp to its deployment at the edge. However, existing solutions often struggle with the heterogeneity of integrated terrestrial and non-terrestrial components, where deployment environments are not uniform.

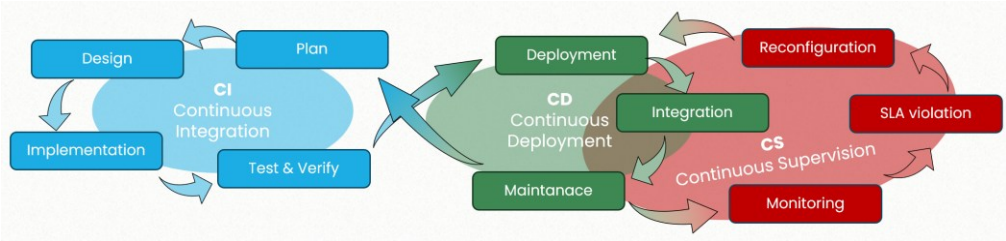

Figure 2-1: Conceptual illustration of the CI/CD/CS pipelines in UNITY-6G

In the UNITY-6G context, CI/CD is being evolved into a more robust framework often referred to as CI/CD/CT (Continuous Testing) or CI/CD/CS (Continuous Supervision). Beyond standard unit testing, CI in 6G requires rigorous validation of service descriptions and technical requirements. Before any component is deemed "ready," it must be verified against a Service Registry to ensure it can correctly expose its capabilities via the Service Bus. This phase includes semantic checks to ensure that the microservices can interoperate within the multi-

vendor environment of the UNITY-6G architecture. The "Deployment" phase in the project is not a simple container push. It involves complex dependencies where a service (e.g., an E2E localization service) might depend on several other sub-services being in an "Operational Status OK" state. The state-of-the-art involves using Helm charts and Kubernetes operators, but UNITY-6G extends this by incorporating the Service Orchestrator to handle domain-specific constraints, ensuring that deployment only proceeds when all cross-domain dependencies (O-RAN, Core, and NTN) are synchronized.

The most significant advancement in the 6G CI/CD state-of-the-art is the closing of the loop between deployment and real-time operations. Standard CD ends once the software is "running," but in the SBMA, the pipeline remains active, enabling the monitoring of the deployed service. Hence, the Continuous Supervision (CS) phase monitors the health and performance of the deployed service, as conceptually shown in Figure 2-1. If the Digital Twin or the Intent Handler detects a degradation in performance or a violation of the original intent (e.g., latency exceeding the 50ms threshold), the CI/CD pipeline can trigger an automated rollback or a re-instantiation of the service with adjusted resource parameters. This ensures that the lifecycle management is not a linear path but a circular, self-healing process.

By integrating these automated pipelines, UNITY-6G moves toward a "Zero-Touch" operational model where the time-to-market for new 6G services is minimized, and the stability of the integrated network is maintained through constant, automated validation.

2.1.3 Software Development Lifecycle Management solutions

When we talk about Software Development LifeCycle (SDLC) Management, we are moving beyond simple "software updates" into a realm where the network itself is treated as a dynamic set of living services. In the world of 6G, the sheer complexity of integrating networks means we can no longer rely on manual configuration. With O-RAN bringing thousands of micro-components (xApps/rApps), human engineers cannot manually manage the lifecycle of every single function. To achieve the 6G vision, the network must be able to self-configure, self-optimize, and self-heal. SDLC management provides the framework for this "zero-touch" automation.

In UNITY-6G, we distinguish between two types of lifecycles to ensure the network stays organized, as depicted in Figure 2-2:

- Software Development LifeCycle (SoftwareDLC) - “build” phase - which focuses on the code itself, i.e., writing code, testing, creating containers, Helm charts, and software packages. This phase contains wrapping functions into lightweight containers (like Docker). Moreover, every piece of code is automatically validated against 6G performance KPIs before it ever touches the production network, also using a DT environment.
- Service Development LifeCycle (ServiceDLC) - “run” phase - which focuses on the operational life of the service, i.e., how it is created, deployed, paused, or terminated across the network. Instead of telling the network how to set up a service, we tell it what we want (e.g., "I need a low-latency slice for an ambulance"), and the orchestrator handles the rest. This involves the constant monitoring of "Service Health." If a service degrades, the SDLC framework can automatically restart or migrate it to a different node.

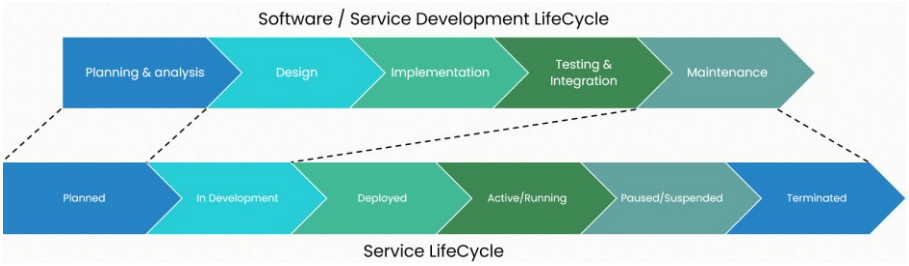

Figure 2-2: Software and Service Development Lifecycle

2.2 SERVICE DEFINITION IN THE UNITY-6G ARCHITECTURE

This subsection presents an overview of the UNITY-6G architecture and provides the definitions of the "services" that we consider in the UNITY-6G architecture compared to typical network services that consist of one or more NFs.

2.2.1 UNITY-6G architecture

The updated version of the UNITY-6G architecture is shown in Figure 2-3 illustrating the different networking domains that aim to offer integrated services, as well as the Inter-Domain Management Orchestrator (IDMO) and the cross-domain SBMA that provides the interconnection between the different domains. In addition, the Data continuum layer is shown (used to transfer data between the domains), the Trust Layer that offers trustworthiness among the multiple domains, as well as the exposure layer, providing interfaces and monitoring capabilities to external stakeholders and vertical application providers.

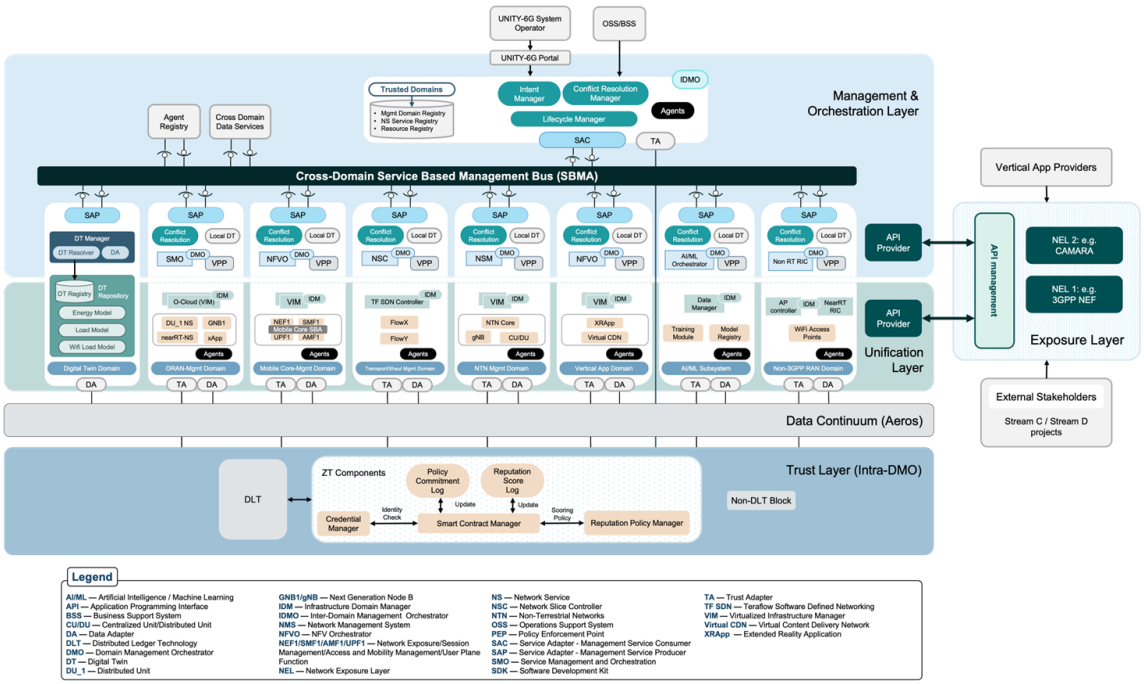

Figure 2-3: UNITY-6G architecture, as presented in [3].

The IDMO is virtually the "Brain" of the architecture. It is responsible for the end-to-end lifecycle management of services. It receives high-level intents and translates them into technical configurations. Using created technical configurations, registered services and domains capabilities it creates and manages services between different domains. It continuously monitors telemetry from the network to perform self-optimization and self-healing.

The SBMA is the "Artery" of the system. It facilitates the discovery and communication between decoupled services, or between domains and IDMO in general. It allows any domain to announce its presence and capabilities to the system. As shown in Figure 2-3, different types of domains communicate with the IDMO, i.e.:

- O-RAN (exposing RAN, xApps/rApps features, near-RT RIC) domain
- NTN domain
- Mobile Core domain
- Transport Network domain
- WiFi and Non-3GPP RAN networks
- AI/ML subsystem
- Vertical application domain
- Digital Twin domain

2.2.2 Service Definition

In UNITY-6G architecture, the network services that can be provided from a single domain (e.g., simple network services), as well as more complex integrated services can be coordinated that span across multiple domains. Finally, business services can be accessed by the management layer.

A service in the UNITY-6G ecosystem is defined as a modular, programmable capability delivered over a heterogeneous communication network to provide specific functionality to end-users, vertical industries, or other internal network services. Service provides functionality (specific capability), defined as VNF or set of VNFs, by exposing it through standardized API to another VNFs (consumers). Service can be combined from multiple services creating complex service.

Each service communicates with IDMO through management service bus. Service can be registered and discovered by the service registry (to be available for other entities). Every service is managed through a SDLC that separates the software build (CI/CD) from the operational runtime management. Services are not manually "turned on"; they are triggered by Intents (high-level requirements, e.g., "ultra-low latency for remote surgery") that the IDMO translates into specific service set and configurations. Services include telemetry hooks that allow the IDMO to monitor health and automatically scale, migrate, or restart the service to maintain SLAs without human intervention. Services can be deployed at any Computing Point of Presence (PoP), from localized Far-Edge nodes to centralized data centers, depending on latency and throughput requirements.

2.2.2.1 Domain-specific services

VNFs inside a single domain may be combined to provide a domain service. Therefore, domain-specific services refer to services instantiated and managed within a single network

domain, where the required VNFs, Cloud-Native Network Functions (CNFs), or intelligent applications belong to the same functional scope. These services expose specialized capabilities of a domain independently from other network segments and are typically optimized according to the operational objectives, interfaces, and performance constraints of that domain. Examples include Control Plane (CP) and User Plane (UP) services provided by the core network, gNB and Near-Real-Time RAN Intelligent Controller (Near-RT RIC) services provided by the RAN domain, or AI model hosting and inference services provided by an AI subsystem.

Domain-specific services constitute the foundational building blocks of the overall UNITY-6G architectural design. Within each domain, VNFs/CNFs and intelligent applications may be chained or orchestrated to realize a higher-level domain capability. For example, a RAN service may combine gNB functions (RU, DU, CU), Near-RT RIC components, and xApps/rApps to enable radio optimization and policy enforcement, while a core-network service may integrate mobility management, session management, and user-plane functions to deliver connectivity and traffic steering. These services are typically exposed through standardized APIs and orchestration interfaces to enable interoperability and composition with services from other domains.

Some service examples offered by specific domains include (as shown in Figure 2-4):

- Control Plane (CP) by core network
- User Plane (UP) by core network
- gNB by RAN network
- Near-RT RIC by RAN network
- xApp/rApp by RAN network
- AI model by AI domain

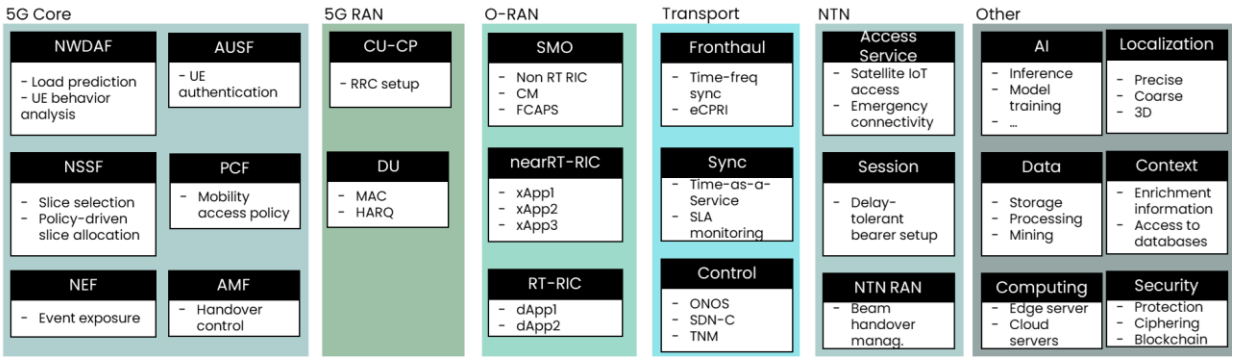

Figure 2-4: Examples of domain-specific services

2.2.2.2 Cross-Domain services

VNFs across domains may be also combined to provide a cross-domain service. These services refer to integrated services composed of functions originating from multiple network domains, such as RAN, core, transport, edge, NTN, or AI domains. In this context, VNFs/CNFs and intelligent applications distributed across different administrative or technological domains are orchestrated together to deliver a unified network capability. Each Domain Management Orchestrator (DMO) is responsible for orchestrating the NFs inside the domain, i.e., RU-DU-CU and Near-RT RIC in the case of O-RAN and AMF, PCF, UPF, etc. in the case of Core. The DMOs expose their capabilities to the IMDO. Cross-domain services enable tighter

coordination among network segments and facilitate end-to-end operational optimization beyond the boundaries of individual domains. Typical examples include RAN-core integration services, AI-assisted mobility management across RAN and transport domains, NTN-TN service continuity mechanisms, or even AI-assisted handover across RAN technologies (e.g., between cellular and Wi-Fi domains).

The realization of cross-domain services requires interoperable orchestration, service exposure, lifecycle management, and policy coordination mechanisms across heterogeneous infrastructures. Such services may involve distributed control loops, shared telemetry, intent-driven orchestration, and AI-assisted decision-making to ensure consistent Quality of Service (QoS), reliability, and scalability across domains. By combining capabilities from multiple domains, cross-domain services support advanced 6G requirements such as ultra-low latency, dynamic service adaptation, seamless mobility, and integrated terrestrial/non-terrestrial connectivity.

2.2.2.3 Business End-to-end services

The domain services can be combined into complex business services, representing end-to-end service offerings exposed to vertical industries, enterprises, or end users, abstracting the complexity of the underlying network infrastructure and domain interactions. These services are outcome-oriented and are defined according to business or application requirements rather than individual network functions. A business service may therefore consume and orchestrate capabilities from transport, RAN, core, edge, NTN, cloud, and AI-native domains to satisfy specific service-level objectives, as shown in Figure 2-5. Examples include end-to-end time-critical connectivity for robotics, immersive extended reality (XR) services, autonomous mobility support, or resilient industrial automation services.

From an implementation perspective, business services are realized through the composition of multiple domain-specific and cross-domain services into a unified service framework governed by intent-based orchestration and AI-native automation. The orchestration system dynamically selects, configures, and scales the required VNFs/CNFs, transport resources, AI functions, and edge computing capabilities according to application requirements, network conditions, and business policies. This abstraction enables service providers to expose flexible, programmable, and SLA-driven service offerings while maintaining operational efficiency and enabling rapid deployment of innovative 6G use cases.

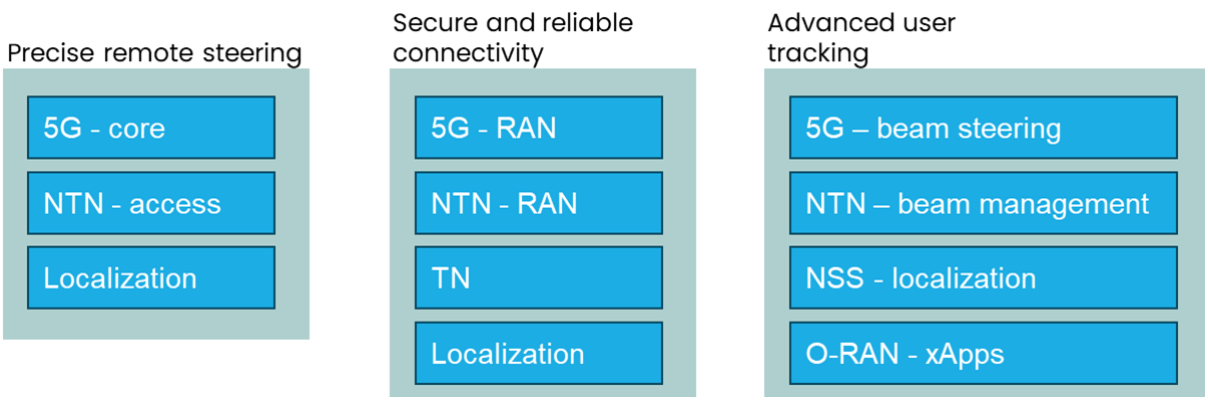

Figure 2-5: Examples of business end-to-end services

In Figure 2-6 , an example of the described above concepts of services and complex services placed in the architecture of Unity6G is illustrated. A basic domain service that uses one or more VNFs can be further combined into more complex services. All services intended to be used on controller in the cross-domain system are exposed through common interfaces, and then further can be combined into business services.

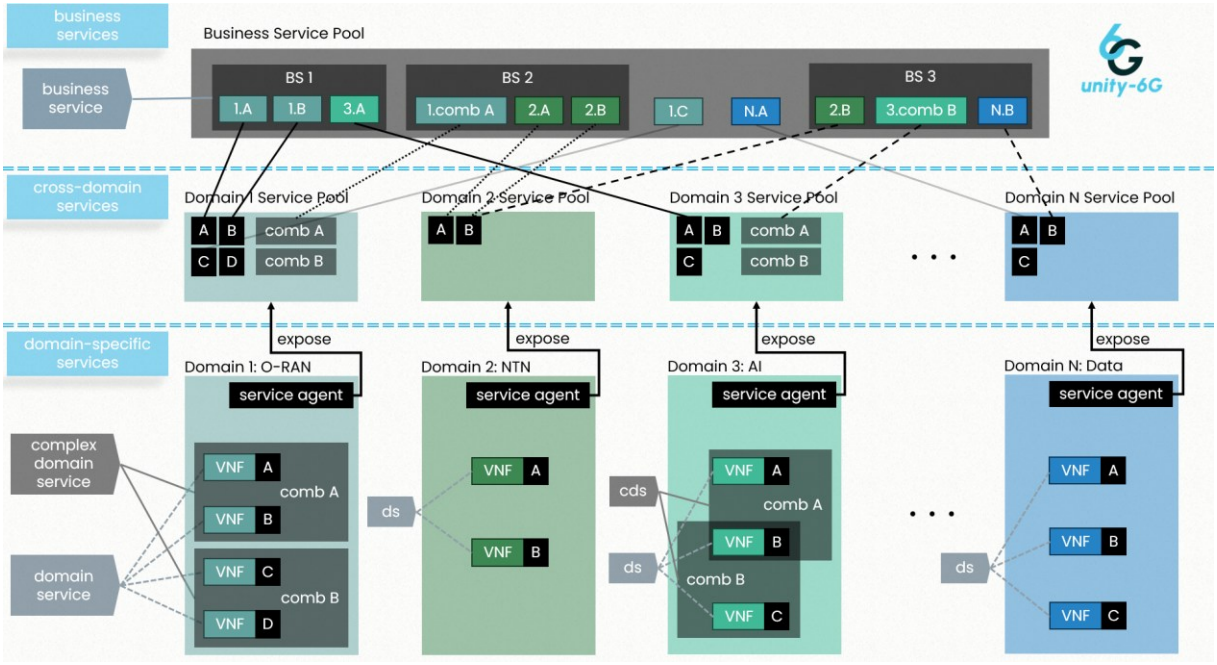

Figure 2-6: Example of service layers and complex services in the Unity6G architecture.

2.3 PRODUCT/SERVICE DEVELOPMENT LIFECYCLE AND MANAGEMENT

This subsection details how a service is developed and deployed in the integrated, heterogenous network: from the planning & analysis phase to the design phase, implementation, testing & integration and maintenance.

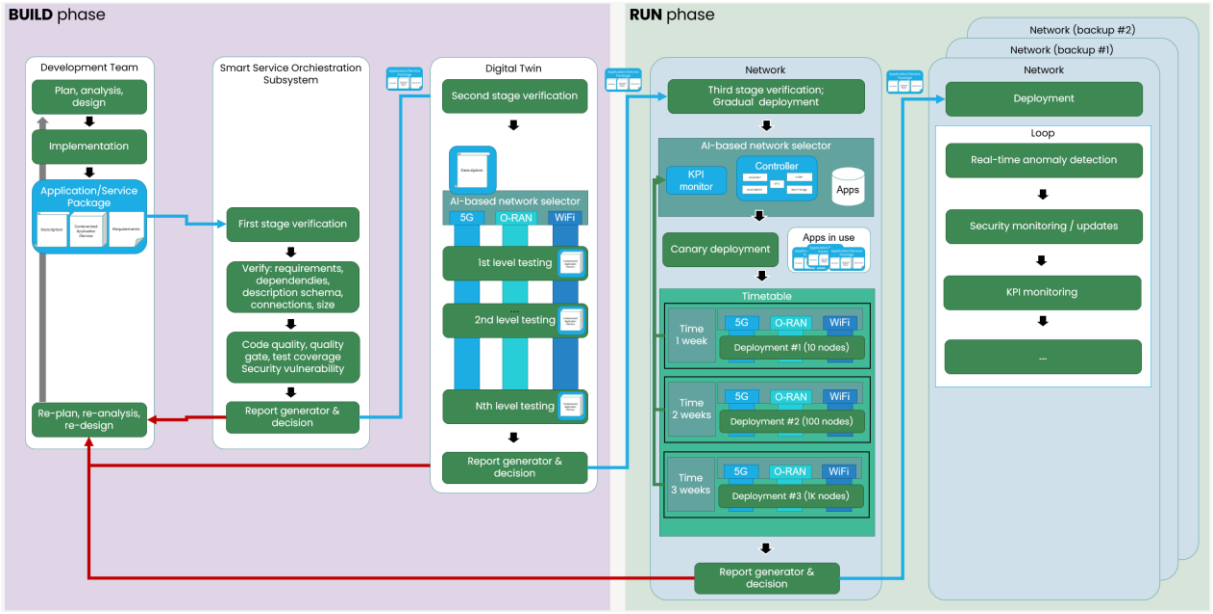

Figure 2-7: Service development lifecycle.

The SDLC process is detailed in Figure 2-7 and follows the following list of steps (grouped by different entities of step execution):

- Steps executed by a development team
    - *Plan, analysis, design* – a development team (i.e. xApp developers) that wants to deploy an intelligent application or add into the network a new service (i.e. energy saving feature offered as additional option for O-RAN cells) need to plan and design this product in form of compatible application (i.e. xApp, compatible with specific O-RAN version, protocol versions, etc.). The team needs to also take into consideration all requirements and dependencies (e.g. availability of different services, their versions, parameters, etc.).
    - *Implementation* – the development team after planning phase can start implementation of the product.
    - *Packaging* – As a result of the implementation the product should be packaged in the standardized form (for UNITY-6G services) containing containerized application (e.g. docker image), list of requirements and dependencies (services and their versions, domains, resources, etc.), and descriptor (used by the IDMO). This package (result) has to be delivered to the IDMO (Smart Service Orchestration Subsystem – in the figure above).
    - *Problem Detection* – In case of any problem detection (at any phase of deployment), a deployment report with a description of the problem will be delivered to the development team. After introducing changes in the package, it can be transferred to the IDMO again (re-upload).
- Steps executed by the IDMO (Smart Service Orchestration Subsystem)

- *First stage verification* – a received package is verified in terms of required elements (missing elements, fields of descriptor, unclear or incomprehensible values), dependencies (allowed values), size, etc.
- *Initial Testing* – then a containerized application is tested against security vulnerability, code quality, quality gate, test coverage, etc.
- *Package Transfer/Problem Notification* – When the minimal requirements (e.g., minimal set of descriptor fields filled, containerized application included) for the service are met, the whole package is transferred further to the IDMO module responsible for Digital Twin Management. Otherwise, a report about criteria that were not met is transferred back to the development team.

- Steps executed by the Digital Twin Manager

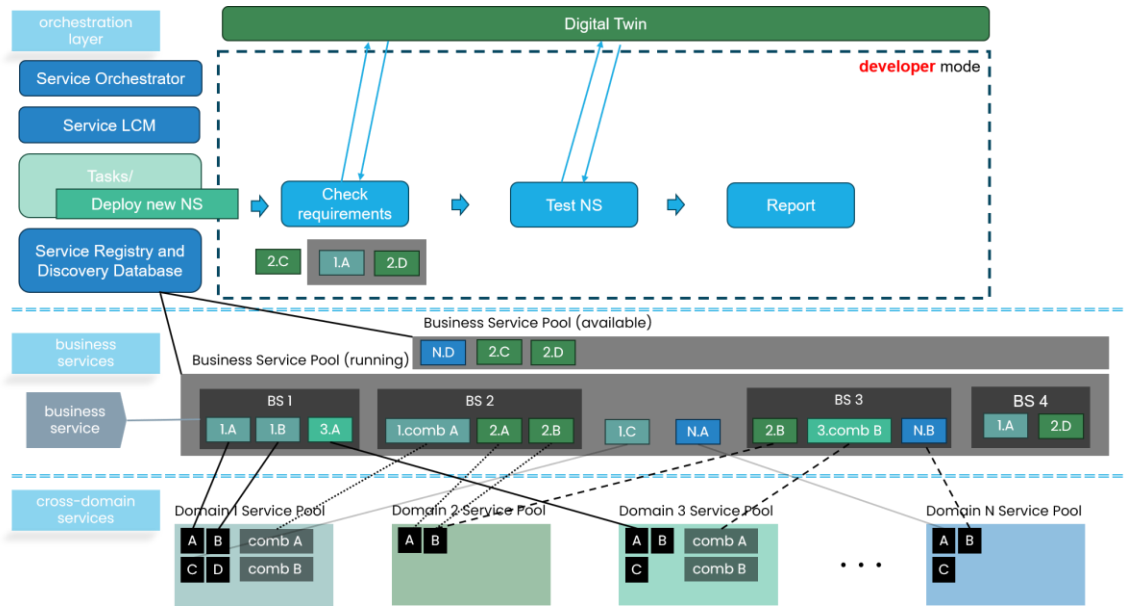

Figure 2-8: Second stage verification during a service development using the digital twin.

- *Second stage verification* – AI-based network selector prepares available digital twins for a set of configurations that are required to test the service. The AI-based network selector needs to use domains that are required by the service and additionally evaluate if the service could influence another domain. The testing of the service within digital twin should be gradual, i.e. starting with the basic and quick configuration that confirms no errors during service deployment, through longer simulations of the service operation, towards the test of the whole network operation. During each test campaign, all errors should be collected, as well as KPIs of tested service (if they are not met) and other services running in the network (if their KPIs are not met). The process is depicted in Figure 2-8. Passing the package to Digital Twin Manager is an optional step that should be considered based on the availability of the Digital Twin functionality. Such an additional verification phase should be very beneficial; however, this is not an obligatory step for each package and implementation.

- *Package Transfer for Deployment* – when all tested scenarios finish successfully, the service package is transferred further to deployment in the network. This procedure is coordinated by the IDMO. In case of any problems, errors, etc., a report about the test procedure results is transferred back to the development team.

- Steps executed by the IDMO (Service Lifecycle Manager)
  - *Ready for deployment* – From this moment, the service is tested and ready for deployment (registered in the form of capabilities). However, actual deployment of the service will start only if the specific conditions are met (as depicted in Figure 2-9):
    - An intent will be received, that after intent processing and translation the service will be selected to be deployed.
    - Any change in the network (emergency, failure of another services, alerts observer by monitoring entity, etc.) causes the change of service deployment that will include deployment of the service.
  - *Deployment* – When the above criteria are met, the AI-based network selector will decide where and when to deploy the service in the real network. This procedure should follow canary deployment rules, to minimize potential failure of the network. Deployed services should be monitored, i.e., whether it is running normally and if KPIs targets related to this service are met. The IDMO manages the lifecycle of the service, so can decide about its pausing (e.g. for temporary resource optimization), termination (e.g. when the service is no longer needed, their dependencies are no longer met, etc.), modification (e.g. due to update of dependent service).
  - *Service monitoring* – The IDMO keeps tracking the network deployments (sets of domains, services with their configurations, etc.) to allow quick backup in case of failure.

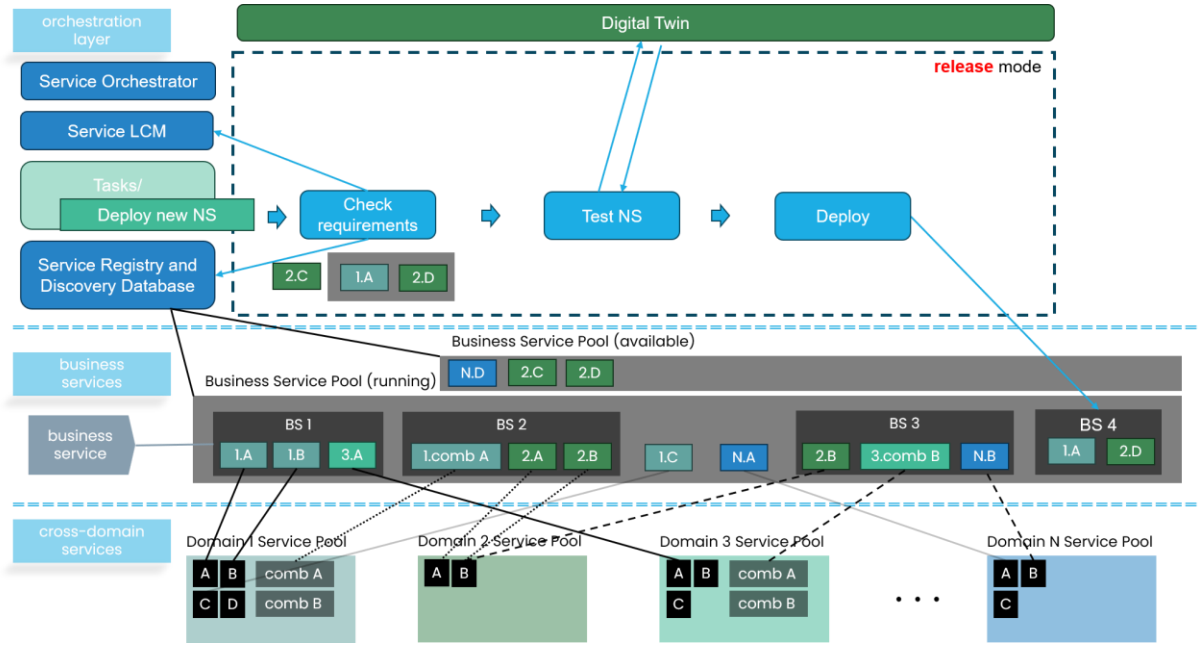

Figure 2-9: Business end-to-end service deployment.

The lifecycle represents the states a service transitions through (*activation, running, pausing, suspending*, or *closing*), as shown in Figure 2-10, managed by the IDMO via the Service Management Bus (SMB). To enable the services to transition smoothly between these states and to be deployed, operated, and adapted across multiple domains the proposed architecture comes along with lifecycle management capabilities. Under this scope, a service is not treated as a static entity. Instead, it is considered a dynamic object whose state evolves according to operational requirements, user demands, resource availability, network conditions, and business policies. Throughout its lifetime, a service may require resource allocation, performance monitoring, scaling, reconfiguration, temporary deactivation, recovery from failures, or complete removal from the infrastructure. To address these needs, IDMO supervises the possible states and transitions and ensures SLAs are continuously fulfilled by interacting with domain-specific orchestration and management functions and triggering actions whenever required.

- *Activation* – The IDMO receives a high-level intent and translates it into technical requirements. It then reserves resources across the selected domains, instantiates the necessary VNFs, and establishes the end-to-end connection. Procedures can be also initialized by different events than receiving intent, e.g. failure of another service initializes the procedure to find alternative solution. A solution can therefore consider the following service, and in that case (without received intent) activation of the service will begin. A service needs to follow a list of rules to be considered for activation:
    - The service description contains all data required for the selected service type.
    - All fields, values and units can be translated into unified template (service definition)
    - Description contains information about author's, organization's, version's, dependencies, and interfaces.

- *Running* – the service is in the running phase. After service activation, it will remain in this phase. This phase involves Closed-Loop Assurance. The system monitors telemetry data and if the performance / KPIs drops, the IDMO may trigger an action for service reconfiguration in order to maintain the SLA. The IDMO may also pause, terminate, suspend a service based on the service state (monitoring), received intent, or other events observed in the network:
    - The service can be paused if the IDMO will temporarily decide that it is needed for resource optimization.
    - Relevant tasks (service deployment request, update, scaling, etc.) should be received by the IDMO
    - All requirements in the moment of deployment should be met (CPU, memory, location, delay, power, etc.)
    - All dependencies should be available and running in the versions specified in Technical Service Specification and related requirements met
    - There is no conflict between services
- *Pausing* – This is a temporary freeze of the service. The service configuration remains in the system, but data flow is halted to conserve resources or energy. This is common in "Green 6G" scenarios where resources are throttled during low-demand periods. Resuming the service from this state should be very quick. Typical scenarios for pausing the service include:
    - No users / requests uses service for longer than defined value X1 (to save resources)
    - Priority of this service is the lowest and higher priority service is needed in this location / part of network
    - A conflict between the service and other services was detected
    - Observation of external triggers that should move to running state
    - Temporarily (short time) low priority requirements are not met
- *Suspending* – A deeper state than pausing, often triggered by a priority of conflict or failure. The IDMO may suspend a low-priority service to reallocate resources, e.g. to a mission-critical emergency service. Resuming the service from this state takes more time (as some resources are dismissed due to suspending) than from pausing. This action occurs in the following typical circumstances:
    - No users / requests use service for longer than defined value (to save resources)
    - Priority of this service is the lowest and higher priority service is needed in this location / part of network along with resources
    - Observation of critical external triggers that should move to running state

- *Closing* – The decommissioning phase. IDMO releases the resources and shuts down the instances. Resuming from the closing state is not possible; the service should be activated from scratch once again when will be needed. The closing of the service happens in scenarios where:
    - Relevant tasks were detected (e.g., service no longer needed, end of service planned time, etc.)
    - Health check service failure or Health check failure (by X times) in one of the critical requirements (depending on service).

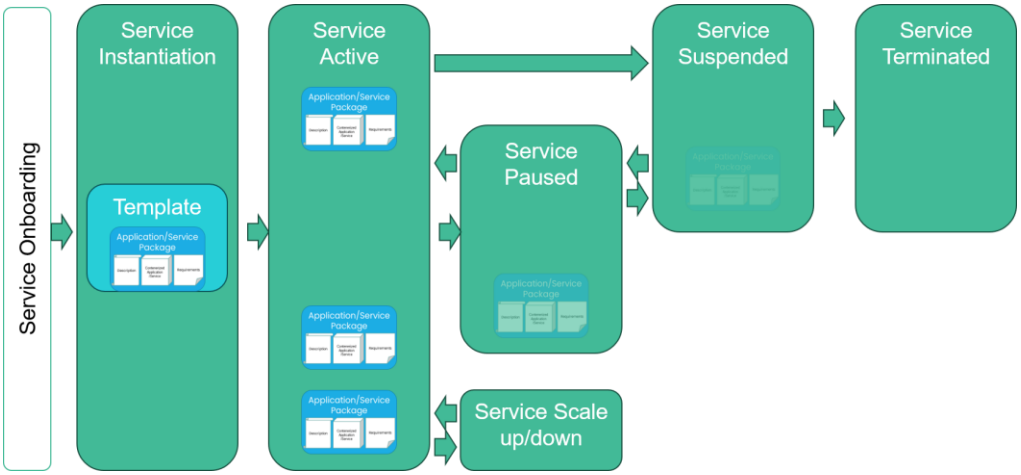

Figure 2-10: State transitions of a service in UNITY-6G.

3 SERVICE BASED ARCHITECTURE (SBA)

One of the focus points of the UNITY-6G architecture is the development of an SBA approach capable of integrating different components such as Open RAN, transport, and NTN based on satellite systems to provide end-to-end networking solutions. In conjunction with WP2, this work targets a reliable and scalable architecture with the aim of accommodating the dynamic and diverse requirements of services in integrated networks. The SBA will be designed to facilitate communication between the consumer pool and the specific produce pool to allocate resources to run mobile network entities providing communication services through the use of a dedicated cross-domain service-based management bus. To achieve this objective, the consumer component, the IDMO, needs to account for a service registry and discovery database to provide seamless distributed deployment and integration of services. The messaging exchange and communication flow between different elements of the SBA will also be defined to ensure smooth functioning and efficient management of integrated network.

3.1 SERVICE BASED ARCHITECTURE (SBA) FOR INTEGRATED HETEROGENEOUS NETWORKS

The first time that SBA made an official appearance, in the context of the 5G Core Network, was in 3GPP TS 23.501 Release 15 [8][1], while its development has been done within 3GPP SA2 strand. This was a fundamental shift from monolithic designs toward modular, service-oriented, multi-interface network architectures. Unlike previous generations (2G through 4G) which relied on "Reference Point" architectures, with specific interfaces defined for every pair of communicating elements, the SBA is developed around a Service based Interface (SBI) bus. This allows any authorized NF to expose its capabilities as reusable services through standardized SBIs. In other words, it enables NFs to expose their services via a common communication framework, thus replacing traditional point-to-point reference interfaces with a service-oriented model. In this framework, each NF (like the AMF for mobility or SMF for session management) can act both as a “Service Producer” and a “Service Consumer”. They register their capabilities with a Network Repository Function (NRF), which acts as a central repository. When one function needs to communicate with another, it "discovers" the target service via the NRF and interacts with it using a Request-Response or Subscribe-Notify model. These rely on standardized web technologies and protocols, including HTTP/2, RESTful APIs, and JSON [11], [12].

Apart from the Core Network, service-based principles are used also in the Management and Orchestration domain, as defined by 3GPP TS 28.533 [13]. Multiple management services act in a similar way with NSs to expose their management functionalities and interact with other management entities. This allows reusability in different operation workflows and decouples management logic from the underlying infrastructure. As a result, SBMA extends the SBA concept from the control plane to the management plane, enabling scalable and flexible orchestration mechanisms required for complex multi-domain environments.

In a similar way, research initiatives, extend the SBA footprint even more, by introducing a Network of Networks (NoN) paradigm where SBA adheres to multiple domains – such as Radio Access Network (RAN), Transport network (TN), and Non-Terrestrial Network (NTN) – of

integrated and heterogeneous network environments. This also underpins network slicing, where logical networks are dynamically created by selecting and combining appropriate NF instances with the aim of meeting specific service requirements. This evolution reflects the growing need for scalable, interoperable, and flexible architecture able to support dynamic service composition and orchestration across diverse domains.

Another advancement that comes in complete alignment with SBA is O-RAN, as it concerns a disaggregated and service-oriented RAN architecture by default. In particular, O-RAN defines functional components such as the Service Management and Orchestration (SMO) framework and the RAN Intelligent Controllers (RICs), which expose capabilities through standardized interfaces and APIs, enabling modular interaction between control, management, and optimization functions. In addition, the interfaces support not only communication but also integration of application (e.g., xApps and rApps) that consume and/or provide services within the RAN ecosystem. All these align with the SBA philosophy by enabling programmability, interoperability, and dynamic control of network functions through standardized service exposure mechanisms [14].

SBA continues to evolve as a result of the increasing complexity of modern 5G/6G systems. In this context, NFs are incorporated into cloud-native and microservice-based implementations in a way that can be deployed and scale independently. Another trend is associated with the adoption of event-driven interaction models which enable responsive and dynamic service execution as well as efficient network management. Research also identifies that web-based technologies are used more and more in order to align telecom systems with modern software engineering practices.

On the other side, some open challenges need to be outlined too. The main issue that restricts the full establishment of SBA is the large number of interacted devices which leads to complex systems and many points of added overhead. In addition, multiple domains may fit in dynamic environments but raise concerns about interoperability if different vendors do not comply with common protocols. The same applies to lifecycle management and orchestration. Instantiation, scaling, modification, and termination of network functions and slices is an attracting field of service-based interactions, but requires coordination and end-to-end standardized mechanisms across the different domains.

3.2 MOTIVATION FOR SERVICE-BASED ARCHITECTURE

3.2.1 The need for Service-Based Architecture in beyond 5G networks

Through the SBA approach, the mobile core enabled a further development of control and user-plane separation (CUPS) principle proposed as evolution of 4G LTE networks and a cloud-native transition, moving from virtualised software running former hardware functions to cloud-native artefacts exploiting the disaggregated NFs and providing flexibility on the deployment and management capabilities.

More specifically, the transition to SBA was not merely a technical preference but a necessity driven by the requirements of next generation mobile networks. In this transition, the following main aspects can be highlighted:

- *Scalability and Flexibility*: Traditional point-to-point architectures were unflexible. Adding a new feature often required updating multiple interfaces across the entire network. SBA allows operators to deploy or update a single NF independently. This loose coupling ensures that the network can scale dynamically based on demand - for example, spinning up additional mobility management or UPF instances during a major public event without affecting other services. Furthermore, many NFs are designed to be stateless, storing user data in a separate Unified Data Repository (UDR). This makes the network more resilient; if a function fails, a new one can take over instantly without losing the session context.
- *Support for Network Slicing*: One of the headline features of 5G is Network Slicing, which allows a single physical infrastructure to be partitioned into multiple virtual networks tailored for specific uses (e.g., Ultra-Reliable Low-Latency Communications for autonomous vehicles vs. Massive IoT for smart cities). SBA's modularity is what makes this possible, as different "slices" can select and combine only the specific NFs instances as needed and perform distributed deployments to meet requirements (e.g., latency) demanded by vertical stakeholders.
- *Faster Time-to-Market:* By using standard web protocols (HTTP/2, JSON, REST), the telecom industry has embraced industry IT and software management paradigms. This allows developers to use common DevOps tools and CI/CD pipelines. Consequently, new services that used to take months to integrate can now be deployed in days or even hours.
- *Multi-Vendor Interoperability:* SBA reduces vendor lock-in. Since the interfaces are standardized APIs rather than proprietary hardware connections, operators can "mix and match" Network Functions from different vendors more easily, thus reducing vendor lock-in and fostering a more competitive and innovative ecosystem.

3.2.2 SBA evolution paths

The 6G vision adopts the "network of networks" (NoN) concept, integrating multiple domains into a highly sustainable and scalable AI-native architecture designed to address diverse network requirements, as proposed in European flagship Hexa-X project [15]. Under the NoN paradigm, the SBA is not just a software pattern for the mobile core, but also a holistic philosophy. This view considers not only the management of traditional domains (e.g., (O-)RAN, Core Network, transport) but also the integration of additional heterogenous domains, like NTNs, non-3GPP access (i.e. IEEE Wi-Fi). This management must be highly distributed and interoperable, as the requested network services are highly dynamic, accommodating the QoS needs of different tenants and allocating varying network resources, which have to be consumed following the same architectural rules as the mobile core. Due to the relevance of the SBA approach, different HE-SNS JU European research projects in previous calls (e.g., Call 1 and Call 2 in 2022 and 2023, respectively) have been developing this concept and proposing architectures for shaping the management, orchestration and deployment of 6G networks.

6G-BRICKS [16] project proposes a decentralized Management plane via a unified framework of Domain Managers and orchestrators (DMOs) performing per site management and analytics collections, going beyond the centralized model of the 5G SBA via the NWDAF function. Other projects like 6G-CLOUD [17] proposes the extension of the SBA concept towards the RAN domain, that is a service-based RAN, which has been also proposed in the literature [18], [19]. In 6G-CLOUD, they go beyond this initial approach dealing with RAN by designing a RAN

control fabric design and defining mechanisms for inter-NF communication to achieve the RAN-core convergence, as depicted in Figure 3-1. The 6G-CLOUD project also proposes connected service bus interfaces to be also exploited by AI and exposure network functions.

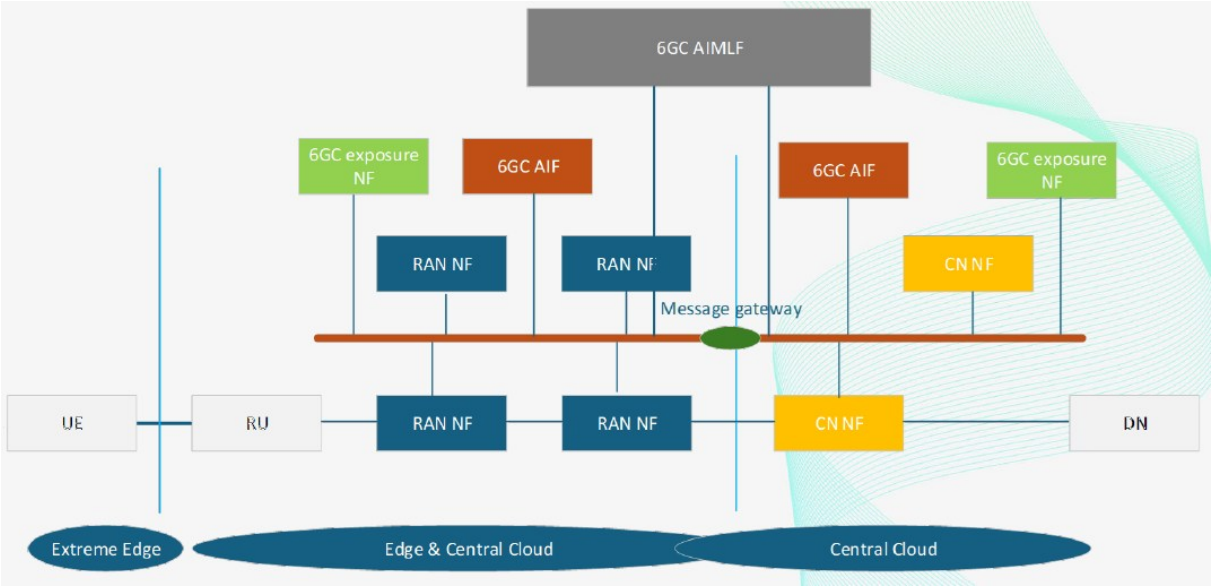

Figure 3-1: 6G-CLOUD architecture proposal RAN-CN convergence of SBA (extracted from [20])

In the same direction, the HE-SNS JU ORIGAMI project [21] proposes the Global SBA (GSBA), extending the service-based architecture principle not only across all network domains – including the RAN- but also even across operators. According to ORIGAMI [22], the GBSA provides a unified service "bus" that links formerly siloed components, making the entire network more flexible, programmable, and easy to orchestrate. This modularity and orchestrated control are key to dynamic resource allocation and, by extension, energy efficiency and sustainability. Such idea is depicted in Figure 3-2. According to ORIGAMI, the extension/inclusion of SBA into the RAN is expected to bring several benefits to the network operations and management, such as improved scalability, enhanced modularity, improved automation, higher resilience and futureproofing for long-term evolving technologies.

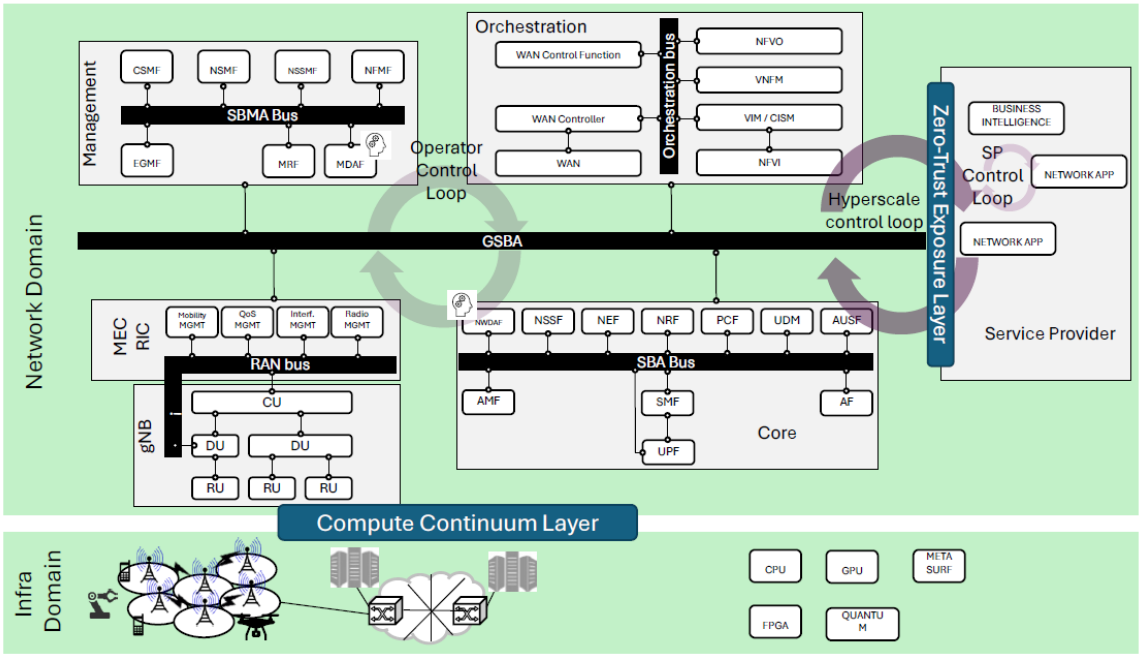

Figure 3-2: Overall architecture envisioned by ORIGAMI (extracted from [23])

Other European project that advocates for the extension of SBA into RAN is 6Green [24], as well as defining new NFs extending the original idea of SBA. The proposed architecture, organised into 5 frameworks, as depicted in Figure 3-3, also adds energy-aware logic to the management and orchestration of the SBA. Relying on AI-based services, they can trigger Zero-Touch operations within the extended SBA framework. 6Green describes in [25] how their proposed architecture, extending SBA, can reduce the overall energy and carbon footprint induced in the infrastructure, which is combined with separate cognitive and automated decision-making in the global network and/or within a part of the network, respectively, carried out by the ENI Function (ENIF), a strategic network function that has been developed and conforms to the ETSI ENI framework.

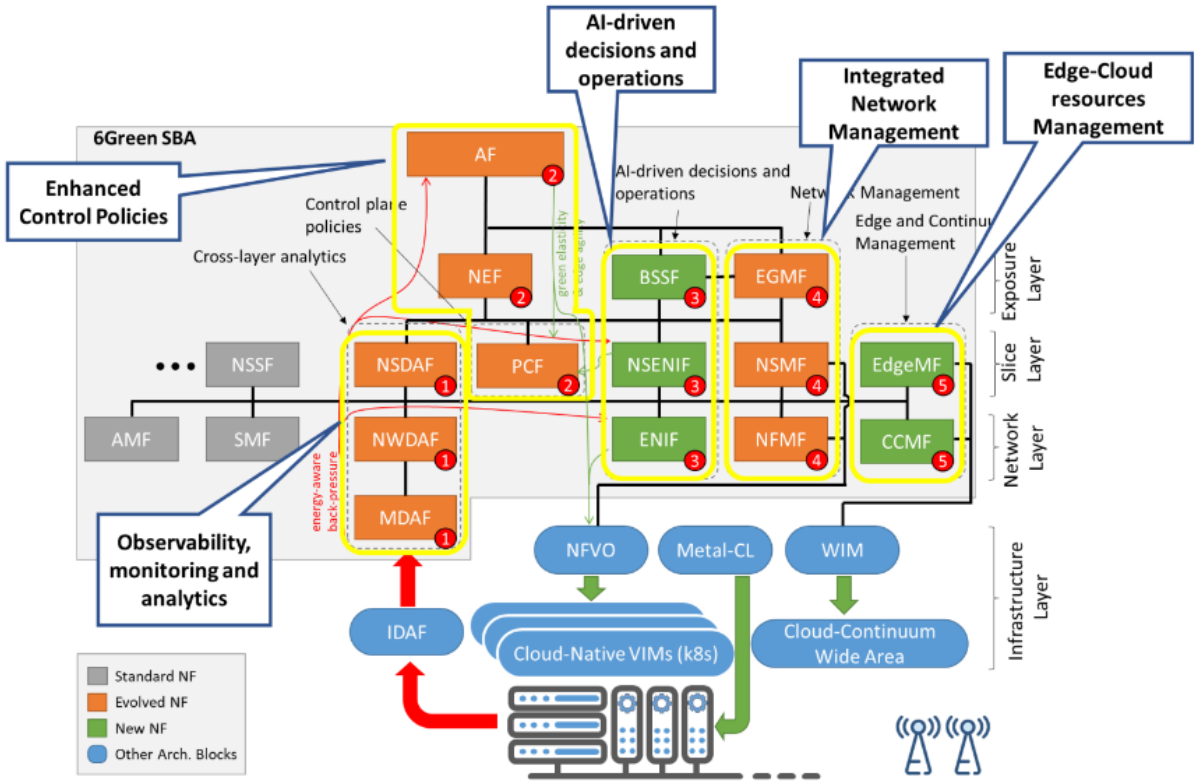

Figure 3-3: The 6Green SBA architecture and its components (NFs) framework (extracted from [25])

Hexa-X-II project [26] provides a detailed analysis of the extension of SBA into RAN in [27]. The approach proposed by 6G-CLOUD and ORIGAMI projects corresponds to what is called Option B in Hexa-X-II. This project also considers Option C, a hybrid approach where service-based interaction is introduced with minimal impact on the existing point-to-point interactions and without creating duplication among the two interfaces, as depicted in Figure 3-4.

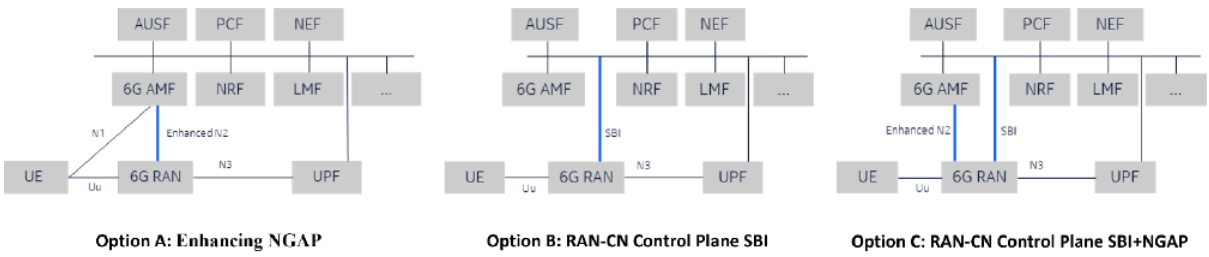

Figure 3-4: HEXA-II Considered RAN-CN control plane options for 6G (Option A-C)

However, Hexa-X-II concludes that Option A, which only consists of the enhancement of N2 interface, that is not adopting *"full SBA approach"* for the RAN segment, is the preferred option because it can handle separate domains, implies similar effort in implementation and

standardization compared to 5G and the other options, less testing and verification efforts and less effort for handling coordination and race conditions.

Considering this observation of Hexa-X-II, UNITY-6G architecture adopts the SBA concept from the perspective considered in 3GPP SA5 [13], as pointed out in [18], which UNITY-6G project considers to be more aligned with the 6G NoN concept presented previously. Covering Operation and Management (OAM) aspects, the SBMA comprises Management Services (MnS). Thus, multiple heterogenous domains, traditional ones beyond RAN, core, transport can be integrated through the definition of abstractions mechanisms and common interfaces to enable communication and data exchange between them. The UNITY-6G approach, presented in the next section, aims at unifying different stakeholders owning independent administrative domains in an end-to-end fashion, which can scale in terms of domains and service instances, compared to the approaches described previously.

3.3 COMPONENTS OF THE SERVICE BASED ARCHITECTURE IN UNITY-6G

This section presents the architecture proposed by UNITY-6G leveraging the SBMA approach [13] to create a central coordination layer to perform the unified cross-domain service orchestration and intelligent management in multi-domain environments. This architecture, whose principles and architectural layers have been described in UNITY-6G deliverable D2.3 [3], aims at a reliable and scalable architecture that can accommodate the dynamic and diverse requirements of services in integrated networks. According to 3GPP TS 28.533 [13], there are MnS producers and consumers. In this case, an MnS provided by an MnS producer can be consumed by any MnS consumer entity with appropriate authorisation and authentication. An MnS producer offers services via a standardised service interface composed of individually specified MnS components. In UNITY-6G, the IDMO is the MnS consumer of the MnS services provided by the different domains, which enable the construction and lifecycle management of complex management services and functions using the composition and interoperation of more basic components, i.e., through the available Domain Management Orchestrators (DMOs) available at the different domains. IDMO and Domains communicate through the cross-domain service-based management bus. The following subsections focus on the description of the main components included in UNITY-6G architecture build around the SBMA approach, and which can be identified in Figure 2-3.

3.3.1 Cross-Domain Service Based Management Bus

The cross-domain service-based management bus is the substrate providing the logical connection between the different domains (service producers) and the IDMO (service consumer).

Through this bus, the different domains can register at the IDMO, so they can be part of the UNITY-6G system. Information objects (e.g., JSON-formatted messages) are exchanged to populate IDMO database modules by announcing the domain capabilities. This information includes the service catalogues available at each domain, the computing resource availability (i.e., CPU, RAM, Storage), potential geographical footprint, transport network interconnection points and available service level objectives (SLOs) for transport network paths. This information is refreshed based on system operations, so IDMO can select the appropriate

domains fulfilling the demands of service requests. Lifecycle management operations (e.g., instantiation, termination) are also exchanged between IDMO and the different domains through the cross-domain SBMA bus.

From a practical perspective, the implementation and its performance will depend on the scenario, expected features (i.e., scale, target domain, latency, etc.) and focus (messaging, cache and memory, event streaming). Based on that, various solutions can be considered, including Apache Kafka, gRPC, Apache Pulsar, RabbitMQ, Zero Message Queueing (ZMQ), Redis, Message Queuing Telemetry Transport (MQTT). From all of them, UNITY-6G considers Apache Kafka as a suitable option due to its trade-off between features, complexity and performance.

Next, we present the organization of the different message topics defined in the Apache Kafka instance to organize and implement the communication of domains and IDMO through the cross-domain service-based management bus. The main principles are:

A) a single well-known registration topic (request/response) to ensure a single entry point to join the system: this is the topic where a domain states its willingness to join the UNITY-6G system by announcing an abstraction of its capabilities in terms of resources (CPU, RAM, storage) and the information about the catalogue of MnS services it is able to deploy, as well as the connections it has towards neighbouring transport domains. The request message should include a temporal ID so previous registered domains can discard the answer from IDMO. The response message includes the ID given by IDMO to this domain and the topics used for dedicated communication with this domain (next).
B) per-domain topic for bidirectional communication: as explained before, IDMO will assign an identifier to the registering domain and will create a pair of topics to exchange dedicated communication like heart-beat messages (for domain liveness) and resource update. The subscription at IDMO to this topic should be removed and the topic removed if the heart-beat signal is loss for a period of time (implementation-wise defined). This topic will also be used to send instantiation messages to the given domain. In this instantiation message there will be also the information for the per-service topic (next).
C) per-request/service topic: this topic will be created at each domain involved in the creation of a service request upon the reception of the instantiation command of the sub-service component. During lifetime of a service, this topic can be used for the exchange of information among domains or for synchronization (e.g., service termination request). This topic and its subscriptions disappear when the service is terminated.

3.3.2 Inter-Domain Management Orchestrator (IDMO)

IDMO serves as the central intelligence of the UNITY-6G orchestration framework, collecting intents from users to coordinate the different domains and enabling unified end-to-end orchestration and coordination among the heterogeneous domains distributed across the edge-cloud continuum. Figure 3-5 presents a diagram of the envisaged main basic orchestration components of the IDMO and its relationship with the cross-domain SBMA and domains elements. It is organized into two primary layers connected by an Internal bus, the Intent Management Layer and the Service Management Layer. IDMO contacts with the cross-domain service-based management bus through the Service Adapter component.

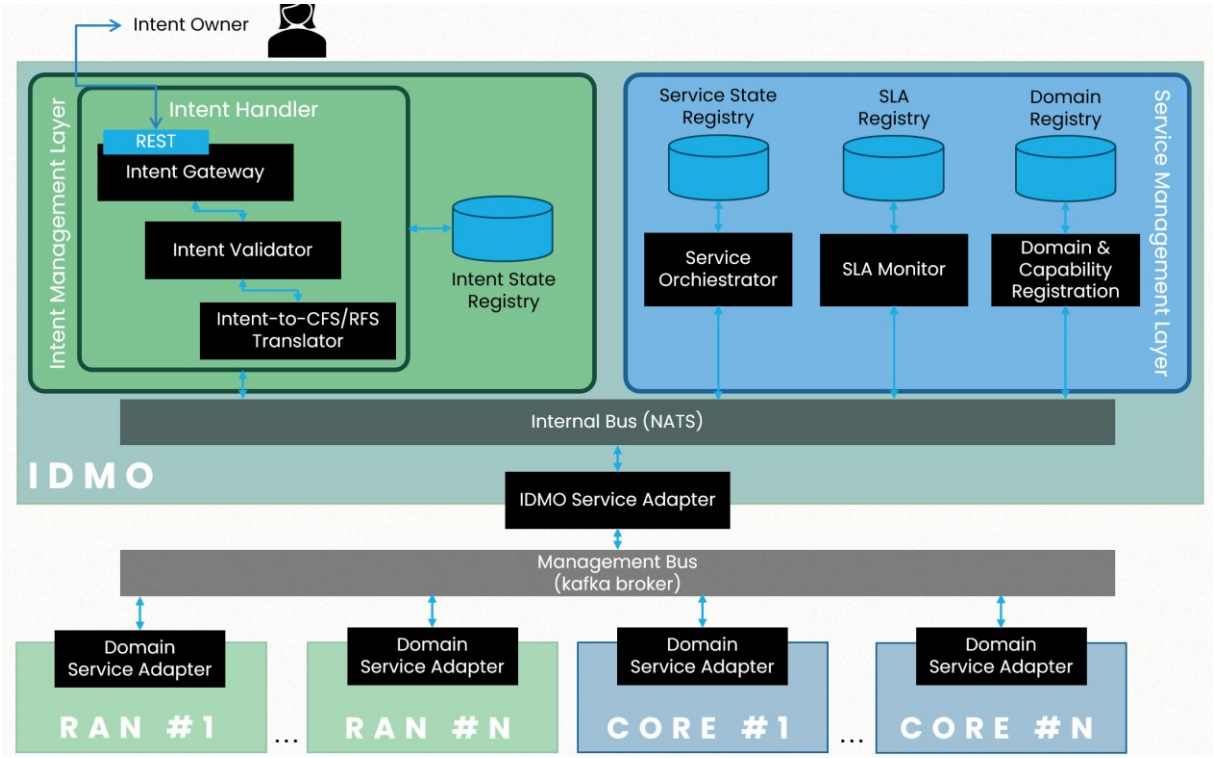

Figure 3-5: IDMO Architecture

3.3.2.1 Intent Management Layer

This layer acts as the Northbound interface for the UNITY-6G user, also known as Intent Owner. He/she expresses intents according to 3GPP standards [33], where expectations are defined including goals and constraints for a specific service (e.g., mobile network deployment) is specified. These intents are expressed to the Intent Gateway, a front end chatbot that interacts with a commercial Large Language Model (LLM)-agent based to clarify customer needs through the interaction with the intent owner, understand and validate the intent, and handle potential conflicts and business rule violations with the Intent Validator. These intents, following TM Forum approach [34], are translated first into a Customer Facing Service (CFS) template and then later, translated into different Resource Facing Service (RFS) templates, one for each involved domain (e.g., mobile core, RAN).

CFS and RFS approach is a modelling concept that allow to decouple high-level customer services, technology-agnostic focusing on customer-perceivable attributes (CFS) to derive different technical components mapping to specific networks domains relating to physical or logical resources that will be exchanged with the Service Management Layer to perform the domain selection. Their implementation will be based on service ordering (tmf641 [35]), resource ordering (tmf652 [36]) specifications from TM Forum. The intent management layer records the transactions for a given intent in the Intent State Registry (ISR), which is also aligned with the result of the Service Management Layer.

3.3.2.2 Service Management Layer

This is the core operational layer where the orchestration and autonomous management takes place. The Service Orchestrator (SO) module functions as the central intelligence, acting as the service lifecycle manager coordinating the different domains. It considers the “Saga

Pattern" to ensure transaction integrity across the distributed domains by managing the intent/service lifecycle management through the operational states defined in 3GPP: preparation, commissioning, operation and decommissioning. This activity is registered in the Service State Registry (SSR) database. It coordinates the intent realisation from the different derived RFSs by first validating the feasibility across selected domains, and only upon total success, proceeds to dispatch provisioning commands via the IDMO Service adapter (SA). Upon initiating this deployment, the SO dynamically provisions a dedicated Kafka Service Topic for that specific intent-driven service instance, establishing the required control and telemetry channel between the SO and the deployed service components. These commands are sent sequentially because the action on one domain depends on the other (e.g., there is the need to create the mobile core instance first to then later point out gNB instances to the appropriate AMF instance).

After domain registration and initial deployment of network function instances in the different domains for a given service request, the seriously considered potential approach is that the SO listens for heartbeat loss events from the Domain Registry and fault alerts from the SLA Monitor. In the event of a domain failure or a consistent SLS violation due to performance problems or dynamic runtime conflicts, it is responsible for triggering recovery procedures, such as service migration or emergency termination. Finally, the SO manages the infrastructure's "clean-up" by requesting the decommissioning of dedicated Kafka Service Topics once a service's lifecycle expires, preventing resource leakage and maintaining a consistent network state.

The Domain and Capability Registration module acts as the authoritative inventory and lifecycle manager for the network infrastructure layer. This module is responsible for handling the registration process of new Domains by processing the message received in the IDMO Service adapter (SA) from the global handshake channel and validating incoming registration requests from the different domains (e.g., RAN, Core, transport), storing associated data in the Domain Registry database. Once a domain is successfully authenticated, the module initializes the communication infrastructure with domains by requesting the creation of dedicated, bi-directional Kafka topics through the IDMO SA. This proactive approach ensures traffic isolation and the readiness of control channels before any actual service orchestration begins. Furthermore, the module continuously monitors the operational health of registered domains via a heartbeat mechanism; if a predefined timeout is exceeded, it immediately broadcasts a "domain heartbeat lost" event to the Service Orchestrator to trigger recovery procedures. In addition, it maintains a dynamic map of technical capabilities, such as available CPU, RAM, and supported network services catalogues available at each domain, providing the essential information foundation for domain selection and feasibility verification during the preparation phase. Finally, the IDMO SA module is the interface with the cross-domain management bus to send and receive message to/from the different domains, which also count with their corresponding Domain SA.

3.3.3 Domain Components

A domain can be defined as the set of infrastructure resources (e.g., computing Points of presence (PoPs), transport network nodes) and the management and orchestration stack able to satisfy the requests issued by the IDMO through the cross-domain management bus. These management and orchestration components are the Domain Service adapter, the DMO and the Infrastructure Domain Manager (IDM).

3.3.3.1 Domain Service Adapter

The Domain-specific SA is the module that connects the Domain to the IDMO through the cross-domain SBMA bus. This SA translates the intended requests coming from the IDMO for requesting the allocation/deallocation of services in the different domains and forwards them to the DMO, which performs the lifecycle management of such services. The Domain SA is also in charge of launching the domain registration and composing the messages in coordination with the DMO to report the collection and abstraction of domain-level information (e.g., service catalogues, resource availability) to update the domain-view included in the IDMO database module.

3.3.3.2 Domain Management Orchestrator

A DMO is the domain-level orchestrator responsible for the execution of domain-scoped service requests issued by the IDMO and processed by the Domain SA. UNITY-6G assumes that a DMO is in charge of a single administrative and technological domain (e.g., RAN, Core Network, Transport). The DMO executes the domain-internal orchestration workflows accordingly ruling the lifecycle management (i.e., instantiation, configuration, scaling, modification, and termination) of the services associated to the received requests on its available northbound interface. For this execution, the DMO uses its southbound interface towards one or more Infrastructure Domain Managers (IDMs). The concrete realization of the DMO is domain dependent. For instance, those domains related with the creation of software-based network services (e.g., an instance of the 5G core), the DMO can be aligned with the NFV Orchestrator (NFVO) component of ETSI NFV specifications, like ETSI Open-Source MANO (OSM) or NearbyOne Controller. Other example of a DMO component for an O-RAN domain is the Service Management and Orchestration (SMO) component defined in O-RAN specifications. In the Transport domain, the DMO can be implemented as the Network Slice Controller described in UNITY-6G D2.3 [3], which is responsible to push the configuration of transport network elements, such as routers, for the end-to-end lifecycle management of transport network slices.

Specifically for the vertical application domain/s, the UNITY-6G architecture provides an additional mechanism to directly interact with the DMO. Concretely, the service orchestration capabilities for the lifecycle management of vertical applications are exposed by the DMO using a specific CAMARA API, as explained in UNITY-6G D3.1 [31]. In the context of the UNITY-6G exposure layer, CAMARA exposes a northbound interface through which vertical application providers can also interact with the underlying network and edge infrastructure to perform application lifecycle management using standardized APIs.

3.3.3.3 Infrastructure Domain Manager

An Infrastructure Domain Manager (IDM) is the domain-internal management function component offering an interface to the DMO to perform the lifecycle management of infrastructure resources, ensuring that requirements of the received request are translated into tangible resource allocations (e.g., CPUs, RAM memory, networking capabilities) thanks to the exposure of resource inventory, capacity, availability and topology information. Similarly to the DMO, the concrete realization of the IDM is domain dependent. In computing-oriented domains, where services are realised by the creation of virtualised software (e.g., mobile core, RAN), the IDM corresponds to a virtualization or container infrastructure management platform like Openstack or Kubernetes, respectively. In particular, UNITY-6G project focuses on the

use of containers due to is agility, portability and self-healing capabilities. In the Transport domain, the IDM may be implemented as a Software-Defined Networking (SDN) Controller, responsible for the control and programming of transport network resources via standardized southbound interfaces. The IDM operates strictly within the domain boundary and does not perform cross-domain orchestration or decision-making. As mentioned before, a DMO can manage multiple IDMs instances.

4 INTEGRATED SERVICES

This section presents services that are being developed in the framework of the UNITY-6G activity linked with the key project objectives, including the semantic communication idea, the multiple Radio Access Technologies (multi-RAT) scenario and the digital twin as a service for sustainable networks and resource optimization. In addition, several intelligent applications that can be provided as xApps, rApps, and potentially dApps in the integrated network are described from the WP5 service (and not from the algorithmic) perspective.

4.1 DEFINITION OF AN INTEGRATED SERVICE

In the context of UNITY-6G, an Integrated Service is understood as a service that is realized through the coordinated composition of multiple applications operating across multiple network domains, with the objective of delivering end-to-end functionality for a specific Use Case or Proof-of-Concept. The term integrated therefore explicitly captures two complementary dimensions: the integration of applications and the integration of network domains.

An integrated service is not a standalone application or a domain-specific function. Instead, it represents a logical service construct built by combining different types of applications - such as xApps, rApps, dApps, and vertical applications - that interact and cooperate through UNITY-6G SBMA. These applications expose and consume services via standardized interfaces, enabling modularity, interoperability, and flexible deployment across the UNITY-6G architecture. When relevant, Digital Twin capabilities are incorporated as part of the service to support monitoring, evaluation, prediction, and optimisation of network behaviour.

From the application perspective, each integrated service is composed of a set of core applications, which implement the primary service functionality, and a set of supporting applications, which enhance or complement the core service by providing additional capabilities such as monitoring, analytics, prediction, conflict management, trust, or orchestration. All applications within an integrated service act in coordination and are jointly responsible for achieving the service's end-to-end objectives rather than optimizing isolated components.

From the network perspective, an integrated service spans multiple network domains, including but not limited to RAN, transport network, core network or NTN. The service logic is therefore not confined to a single domain but enables cross-domain awareness and decision-making, allowing optimisations and adaptations that consider the end-to-end service context. This cross-domain integration is a fundamental aspect of UNITY-6G and differentiates integrated services from traditional, domain-specific services.

Within UNITY-6G, four integrated services are defined, each corresponding to a specific PoC, as depicted in Figure 4-1. These integrated services serve as the main vehicle to demonstrate how multiple UNITY-6G applications and network domains can be combined into coherent, end-to-end services that address the requirements of advanced 6G use cases. As such, integrated services provide the conceptual and functional link between low-level network control and optimization mechanisms and high-level, use-case-driven service delivery.

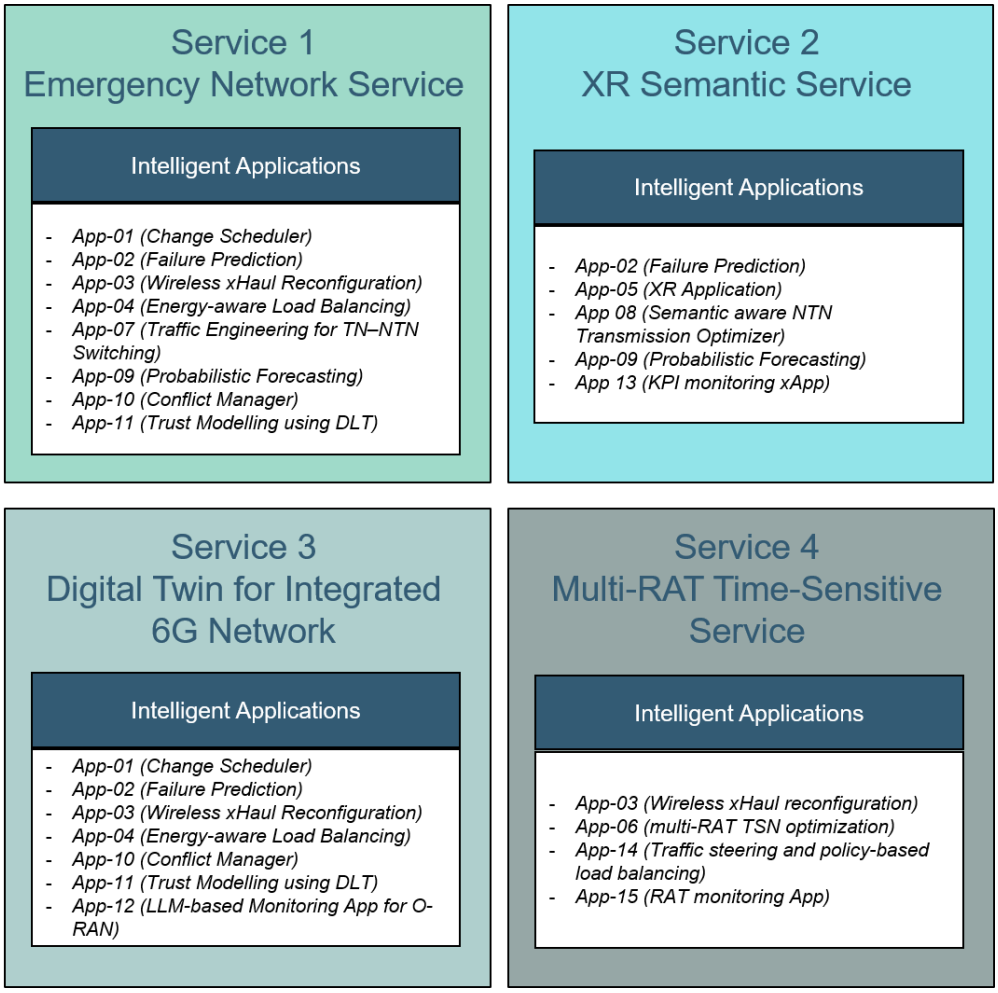

Figure 4-1: Integrated services and associated intelligent applications in UNITY-6G

4.2 LIST OF INTEGRATED SERVICES

4.2.1 Emergency Network Service

4.2.1.1 Service Overview and Scope

The **Emergency Network Service** is an integrated end-to-end service designed to ensure resilient, sustainable, and trustworthy communications in disaster scenarios, where network resources are scarce and subject to rapid topology changes. The service operates over a heterogeneous integrated network, combining terrestrial networks and Non-Terrestrial Networks NTNs, and supporting multi-tenant operation involving mobile network operators, internet service providers, and satellite operators.

In emergency conditions, multiple tenants simultaneously compete for limited radio, transport, and energy resources, often with contradictory requirements in terms of latency, capacity, energy consumption, and service prioritization. The Emergency Network Service addresses this challenge by integrating AI-driven resource optimization, policy-based conflict resolution,

trust mechanisms, and energy-aware decision making into a unified service logic. Each tenant expresses its service, capacity, latency, and energy policies, which are continuously monitored and enforced at network level.

The service leverages distributed AI techniques to monitor network status and resource usage across domains, detect violations of service level agreements, and dynamically adapt resource allocation. Policy-based decision engines are used to prioritize critical and emergency communications over non-critical traffic, while renewable energy awareness is integrated into resource allocation decisions to improve sustainability. Explainable AI (XAI) mechanisms provide transparency on how allocation and conflict-resolution decisions are taken, fostering trust among tenants. In parallel, Distributed Ledger Technology (DLT) is employed to provide immutable, auditable records of resource allocation and policy enforcement actions, enabling accountability and dispute resolution in multi-tenant environments.

When conflicts arise - such as competing energy usage or capacity demands - the Emergency Network Service proactively resolves them by reallocating resources, adapting configurations across domains, or notifying affected tenants. Additional energy-efficiency enablers, including advanced techniques for reducing peak-to-average power ratio in NTN transmissions and IoT-based energy management mechanisms, further enhance the sustainability and robustness of the service.

4.2.1.2 Integration of applications in the service

Designing and maintaining a sustainable Emergency Network Service in UNITY-6G is structured around four main lifecycle stages: planning, normal operation, moment of disaster, and post-disaster recovery. Each stage activates different applications of the integrated service, which act in coordination to preserve service continuity, sustainability, and trust across the integrated terrestrial–NTN network.

In the **planning stage**, offline and near-real-time planning activities are performed to anticipate disruptive events. Network operators rely on “what-if” analysis and predictive insights to prepare recovery strategies and contingency plans. In this phase, ***App-09 (Probabilistic Forecasting)*** predicts future requirements and potential resource shortages based on historical network utilization and expected traffic evolution. These forecasts would enable to proactively adjust resource allocation, reserve additional capacity, and optimize multi-domain resource provisioning to avoid SLA violations. Complementary to this, the outputs of ***App-02 (Failure Prediction)*** are used to identify vulnerable network segments and failure-prone zones, enabling proactive planning. These activities allow the network to enter a disaster situation with predefined fallback options and coordinated change plans.

During **normal operation**, the network operates under standard conditions while remaining prepared for emergency situations. In this stage, monitoring, prediction, and trust mechanisms are continuously active. ***App-02 (Failure Prediction)*** analyses transport and wireless telemetry to detect early signs of degradation and predict potential failures, providing valuable inputs for preparedness and risk mitigation. In parallel, ***App-11 (Trust Modelling using DLT)*** operates recording resource allocation decisions and policy enforcement actions in an immutable and auditable manner, ensuring transparency and accountability among multiple tenants. This continuous operation establishes a trusted baseline for subsequent emergency handling.

The **moment of disaster** is characterized by sudden topology disruptions, energy constraints, and traffic surges for critical services. At this stage, the Emergency Network Service focuses on maintaining the maximum possible level of connectivity using the remaining active infrastructure. ***App-07 (Traffic Engineering for TN–NTN Switching)*** plays a central role by enabling fallback connectivity through NTNs when terrestrial coverage is lost or degraded. By dynamically steering traffic between TN and NTN components, it ensures that end users equipped with appropriate capabilities can maintain cellular connectivity even when terrestrial access is compromised. Simultaneously, ***App-04 (Energy-aware Load Balancing)*** becomes critical, optimizing user association and resource allocation to minimize energy consumption while preserving essential services under scarce resource conditions. ***App-03 (Wireless xHaul Reconfiguration)*** complements these actions by dynamically adapting wireless transport paths, identifying alternative routes, and adjusting link parameters to preserve emergency-service connectivity when links become overloaded or unavailable. During this highly dynamic phase, ***App-10 (Conflict Manager)*** detects and resolves conflicts arising from competing optimization objectives and tenant policies, ensuring that emergency priorities are enforced consistently and transparently.

Finally, in the **post-disaster** recovery stage, the focus shifts to restoring full network functionality in a controlled and stable manner. The ***App-01 (Change Scheduler)*** is a key enabler in this phase, supporting the coordinated and phased re-introduction of network resources. By reassessing scheduled recovery actions against the updated topology, it ensures that restoration steps do not introduce new disruptions or conflicts. ***App-04 (Energy-aware Load Balancing)*** continues to optimize energy usage during recovery, supporting gradual scale-up of services while avoiding unnecessary consumption peaks. ***App-10 (Conflict Manager)*** remains active to resolve conflicts that may arise between recovery actions and ongoing emergency or priority services. Throughout the recovery process, ***App-11 (Trust Modelling)*** maintains an auditable record of decisions and resource usage, reinforcing trust among tenants and stakeholders.

4.2.2 XR Semantic Service

4.2.2.1 Service Overview and Scope

The XR Semantic Orchestrated Service is an integrated end-to-end service designed to support immersive XR and holographic communications over the UNITY-6G architecture by shifting the communication paradigm from conventional bit-level transmission to meaning-oriented semantic information exchange. The service targets highly demanding applications requiring ultra-high data rates, low latency, and efficient use of spectrum and energy across integrated terrestrial networks and, where link conditions permit, through intelligent switching to non-terrestrial network (NTN) segments. Direct XR streaming over satellite links is acknowledged to be challenging due to inherent NTN latency and bitrate constraints; the service therefore could be enriched by relying on intelligent TN/NTN path selection and seamless switching based on connection availability rather than sustained only delivery through a single connection path.

In particular, the service incorporates HOLO (holographic communication) capabilities, enabling real-time transmission and reconstruction of 3D volumetric content, where semantic communication plays a key role in reducing the overwhelming data requirements of

holographic scenes by transmitting only task-relevant semantic representations instead of raw point clouds or video data.

To offer semantic communications through this service, the communication paradigm needs to be fundamentally redesigned from the conventional bit-level transmission to the meaning-oriented information exchange for supporting and implementing the semantic communication services in the UNITY-6G framework, where only the task-relevant semantic information is extracted, transmitted, and reconstructed. This necessitates the design of an end-to-end semantic communication pipeline, where the user equipment (UE) or sensing devices first extract the semantic information (SI) using AI-driven models, e.g., deep learning feature encoders or knowledge graph representations, to capture the contextual meaning of data rather than its raw form. In the context of HOLO-enabled XR services, this includes extracting semantic features from volumetric captures, spatial geometry, object relationships, and user interactions in 3D space. It is important to note that the above semantic pipeline primarily applies to the general and DLR-driven semantic service components. For HOLO-based deployments using XR application, the delivery model differs: the full 3D scene is rendered server-side and transmitted as a pixel stream to XR headsets via Hololight Stream. In this case, the XR Semantic Service's role is to provide dedicated low-latency network resources for the pixel stream and to supply closed-loop network feedback to the rendering server, rather than performing semantic feature extraction at the device.

The extracted semantics are then filtered by task importance, Quality of Experience (QoE) needs, and system constraints including latency, energy, and Age of Information (AoI), such that only the most relevant information is transmitted. The filtered semantic content is then encoded by using the adaptive semantic-aware encoding schemes that adapt to the channel conditions and are efficiently transmitted over terrestrial network segments, with NTN segments supported through intelligent TN/NTN path switching where link conditions permit. For holographic applications, this filtering ensures that critical elements such as user presence, gestures, and key scene components are prioritised, while redundant spatial details are omitted or inferred at the receiver.

At the receiver side, the semantic decoding is not limited to reconstructing signals but also includes inference using locally available or shared knowledge bases, which can recover or even predict missing information, thus significantly reducing communication overhead. In HOLO scenarios, this enables high-quality reconstruction of holograms through semantic completion and AI-based rendering, even under constrained bandwidth conditions. This whole process is supported by a distributed intelligence architecture in UNITY-6G where lightweight semantic processing is done at the UE, more complex inference and model adaptation at the edge, and global knowledge base management and training in the cloud.

The O-RAN integration further allows real-time and non-real-time control of semantic communication through xApps and rApps, which can optimise semantic filtering, resource allocation and scheduling based on network-wide conditions and user-centric requirements. Furthermore, AI-driven techniques such as reinforcement learning and federated learning are essential to enable semantic models and resource management policies to continuously adapt in dynamic environments. UNITY-6G can achieve significant spectral efficiency improvement, latency reduction and energy efficiency improvement by jointly optimising the semantic extraction, transmission and inference over multiple network layers. Semantic communication can enable intelligent context-aware services over heterogeneous networks and is a key

enabler for the future 6G applications, particularly for HOLO-driven immersive XR experiences requiring real-time, interactive, and high-fidelity holographic communication.

4.2.2.2 Integration of applications in the service

The XR Semantic Service, including HOLO-enabled holographic XR communications, is realized through the coordinated integration of applications spanning the application, RAN, transport, core, and NTN domains within the UNITY-6G framework. The service operates as a system in which monitoring, analytics, decision-making, and actuation components interact continuously to support immersive XR and real-time holographic experiences.

At the application domain, the ***App-05 (XR Application)*** generates immersive 3D and holographic content, including HOLO-based volumetric media and digital twin representations, with server-side rendering hosted on GPU-enabled virtual machines at the edge or cloud. The fully rendered holographic scene is then delivered to end-user XR headsets (e.g., Meta Quest, Apple Vision Pro) via Hololight Stream, a low-latency pixel streaming technology. This server-side rendering and pixel streaming model is the defining delivery mechanism for HOLO within the XR Semantic Service, enabling immersive holographic experiences without requiring high-end processing on the headset itself. Semantic information related to XR, and holographic content is extracted and processed by the ***App-08 (Semantic-aware NTN Transmission Optimizer)***, which enables meaning-oriented data transmission across NTN links. By transmitting only task-relevant semantic information such as object structure, spatial relationships, user interactions, and key holographic scene elements this application directly reduces bandwidth consumption and energy usage while maintaining XR and HOLO quality.

Resource allocation is optimized across multiple domains based on semantic context, XR/HOLO task requirements, and user preferences. This includes bandwidth allocation in the NTN domain, compute resource allocation at the application and edge layers based on task criticality and holographic rendering demands, and radio and transport scaling in terrestrial networks to meet strict XR and holographic performance requirements such as low latency and high data fidelity. ***App-09 (Probabilistic Forecasting)*** supports this by forecasting future resource requirements based on network resource utilization trends and allocate resources to maintain the required quality of experience.

Network-level awareness is further enhanced through monitoring and analytics applications such as ***App-13 (KPI monitoring xApp)***, which provides radio-level KPIs, and ***App-02 (Failure Prediction)***, which detects and predicts degradations that may affect XR and holographic service quality. These inputs allow the system to adapt semantic filtering strategies, resource assignments, and routing decisions in response to changing network conditions, ensuring continuity and quality of HOLO immersive experiences. For HOLO-based XR application specifically, these network feedback signals - including available bandwidth, link latency, and QoE metrics collected by ***App-13 (KPI monitoring xApp)*** and ***App-02 (Failure Prediction)*** - are passed via the SBM to the HOLO-based XR application running on the rendering server. The application uses this feedback to dynamically adapt encoder bitrate, compression level, and streaming resolution via Hololight Stream. This application-layer adaptation loop ensures that the pixel stream quality is continuously optimized to match available network capacity, maintaining holographic fidelity under varying network conditions while respecting strict latency requirements.

The integration with O-RAN technologies enables both near-real-time and non-real-time control of semantic communication through xApps and rApps, optimizing semantic filtering, scheduling, and resource management based on network-wide conditions. This is particularly important for HOLO scenarios, where dynamic adaptation is required to maintain synchronization, spatial consistency, and real-time interactivity. Techniques such as reinforcement learning and federated learning allow semantic models and resource-management policies to adapt to dynamic environments.

Overall, the XR Semantic Service represents an integrated service where multiple applications operate in coordination across domains to optimize semantic extraction, transmission, inference, and resource allocation. By incorporating HOLO-driven semantic communication for holographic XR, the service improves spectral efficiency, latency, energy efficiency, and QoS, enabling intelligent, context-aware, and immersive holographic experiences over future 6G networks.

4.2.3 Digital Twin for Integrated 6G Network

4.2.3.1 Service overview and scope

The Digital Twin for Integrated 6G Network service provides an AI-powered digital representation of the entire heterogeneous UNITY-6G infrastructure, spanning Open RAN, X-haul/transport, and Non-Terrestrial Network (NTN) domains. The service is designed to support advanced network monitoring, evaluation, and optimization, with a strong focus on energy efficiency, sustainability, resiliency, and proactive operation.

In the third scenario of the proof of concept, a 6G-powered Digital Twin (DT) for integrated network infrastructure monitoring is demonstrated. The Digital Twin models physical assets such as network components, equipment, and their interactions across heterogeneous domains, enabling operators to observe, analyze, and predict network behavior in a virtual environment. The DT application is AI-powered and focuses on monitoring sustainability metrics of the whole network, including power consumption, energy source utilization, and overall resource efficiency. In particular, it enables the evaluation of the interplay between microgrid power management and network operations, which is a key aspect for sustainable 6G infrastructures.

By integrating Open RAN, X-haul, and satellite technologies, the Digital Twin enables real-time and predictive monitoring of the network infrastructure. The virtual representation of the physical network allows operators to identify potential issues, evaluate mitigation strategies, and make proactive decisions before problems impact live services. Leveraging AI and the capabilities of 6G networks, this service provides a comprehensive solution for infrastructure monitoring that helps to reduce operational costs, improve efficiency, and minimize the environmental footprint of the network.

This service corresponds to UNITY-6G Use Case 3, which establishes a Digital Twin framework aimed at enhancing network resiliency and robustness in complex integrated 6G environments. By accurately replicating the behaviour of physical assets and their cross-domain interactions, the DT enables predictive control and proactive decision-making, allowing failures to be prevented and performance to be optimized ahead of time. The Digital Twin also supports several key objectives, including providing a realistic environment for AI/ML model

training, rapid testing and deployment of new 6G services, support to preventive maintenance, and advanced resource management across the edge–cloud continuum.

The novelty of this service lies in its AI-driven closed-loop operation, where the Digital Twin predicts future behaviours using data-driven and GenAI-enhanced models. This proactive perspective significantly improves the reliability, sustainability, and overall efficiency of the integrated 6G network, making the Digital Twin for Integrated 6G Network a central pillar of the UNITY-6G vision.

4.2.3.1 Integration of applications in the service

The Digital Twin for Integrated 6G Network service is realized through the coordinated integration of multiple applications operating across transport, RAN, NTN, and orchestration domains, forming a closed-loop control system centered on energy-aware and sustainability-driven operation.

At the core of the service, the ***App-01 (Change Scheduler)*** acts as a key Network Digital Twin application by managing and validating planned and unplanned topology changes over time. It ensures that the Digital Twin maintains a time-consistent and accurate representation of both current and future network states, recalculating expected topologies when unexpected events occur. This capability is essential for energy-aware planning, allowing the DT to evaluate how scheduled actions and recovery strategies impact power consumption and sustainability metrics before they are applied to the real network.

The ***App-02 (Failure Prediction)*** and ***App-03 (Wireless xHaul Reconfiguration)*** applications provide critical monitoring and analytics inputs to the Digital Twin. ***App-02 (Failure Prediction)*** predicts failures and localizes their root causes, enabling the Digital Twin to anticipate disruptions and assess their impact on energy usage and service continuity. ***App-03 (Wireless xHaul Reconfiguration)*** supports dynamic adaptation of the transport network, allowing the DT to evaluate alternative routing and configuration strategies that minimize energy consumption while maintaining connectivity across terrestrial and NTN segments.

Energy-aware optimization is further reinforced by applications such as ***App-04 (Energy-aware Load Balancing)***, which contributes models and metrics related to computing and radio resource usage to the Digital Twin. These inputs allow the DT to assess different resource-allocation strategies and their sustainability impact, particularly in scenarios where energy availability is constrained or dependent on renewable sources.

The Digital Twin also integrates higher-level analytics and reasoning capabilities through applications such as the ***App-12 (LLM-based Monitoring App for O-RAN)***, which enhances explainability and supports advanced analysis of telemetry and sustainability indicators. Conflict management and trust mechanisms, such as ***App-10 (Conflict Manager)*** and ***App-11 (Trust Modelling using DLT)***, ensure that optimization decisions evaluated within the Digital Twin remain consistent, auditable, and aligned with multi-tenant policies.

Together, these applications enable a closed-loop, AI-driven Digital Twin service that continuously monitors energy consumption, resource utilization, and network performance across integrated domains. By allowing operators to test, evaluate, and optimize configurations in a virtual environment before applying them to the physical network, the Digital Twin for

Integrated 6G Network service plays a fundamental role in achieving sustainable, resilient, and energy-efficient 6G network operation.

4.2.4 Multi-RAT Time-Sensitive Service

4.2.4.1 Service overview and scope

The multi-radio access technology (RAT) time sensitive service will enable end-to-end deterministic communication utilizing different RATs in the UNITY-6G architecture. The multi-RAT time-sensitive services (TSN) will provide (i) ability to monitor each of the RATs separately (cellular and IEEE 802.11), (ii) means to control each of the RATs, (iii) TSN-related features on both RATs, as well as (iv) ability to perform traffic steering/duplication between both RATs.

The main involved domains from the UNITY-6G architecture to support this end-to-end multi-RAT time-sensitive service are the non-3GPP IEEE 802.11 RAN domain, the xHaul transport domain and the 3GPP RAN domain. Depending on the data plane path termination between RAN domains (e.g. if the data path in the 3GPP network is terminated in the edge or in the cloud), then the core domain can be included in the end-to-end path for this service.

The main enablers for time-sensitive service are the mechanisms to provide deterministic and low-latency channel access in the RAN part. In the non-3GPP IEEE 802.11 domain these mechanisms will relate to the Continuous Multi Link Operation (ConMLO) and time-based scheduling that will support dedicated channel access for certain station(s) and access point(s) [37], [38]. In the cellular RAN domain, the time-sensitive service will be enabled by the mechanisms such as uplink (UL) prescheduling, UL configured grant, intra-slice prioritized scheduling for specific traffic flows and one-way latency monitoring through in-band network telemetry. Such mechanisms will allow the bounding of the round-trip time of targeted traffic flows under a certain threshold. In the xHaul domain the main mechanisms that will be used to support time-sensitive services will be related to priority-based queue operation utilizing traffic flow mapping and real-time capacity monitoring of the link.

The multi-RAT Time Sensitive Networking service is associated with UNITY-6G Use Case 4, which establishes NPN deployment for different Industry 4.0 applications where the bounded and low latency as well as high reliability is a key. By utilizing the O-RAN principles of disintegrated RAN, as well as distributed management utilizing different controlling apps at different controlling granularity, multi-RAT TSN service will offer highly cross-RAT utilized NPN for the seamless network connectivity. This service will be demonstrated in PoC 2 in experimental scenario one. The first experimental scenario will relate to enabling end-to-end time-sensitive network across different RATs (cellular and IEEE 802.11 based), xHaul and core, capable of maintaining end-to-end latency smaller than 20 ms, while the latency on radio part only will be below 10 ms.

4.2.4.2 Integration of applications in the service

In addition to the main low-level mechanisms, there are several applications that are involved in supporting this service. To be able to adjust the service accordingly, monitoring APPs need to be in place in all the involved network domains. As such the ***App-15 (RAT monitoring App)*** will monitor the low-level performance of traffic flows as well as the status of the channel. In case of the non-3GPP IEEE 802.11 domain the RAT monitoring APP will monitor the load in the network by extracting the queue information over time, number of re-transmissions of the

packets, as well as the statistics of the traffic flows (e.g. number of corrected transmitted packets, number of an-acknowledged packets etc.). The RAT monitoring APP in the cellular RAN domain will monitor: signal-related metrics (e.g., RSSI, RSRP, RSRQ), Channel Quality Indicator (CQI), downlink and uplink Modulation and Coding Scheme (MCS), Transport Block (TB) related metrics (e.g., TB size, TB Error Rate (TBLER), HARQ retransmission rate, TB latency) and UE throughput related metrics (e.g., DL/UL instantaneous/average/peak UE throughput, DL/UL throughput efficiency/variability, DL/UL goodput).

Once the wireless channel is monitored as well as traffic flow performance is visible to the network, the next step is to be able to perform re-configuration of each domain to maintain the deterministic latency budget in the end-to-end path. As such the ***App-06 (multi-RAT TSN optimization)*** will control and manage different parameters in each of the UNITY-6G domains. In case of the non-3GPP IEEE 802.11 domain the controlling APP will update schedule on each of the operational link (in case of MLO). The update will be performed based on locally (domain-based) monitored parameters. Regarding the cellular RAN domain, the ***App-06 (Multi-RAT TSN optimization)*** will update in (near) real-time the RAN slicing configurations such as the dedicated, min and max physical resource block (PRB) ratios based on the O-RAN RAN Control (RC) Service Model and the scheduling policy, and scheduling-related parameters for dedicated time-sensitive traffic flows. Similar to the non-3GPP case, the update will be based on domain-based monitored parameters.

The next APP that is involved in both domains (cellular and non-3GPP IEEE 802.11 RAN) is ***App-14 (Traffic steering and policy-based load balancing)***. In case that the domain cannot maintain the actual latency or reliability budget by (re)-configuring the domains' controlling mechanisms, the traffic steering APP can initiate to move the traffic flows from one RAN to the other, or even by duplicating the transmissions in both domains. Once the traffic steering is initiated, the xHaul transport network need to be (re)configured to support routing of the new steered/hand-overed traffic flows. As such, ***App-03 (Wireless xHaul reconfiguration)*** will help to achieve this.

4.3 INTEGRATION OF INTELLIGENT APPLICATIONS

4.3.1 App-01: Change Scheduler

4.3.1.1 Definition

The change scheduler is an NDT application for managing and coordinating scheduled and unscheduled changes in a transport network topology to ensure service continuity and operational stability. It introduces an architecture capable of detecting, analyzing, and, jointly with the NDT, resolving conflicts between multiple planned changes, as well as adapting to unexpected events that alter the network structure that was initially used in the past as base for defined scheduled changes.

This section begins by presenting the context in which the change scheduler operates, including the scenarios where scheduling conflicts may arise and the necessity of addressing them. Following this, the architecture is described, outlining its overall functioning, workflow, and the interaction between its subcomponents. Finally, each subcomponent of the change

scheduler is examined in detail, explaining its specific role within the system and its contribution to the overall operation.

4.3.1.1.1 Problem 1: Issues due to inconsistency among planned changes

In the operation of a network operator, it is usual the need of introducing upgrades of hardware or software in distinct nodes and links in order to satisfy customer demands. This means that topology can change over time. The changes may be planned for reasons like maintenance of windows, load balancing, energy savings, or network upgrades.

Recently, the IETF has proposed an operational approach called Time-Variant Routing (TVR), where network topology elements such as nodes, links, and adjacencies change in a predictable, scheduled manner [39].

In this operational approach presented by the IETF, a network engineer can plan a scheduled change at a given time on the proposed network. If considered individually, that scheduled change can be assessed and executed at due time. The assessment of the scheduled change can be performed by means e.g. of algorithms embedded in the Network Controller. That assessment can help to understand the implications of such particular change.

However, when taken in isolation, any change cannot be put in context of many other different scheduled changes that could run in parallel in the network. Figure 4-2 illustrates the case when two different network engineers propose decoupled scheduled changes in the network. Separately, both changes can be viable, but in combination, they could result in network disruption (as described in the figure) or in other operational issues, like the creation of routing loops, etc.

In current IETF TVR documents, there is a lack of coordination between scheduled changes. Thus, a cascade of subsequent uncoordinated scheduled changes may cause network segments to become disconnected or create routing loops, thereby harming network stability and operation. Proper coordination and conflict resolution mechanisms are therefore essential to avoid inconsistencies, loops, and disconnected segments in the network when multiple scheduled changes occur.

4.3.1.1.2 Problem 2: Issues due to network context change motivated by unplanned changes

While scheduled changes can be anticipated and coordinated, telecom networks frequently face unexpected disruptions that alter their topology without prior notice such as natural disasters or power outages showcased in UNITY-6G Disaster Use Case.

These events can invalidate previously scheduled changes or create new operational constraints. For instance, a maintenance window planned for a specific link may become infeasible if that link is already down due to an outage. Similarly, a routing optimization scheduled for a future time may introduce loops or disconnect segments if the underlying topology has changed unexpectedly. The absence of a mechanism to re-evaluate and adapt scheduled changes in light of unplanned events can lead to severe service disruptions. Operators must therefore:

- Detect topology changes in real time.
- Assess the impact of these changes on future scheduled operations.

- Recompute and reschedule actions to maintain network integrity and service continuity.

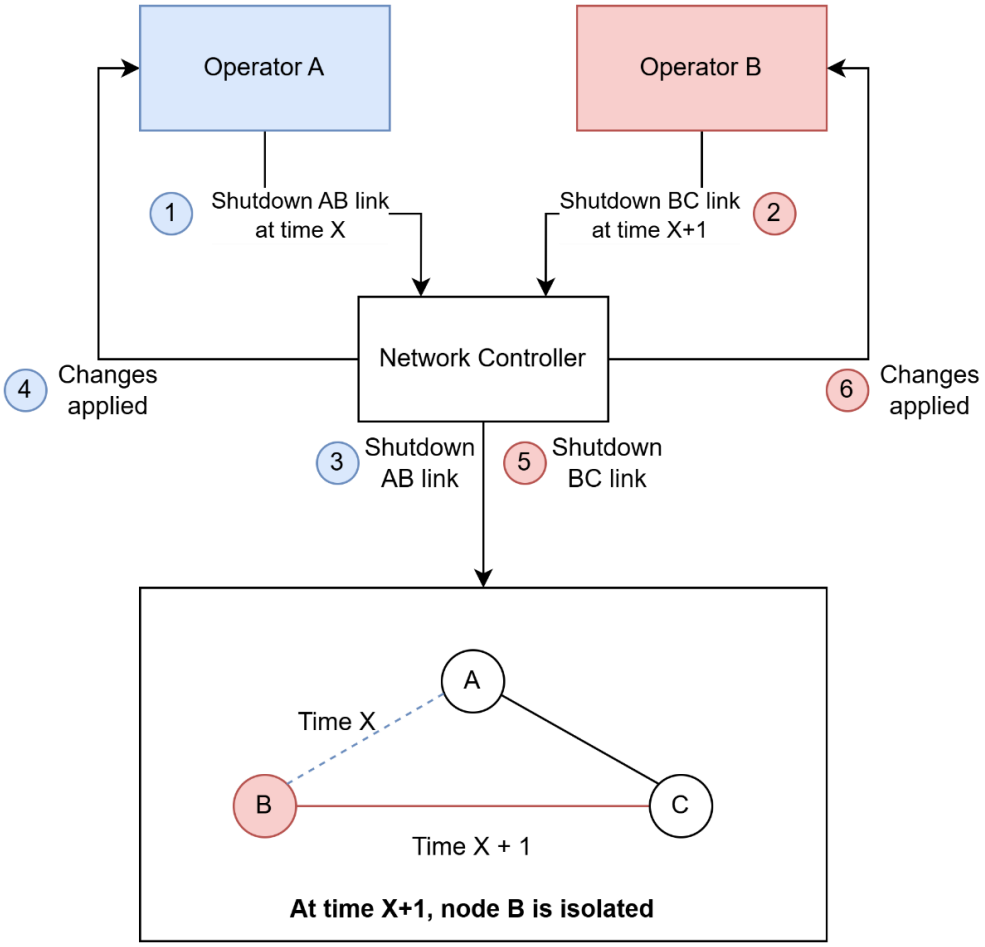

Figure 4-2: Use Case with a two-operator conflict

4.3.1.1.3 Architecture

In order to prevent the aforementioned problems, the change scheduler takes into consideration the network topology and the active and future scheduled changes, performing an evaluation of problems that can be motivated by cascading schedule changes in the network.

It will be in charge of collecting the scheduled changes for the network through a submodule named “Change Scheduler Input”, taking into account the activation /deactivation time for the change. A “Change Scheduler Database” stores the present and future changes already scheduled. The Change Scheduler Analytics takes into consideration the latest proposed scheduled changed, the previous scheduled changes and the existing topology view, if necessary, processing all of them and determining with the NDT if the latest proposed change creates network disruption, as well as the conflicting changes and the disruption created (network disconnection, routing loop, etc.). The resulting analysis is provided as an output for further actions. Figure 4-3 represents the architecture described.

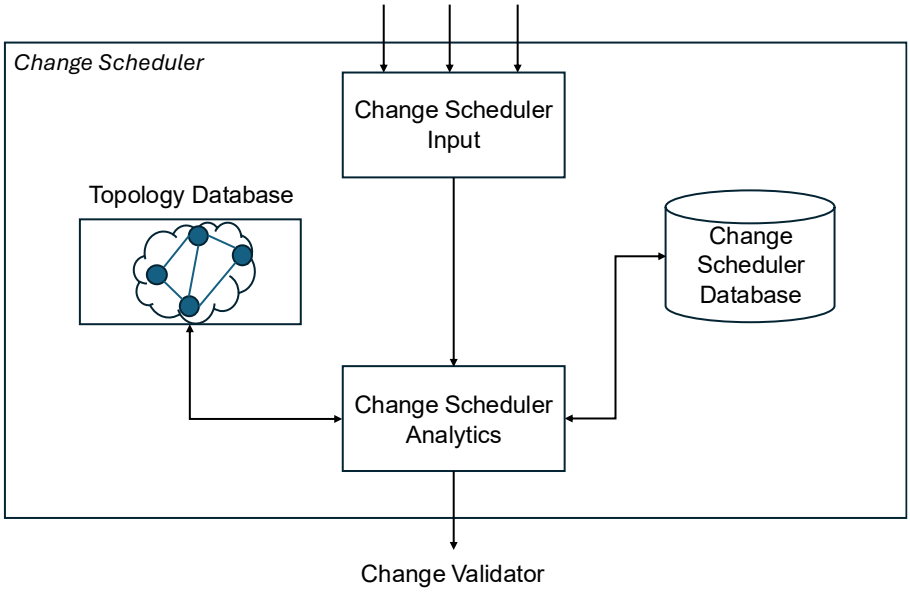

Figure 4-3: Change Scheduler Architecture

The overall system workflow is summarized as follows:

1. A change is planned in the network in a specific point in the future time.
2. The Change Scheduler input component receives the change and extracts it into three variables: the time for the change and the time of the request, and the actions to take in the network ordered in a list. These are sent to the analytics.
3. The Change Scheduler Analytics then receives the new proposed changes, extracts the time of the action to find actions that coincide in time with the new one.
4. The Analytics obtains the topology for the time of the action from the topology database
5. The Analytics sends to the NDT to analyze the result in the topology after applying the collected actions in the analyzed time. It determines if there is any conflict in these actions such as, part of the network disconnected, unintended loops, isolated nodes, etc.
6. In case there are no conflicts, the change is accepted and both databases are updated with the proposed changes and the resulting topology respectively
7. In case there are conflicts, the change is rejected, and a notification is sent indicating which network disruption is provoking the change which actions are conflicting with the proposed one
8. Periodically, the analytics module analyses any unrevised changes in the topology, to avoid that they do not impact in the future network changes

4.3.1.1.4 Components

Change Scheduler Input

The change scheduler input component serves as the entry point to the system. It receives the changes requested and the time for applying them in the network topology. This component

gathers all this information and homogenise it to share the same format to be shared along the system. Three main subjects are extracted from these sources:

- The time when the change is going to be applied. It is expressed in ISO 8601 format (YYYY-MM-DD or YYYY-MM-DDThh:mm:ssZ). This time cannot be previous to the current time, otherwise it will be rejected.
- The time when the request was done, expressed in ISO 8601 format
- The list of actions to apply the requested change. An action is formed by:
    - Type of element receiving the action (e.g. node, interface, link)
    - Reference to the specific element receiving the action
    - Modification over the element (e.g. shut down, power on, filter, delete, etc.)

After processing the information, it is sent to the analytics component.

Change Scheduler Database

The Change Scheduler Database is responsible for storing all planned modifications to the network topology. Each entry in the database includes the time when the change is scheduled to occur, the time when the request was made, and the list of actions associated with the change. Actions are recorded in a normalized format to ensure consistency across different sources, including management portals, command-line interfaces, and signalling protocols.

The database supports chronological indexing and conflict detection by enabling efficient queries based on scheduled execution time. It also maintains a historical log of accepted and rejected changes, which is essential for auditing and for recalculating future topologies in case of unexpected events. Updates to the database occur whenever a new change is approved or when previously scheduled changes are modified or cancelled.

Topology Database

The Topology Database maintains a complete representation of the network topology over time. It stores both the current state of the network and predicted future states resulting from scheduled changes. Each topology snapshot is associated with a timestamp, allowing the system to reconstruct the network configuration at any given point in time.

This database is critical for the analytics component, which relies on accurate topology information to evaluate the impact of proposed changes. When a new scheduled change is approved, the resulting topology is computed and stored as a future state. In the event of an unexpected disruption, such as a failure or disaster, the database is updated to reflect the new current topology, and all future scheduled states are recalculated accordingly.

Change Scheduler Analytics

The analytics component is responsible for evaluating the feasibility and impact of scheduled and unscheduled changes within the network. Its operation begins by extracting the time associated with a proposed action and identifying any other actions scheduled for the same time interval.

It then extracts the network topology prior to the action timestamp from the topology database, which stores historical and future states of the network along with their corresponding timestamps.

Using this information, the component interacts with the NDT, which computes the resulting topology after applying all actions scheduled for that time. It performs a comprehensive analysis to detect potential issues such as disconnected segments, routing loops, or any other condition that may compromise network stability.

Based on the analysis, the NDT communicates the outcome to the analytics:

- If the change is considered viable, the component approves the action, updating the scheduled changes database, and records the resulting topology in the topology database for the specified time.
- If a conflict is detected, the component rejects the action and provides detailed information about the conflicting operations to the management system.

Moreover, the analytics is triggered not only by the input component, but by the network topology itself. The analytics module periodically monitors the status of the network to ensure that unexpected or not scheduled changes do not negatively impact future schedule operations. In case of an unexpected disruption, such as a disaster or equipment failure, it will re-evaluate all future changes against the new topology. To do that, it will go through the scheduled changes database, selecting the actions in the order they were requested (not when they are scheduled) and recomputing the expected topology produced by each action.

From this point, the process is the same as in the previous case. If conflicts arise, the analytics component notifies the rejection of some specific action, indicating which other actions are producing the conflict.

4.3.1.2 Support to Integrated Services

The change scheduler integrates mainly into three of the integrated services outlined in Section 5.3. Specifically, as an NDT application, it is a key app for the two NDT integrated services, as well as the Emergency Network service.

Within the Emergency Network Service, the Change Scheduler contributes to both the incident and recovery phases of operation. During the incident phase, triggered for example by disasters or large-scale failures, the Change Scheduler supports the real-time detection of topology changes and evaluates their impact on already scheduled network operations. By continuously reassessing future scheduled actions against the updated topology, it prevents the execution of changes that could further degrade connectivity, create routing loops, or disrupt critical services. During the recovery phase, the Change Scheduler enables the controlled and phased re-introduction of network resources, assisting operators in scheduling recovery actions in a coordinated manner. This allows the prioritization of essential and mission-critical services while gradually restoring full network capacity, ensuring stability throughout the recovery process.

In the Digital Twin services of UNITY-6G, including both the Digital Twin for the Integrated 6G Network and the Digital Twin for the TSN Network, the Change Scheduler operates as an NDT application that enables an accurate and time-consistent modeling of network evolution. By

providing validated schedules of accepted changes and by dynamically recomputing future topologies when unexpected events occur, the Change Scheduler ensures continuous consistency between the real network and its digital representation. Through its interaction with the NDT, it allows the Digital Twin to account for planned and unplanned topology changes across domains. This capability is essential for predictive analysis and what-if evaluations, as well as for preserving bounded latency and service continuity for time-critical applications, all of which rely on a reliable temporal view of network topology evolution.

Across all these services, the Change Scheduler acts as an enabler of cross-domain integration, ensuring that network adaptations, planned or reactive, are executed in a consistent, conflict-free manner. By doing so, it strengthens the reliability, predictability, and operational robustness of UNITY-6G integrated services, particularly in dynamic scenarios where network topology evolves continuously.

4.3.2 App-02: Failure Prediction

4.3.2.1 Definition

The Failure Prediction and Root Cause Analysis application operates within the wireless transport segment of the integrated, heterogeneous network and follows the AI-native tuple approach composed of Monitoring, Analytics, Decision, and Actuation-related functions, as depicted in Figure 4-4. The Monitoring function collects and normalizes transport-domain observations, including performance measurements, alarms, operational state, topology information, configuration state, and relevant contextual indicators. These inputs are obtained from transport units and controllers using the available management and control interfaces, including NETCONF/YANG, ONF-based information models, and TeraFlowSDN exposure mechanisms. Where required, transport-domain adapters expose wireless-specific device information and capabilities to the controller, to the data continuum, and the service-based architecture. Real-time events and state updates are exchanged through dedicated interfaces, while historical measurements, telemetry sequences, alarm history, and contextual datasets are retrieved from the Data Continuum for analytics, model validation, and lifecycle improvement.

The Analytics function applies time-series analysis, machine-learning models, and complementary heuristic reasoning to infer whether the observed behavior indicates normal operation, degradation, or an evolving failure condition. The application uses LSTM-based models and additional sequence-processing mechanisms to capture temporal dependencies in transport KPIs, alarms, and device-health indicators. The models used by the application are trained and maintained by the AI/ML domain and made available to the application through a controlled model lifecycle mechanism, via the model repository and model package distribution process. This allows the application to receive updated prediction models when new data, new failure cases, topology changes, traffic-pattern changes, or region-specific operational conditions justify model retraining or adaptation.

The Analytics function produces structured probabilistic outputs, including the likelihood of failure, the estimated failure class, the probable root cause, the affected link or node, the expected time-to-impact, and the confidence level associated with the prediction. These outputs are then provided to the Decision function, which evaluates the analytics results against operational policies, anomaly severity thresholds, service impact, confidence values,

and the availability of mitigation options. The Decision function does not simply forward raw model outputs; it consolidates the results of the analytics process into an operationally meaningful conclusion, such as whether the event should be reported, escalated, monitored, or passed to another control function for preventive action.

The Actuation-related output of this application is therefore not necessarily a direct configuration change. Rather, the application exposes a structured failure-prediction event or mitigation request toward another agent, controller, or service-assurance function that is responsible for executing corrective or evasive actions. Such actions may include rerouting, protection switching, transmission-power adjustment, service migration, maintenance triggering, or additional diagnostics, depending on the affected service and the authority of the receiving control entity. In this way, the application contributes to a closed-loop assurance process while maintaining a clear separation between prediction, decision support, and actual network reconfiguration.

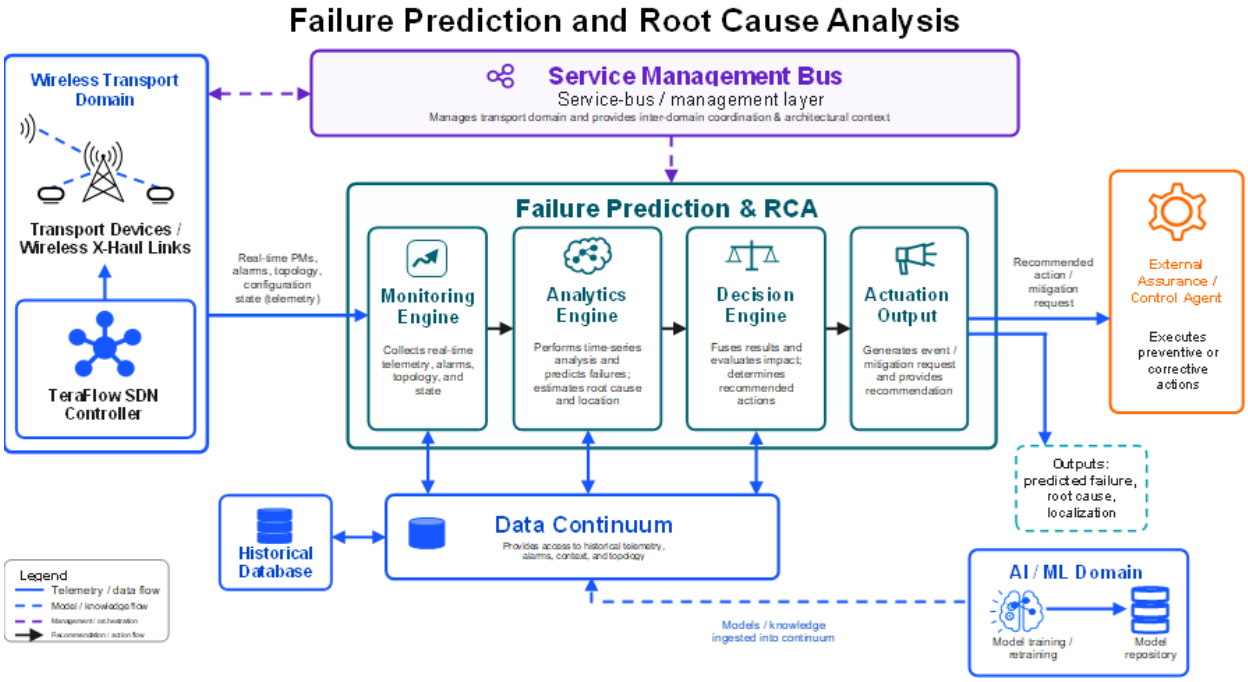

Figure 4-4: Failure Prediction

4.3.2.2 Support to Integrated Services

The Failure Prediction and Root Cause Analysis application integrates seamlessly into several of the integrated services outlined in Section 4.2. Specifically, its predictive capabilities are crucial for ensuring the reliability and resilience of Emergency Network Services, TSN services, and XR Semantics Orchestrated Services.

In the context of Emergency Network Services, the app's ability to predict and localize failures in advance is critical for maintaining uninterrupted communication during emergencies. By providing early warnings and root cause insights, the app allows emergency services to preemptively reroute traffic, allocate resources, and maintain network availability when it matters most. For TSN services, which require ultra-reliable and low-latency communication, the app ensures that any impending disruptions are identified early. This allows for timely

interventions, such as reconfiguring paths or adjusting resources, to uphold the stringent performance requirements of TSN services.

Additionally, the app supports XR Semantics Orchestrated Services by ensuring that the underlying network infrastructure remains robust and capable of delivering the high bandwidth and low latency necessary for immersive experiences. By predicting and mitigating potential failures, the app ensures consistent service quality and user experience in XR applications.

Through its integration with these critical services, the Failure Prediction and Root Cause Analysis application enhances the overall resilience and performance of the network, ensuring that each service can meet its specific operational demands and deliver consistent, high-quality connectivity.

4.3.3 App-03: Wireless xHaul Reconfiguration

4.3.3.1 Definition

The Wireless xHaul Reconfiguration application operates within the wireless transport segment of the integrated, heterogeneous network and provides an optimization-oriented control function for adapting wireless xHaul connectivity according to service demand, transport-domain state, predicted degradation, and service-level requirements. The application follows the AI-native tuple approach composed of Monitoring, Analytics, Decision, and Actuation-related functions. Its objective is to evaluate the current condition of wireless transport links and determine whether the active xHaul configuration should be maintained, modified, or replaced by an alternative configuration that better satisfies the operational and service requirements.

The Monitoring function collects and normalizes transport-domain observations, including link performance measurements, alarms, topology information, available capacity, utilization, traffic demand indicators, operational state, configuration state, and relevant device capabilities. These inputs are obtained from wireless transport units and domain controllers using the available management and control interfaces, including NETCONF/YANG, ONF-based information models, and TeraFlowSDN exposure mechanisms. Where required, transport-domain adapters expose wireless-specific information that may not be natively represented by the controller, including radio-link status, channel conditions, link capacity, transmit-power configuration, modulation behavior, and topology relations between wireless transport nodes. Real-time state and demand indications are made available to the application through the service-based architecture, while historical telemetry, traffic evolution, alarm history, topology evolution, and previous reconfiguration outcomes are retrieved from the Data Continuum for analysis and model improvement.

The Analytics function evaluates the state of the wireless xHaul network using optimization algorithms, machine-learning models, traffic-demand analysis, graph-based reasoning, and heuristic logic. A central capability of this function is the use of dynamic routing mechanisms, including the DPIR-oriented [32] approach for transport optimization, in which the network is represented as a graph whose links are weighted according to operational conditions such as capacity, utilization, delay, degradation risk, service priority, and predicted traffic demand. In this context, dynamic routing is not limited to shortest-path computation; it evaluates candidate paths according to the expected service impact, the state of the wireless links, and the

resilience requirements of the transported services. The application therefore uses DPIR-style routing logic to identify alternative paths, avoid vulnerable or overloaded links, and support traffic steering before service degradation becomes visible.

The application also operates in conjunction with an intent-based transport configuration. In this mode, the desired transport behavior is expressed as a high-level objective, such as maintaining capacity for a service class, preserving low latency for time-sensitive traffic, improving resilience for an emergency service, or reducing the use of unstable links. The Wireless xHaul Reconfiguration application translates these objectives into candidate configuration actions by combining the intent constraints with the actual transport-domain state. This allows the system to move from a purely imperative configuration model, in which specific routes or power values are manually selected, toward a more declarative model in which the required service outcome guides the selection of routing, connectivity, and wireless-link parameters.

The Analytics function makes use of time-series forecasting to estimate near-future demand, graph-based methods to evaluate candidate paths and link alternatives, and optimization logic to assess possible configuration changes. These methods are complemented by AI/ML models trained and updated by the AI/ML domain and made available to the application through the model repository and controlled model lifecycle mechanism. Such models may support demand prediction, link-degradation estimation, path-risk scoring, or reconfiguration-impact estimation. The Data Continuum supports this process by providing access to historical telemetry, prior reconfiguration decisions, previous network states, and the observed effect of past actions. This enables the application to improve its recommendations when traffic patterns, service distributions, topology conditions, or operational policies change.

The Analytics function produces structured outputs describing the evaluated transport state and the candidate reconfiguration options. These outputs may include predicted demand pressure, affected links or paths, expected bottlenecks, candidate route alternatives, candidate link-connection changes, candidate transmit-power adjustments, and the estimated impact of each option on capacity, latency, resilience, and service continuity. In a DPIR-oriented implementation, the output includes ranked routing alternatives and the reasoning behind their selection, such as avoidance of links with high degradation probability, reduction of congestion, or improved support for a service-specific latency requirement. In an intent-based implementation, the output includes the mapping between the expressed intent and the proposed configuration action, allowing the decision process to verify that the recommendation remains aligned with the service objective.

These analytics results are then provided to the Decision function, which evaluates them against operational policies, service priorities, device constraints, guardrails, topology feasibility, and the potential impact on active services. The Decision function consolidates the candidate options into an operational recommendation, such as maintaining the current configuration, rerouting specific flows, adjusting wireless link parameters, modifying link connectivity, or requesting a broader transport-domain reconfiguration. The Decision function is also responsible for resolving trade-offs between competing objectives, for example when a path with better latency has lower resilience, or when increasing transmit power may improve a link but conflict with energy or interference constraints.

The Actuation-related output of this application is a structured reconfiguration recommendation or control request, rather than an uncontrolled direct change to the network. Depending on the deployment policy and authority granted to the application, this output may be sent to TeraFlowSDN, a transport-domain controller, an orchestration function, or another agent responsible for executing and validating network changes. The action may include a new routing suggestion, a proposed change in wireless xHaul connectivity, a transmit-power adjustment, or a request to prepare alternative paths before service degradation occurs. The receiving control entity is responsible for translating the recommendation into device-level configuration actions, enforcing policy constraints, and validating that the resulting configuration remains consistent with the service objectives and transport-domain safety requirements.

In this way, the Wireless xHaul Reconfiguration application contributes to closed-loop transport optimization while maintaining a clear separation between observation, analysis, decision support, and configuration enforcement. It complements the Failure Prediction and Root Cause Analysis application by using transport state, demand information, intent constraints, and potentially predicted failure events to determine how the wireless xHaul network should adapt. While the failure-prediction application identifies what may fail, why, and where, the reconfiguration application evaluates how the transport domain can be adjusted through dynamic routing, intent-driven configuration, connectivity adaptation, and wireless-parameter optimization in order to preserve service continuity, optimize resource utilization, and improve resilience.

4.3.3.2 Support to Integrated Services

The Wireless xHaul Reconfiguration application supports the integrated services described in Section 4.2 by enabling the transport segment to adapt dynamically to service requirements, network conditions, and predicted or observed degradation. Integrated services such as Emergency Network Services, XR Semantics Orchestrated Services, Multi-RAT Time-Sensitive Services, and Digital Twin-based network services depend on reliable and appropriately dimensioned transport connectivity. The application provides the transport-domain adaptation capability required to maintain these services when demand changes, when wireless links degrade, or when the topology must be modified to preserve continuity.

In the context of Emergency Network Services, the application can support the rapid adaptation of the wireless xHaul network when parts of the network become overloaded, degraded, or unavailable. During emergency conditions, traffic demand may shift abruptly and specific communication paths may become more critical than others. The application can use dynamic routing mechanisms, including DPIR-style path evaluation, to identify alternative transport paths that preserve emergency-service connectivity even when some links become unreliable or unavailable. When combined with intent-based transport configuration, emergency traffic can be associated with explicit objectives such as maintaining reachability, prioritizing critical flows, or avoiding predicted failure zones. The application can then propose rerouting, link-connection changes, or wireless-parameter adjustments that support these objectives.

For XR Semantics Orchestrated Services, the application supports the maintenance of high-throughput and low-latency transport paths required by immersive and interactive services. XR services are sensitive to changes in delay, jitter, packet loss, and available capacity. The

reconfiguration application can identify when the current wireless xHaul path is becoming unsuitable for the service and propose alternative routing or link adjustments that maintain the required transport behavior. Intent-based configuration is especially relevant in this context, since the service requirement can be expressed in terms of target latency, throughput, or continuity rather than a fixed route. The reconfiguration application can then evaluate candidate xHaul configurations against those requirements and expose the most suitable recommendation to the control layer.

For Multi-RAT Time-Sensitive Services, the application provides a transport-aware mechanism for preserving deterministic or near-deterministic service behavior. These services require strict control of latency, reliability, and traffic continuity across heterogeneous network segments. The application can evaluate whether the active wireless xHaul configuration supports the required timing behavior and can recommend reconfiguration when a path becomes congested, unstable, or exposed to degradation. DPIR-oriented dynamic routing can be used to assess candidate paths according to time-sensitive constraints, while intent-based configuration can represent the required service behavior as a set of transport objectives and constraints. This enables the application to recommend changes that are not only topologically feasible, but also aligned with the timing and reliability assumptions of the service.

The application can also support Digital Twin for Integrated 6G Network and Digital Twin for TSN Networks by providing candidate reconfiguration options and transport-domain state information that can be evaluated in a simulated or emulated environment before enforcement. The Data Continuum can provide historical telemetry and previous reconfiguration outcomes to the digital twin, while the reconfiguration application can consume digital-twin evaluation results to select safer or more effective actions. This is particularly relevant for DPIR and intent-based transport management, because candidate routing and configuration decisions can be evaluated against predicted traffic demand, topology changes, and service constraints before they are applied to the operational network.

Overall, the Wireless xHaul Reconfiguration application integrates into the service lifecycle as a transport-domain adaptation function. During service deployment, it can support path selection and initial xHaul resource allocation based on service intent. During service operation, it can evaluate whether the transport configuration remains suitable for active service demand. During service assurance, it can recommend evasive or corrective actions when degradation is detected or predicted. Together with the Failure Prediction and Root Cause Analysis application, it enables a predictive and adaptive transport-management capability in which service continuity is maintained not only by detecting failures, but also by dynamically adapting the wireless xHaul configuration through routing optimization, intent-driven configuration, and controlled transport-domain actuation.

4.3.4 App-04: Energy-aware load balancing

4.3.4.1 Definition

The Energy-aware load balancing application operates within the RAN domain. Its primary objective is to dynamically optimize the distribution of computing tasks and user associations across network nodes. Unlike traditional load balancers that focus solely on throughput or hardware utilization, this application employs a tunable utility/cost function that allows network

operators to align infrastructure behavior with specific strategic goals, such as environmental sustainability or service resilience.

The application utilizes a probabilistic user association policy. This is paired with an adjustable transmit power or operating point for each base station, which determines both its instantaneous energy consumption and aggregate radio capacity. The application operates in two distinct modes:

1. During disaster scenarios (UC1), the application is a critical enabler of service continuity. It detects when infrastructure is lost or when traffic spikes due to emergency coordination. By shifting from an "Energy-aware" posture to a "Resilience-first" posture, the application automatically rebalances the remaining active nodes to provide maximum coverage. It ensures that critical emergency traffic is associated with the most stable radio links, even if those links are computationally expensive or energy-inefficient.
2. For services explicitly designed to reduce the carbon footprint of 6G networks, this application serves as the primary execution engine. It uses the Data Continuum to predict low-traffic windows and proactively suggests "sleeping" specific RAN components while reassociating active users to more efficient hardware. This supports the broader goal of ‘green networks’, ensuring that energy savings do not come at the cost of violating SLAs.

The function exposes a REST interface (or equivalent) for communication with orchestration and monitoring layers. The inputs include base-station energy models, residual energy states, predicted user-density fields, and scenario-specific QoS targets. The outputs are the user association strategy and a base station power/operating schedule.

4.3.4.2 Support to Integrated Services

The Energy-aware load balancing application acts as a foundational "Efficiency and Resilience" controller for the integrated services described in Section 4.2. By bridging the gap between hardware energy constraints and high-level service requirements, it ensures that the access segments remain efficient under varying conditions.

The application supports an adaptive control loop. As the network state evolves (e.g., changes in user distribution or loss of power at specific nodes), the algorithm re-optimizes the configuration over a shifted horizon, ensuring that service continuity is maintained even under fluctuating conditions.

The application supports Digital Twin-based services by providing its utility-function outcomes for "what-if" simulations. Operators can test a new energy-saving weight in the Digital Twin to see the predicted impact on user experience before the Energy-aware load balancing app applies to the change to the live production network.

4.3.5 App-05: XR Application

4.3.5.1 Definition

Hololight Space is an industrial Extended Reality (XR) application enabling immersive, real-time, multi-user collaboration and visualization within distributed environments. The application

supports holographic rendering and interaction with complex 3D models, digital twins, spatial annotations, sensor overlays, and engineering data in a shared virtual workspace. Within the UNITY-6G project, Hololight Space is deployed as a server-side XR application running on GPU-enabled virtual machines (VMs) located at edge or cloud infrastructures, while the XR client application executes on end-user XR devices such as Meta Quest or Apple Vision Pro headsets.

The application operates primarily at the application and service layer of the UNITY-6G SBMA, while interacting closely with orchestration, monitoring, AI inference, and cellular network function components. Within the UNITY-6G architecture, Hololight Space operates as a vertical application. As described in [4], the architecture exposes two entry points for vertical application lifecycle management: the IDMO, which provides direct orchestration control, and the CAMARA API exposure layer, which offers a standardised northbound interface for vertical domains to request network resources and trigger service lifecycle operations without requiring direct interaction with internal orchestration components. Hololight Space can leverage either entry point, depending on the deployment context and operator preference. Hololight Space utilizes Hololight Stream for low-latency pixel streaming between the server-hosted XR application and the XR client device through the underlying cellular infrastructure created with the help of UNITY-6G architecture. The application supports adaptive streaming quality based on real-time network feedback, multi-user synchronization across distributed XR clients, and telemetry-driven integration with monitoring and analytics services.

The application interfaces include:

- XR client interfaces for immersive visualization and user interaction
- Streaming interfaces for real-time video and input transmission
- Telemetry interfaces exposing metrics such as latency, bitrate
- Multi-user synchronization interfaces for shared spatial collaboration across geographically distributed XR clients
- Northbound APIs for integration with monitoring services (***App-13 KPI monitoring xApp***), failure analytics (***App-02 Failure Prediction***), Digital Twin services (4.2.3), and AI-driven orchestration components

The application is designed to support low-latency, high-fidelity XR experiences over heterogeneous 5G/6G, edge-cloud, transport, and satellite-enabled network infrastructures.

4.3.5.2 Support to Integrated Services

Within UNITY-6G, Hololight Space acts as the primary immersive XR application integrated into two key integrated services defined in Section 4.2: the XR Semantic Service (§4.2.2) and the Digital Twin for Integrated 6G Network service (§4.2.3). The application interacts with multiple intelligent applications and network functions across these services to enable semantic-aware, adaptive, and energy-efficient holographic XR communication.

The XR workflow begins when a user initiates an XR session through the Hololight Space client application. An orchestration service allocates the appropriate compute and network resources, including GPU-enabled edge/cloud virtual machines, network slices, and communication resources. Once provisioned, the Hololight Space server instance is launched and connected to the XR client through Hololight Stream.

During runtime, the application continuously exchanges telemetry and operational metrics with monitoring and analytics services. These metrics include latency, bitrate, framerate, GPU utilization, user interaction events, and network conditions. The collected data is processed by monitoring systems and AI-driven analytics engines to support closed-loop optimization and intelligent decision-making.

Integrated orchestration and decision services dynamically adapt compute and communication resources based on application and network conditions. This includes:

- Dynamic GPU and CPU allocation
- Adaptive XR bitrate and streaming optimization
- Multi-RAT and network path optimization
- Edge-cloud workload migration
- Energy-aware load balancing
- Transport and xHaul resource reconfiguration

Hololight Space is additionally integrated with Digital Twin services to support simulation, validation, automated testing, and experimentation within controlled 6G environments prior to operational deployment.

Through this integration, Hololight Space serves as the reference high-demand XR application for validating the capabilities of the XR Semantic Service (4.2.2) and contributing to the Digital Twin service (4.2.3) within UNITY-6G. Specifically, App-05 validates the end-to-end semantic communication pipeline described in 4.2.2, the delivery of holographic content under constrained bandwidth conditions, and the dynamic resource orchestration that the UNITY-6G SBA provides for immersive applications. The application thus acts as the primary proof-of-concept vehicle for HOLO-driven holographic XR over integrated 6G networks.

4.3.6 App-06: Multi-RAT TSN optimization

4.3.6.1 Definition

Multi-RAT TSN optimization application operates within the radio access network domains (O-RAN and non-3GPP RAN) of the unified network architecture, to control and maintain the time sensitive nature of the network based on the demands of time critical traffic flows. Given different sets of the TSN features on each of the RAN domains, this application can control each of them depending on the context and the domain it operates. In the context of a joint cross-RAT optimization, the same app logic can be used to manage and control both RATs (cellular and IEEE 802.11), while in case of the siloed-deployment RAT domains, the same app logic can be re-used across domains.

For the IEEE 802.11 RAT the two main features that the app will control are the scheduling mechanism between the AP and the stations, as well as the ContinousMLO (ConMLO) mechanism of the AP. In addition, for highly time-sensitive traffic flows it will support also dynamic traffic distinguished based on DSCP values for enabling traffic steering on ConMLO. The channel access delay in ConMLO will depend on the overlap between the time slots in each of the links. As such, this application controls the overlap between the timeslots assigned to each of the links in ConMLO as a percentage of the time slot length. In certain cases, this might include adding a new additional time slot in the links to account for the overlap.

The application will operate in different modes based on the presence of the time critical traffic flows in the network and other best effort traffic flows. Time critical traffic flows can use (i) a dedicated time slot in a single link, (ii) a shared timeslot in multiple links utilizing ConMLO for fast channel access, or even (iii) dedicated timeslots with overlap between time slots utilizing ConMLO. The overlap will be determined statistically based on the channel load in each of the links as well as the channel access delay requirements of the time critical traffic flow.

In non-3gpp domain, this application will use E2 interfaces to control scheduling as well as ConMLO schedule assignments and overlap. The E2 interface will pass the time-slot sizes and overlaps, while the actual schedule will be created in the IEEE 802.11 AP itself. Once the schedule is updated in the AP, the schedule needs to be updated in the stations as well. This will be triggered by the AP itself. Internally, the E2 interface actions will be mapped to the openwifi interfaces that apply the schedule to the interface itself.

For the O-RAN domain the application will get monitoring information from the monitoring app. Such information include latency related parameters for critical traffic flows such as TB Error Rate, HARQ retransmissions, TB latency, etc. This information will be used for controlling the slicing configurations of critical traffic flows in (near) real-time by adjusting the dedicated, prioritized and shared resource blocks. In addition, within a slice, scheduling policies will be monitored and modified accordingly, so that time-sensitive traffic flows get prioritized in order to maintain the one-way latency under certain threshold. Techniques such as UL prescheduling, UL configured grant, use of mini-slots and deep reinforcement learning models will be used to optimize the radio resource management in real-time.

4.3.6.2 Support to Integrated Services

The multi-RAT TSN optimization application is the main application that supports Multi-RAT Time-Sensitive Service together with monitoring services. The application will predict and anticipate the degradation of the time sensitivity on certain domain based on the monitoring information from the monitoring application and react to maintain the time sensitivity. It cannot be initiated in the network without any monitoring application for each network domain.

4.3.7 App-07: Traffic engineering for TN-NTN switching

4.3.7.1 Definition

The traffic engineering for Terrestrial Network- Non-Terrestrial network (TN-NTN) switching component is based on the Smart Gateway (SGW) component proposed within the 5G-GOVSATCOM [28] and 5G-HUB project [29]. The SGW component supports multi-connectivity and enables traffic steering, with the aim of ensuring continuous reliable service during transitions between TN and NTN coverage areas. This capability is particularly important in case of a disaster where the TN network can be severely damaged, using the NTN segment as the alternative path which can also be used for crisis management and humanitarian aid teams, where the demand for resilient, adaptive communication is paramount.

The SGW component follows a client-server-based architecture to enable the vertical handover (i.e., switching) between TN and NTN 5G connectivity. The key technology supporting this scheme is the implementation of dynamic routing/switching mechanisms (inspired by the virtual routers described in [30]) located on both the server and client sides,

as depicted in Figure 4-5. These SGWs offer a generic framework that could consider 5G network-related parameters or application (APP)-related criteria to ensure optimal route/connectivity decision-making. More specifically, this framework consists of:

- Smart Gateway Client (SGW-C): The smart gateway client resides on the user side, enabling dynamic switching between the TN and NTN 5G networks. It acts as the interface between the client device (typically running applications, such as a video server) and the 5G network connectivity, managing satellite and terrestrial connections. It can be regarded as a hardware component connected to a 2-UE device, where one UE interfaces through the TN access network and the other through the NTN access network. The client's IP address, assigned beyond the SGW-C (beyond the UEs), remains the same throughout the vertical handover process. This IP address is crucial to maintaining reliable connections during the handover between networks, ensuring that the client's connectivity with the application servers remains unaffected by the transition. The SGW-C utilizes the IP addresses assigned by the TN and NTN connections as relay node addresses. These addresses are dynamically selected based on network availability and other decision-making criteria, but they remain hidden from the application layer, thereby simplifying the communication process.
- Smart Gateway Server (SGW-S): The smart gateway server should be located on the application server side to ensure that the application server can communicate with the application client by abstracting the complexity of the underlying networks. A crucial aspect of the SGW-S is that it leverages site-to-site Virtual Private Networks (VPNs) to create secure tunnels through either the NTN or TN, enabling seamless data transmission without revealing the user client's actual IP address. This is particularly relevant since UEs typically operate as clients and are not designed to listen to incoming connections as servers do. This architecture provides security by abstracting the UE IP address behind the gateway while preserving communication flexibility. SGW-S, therefore, acts as the gateway of the application server, allowing it to reach the application client by targeting the static IP address of the application client, which is always the same regardless of the selected route. The gateway selects the optimal path (NTN or TN VPN) based on predefined criteria.

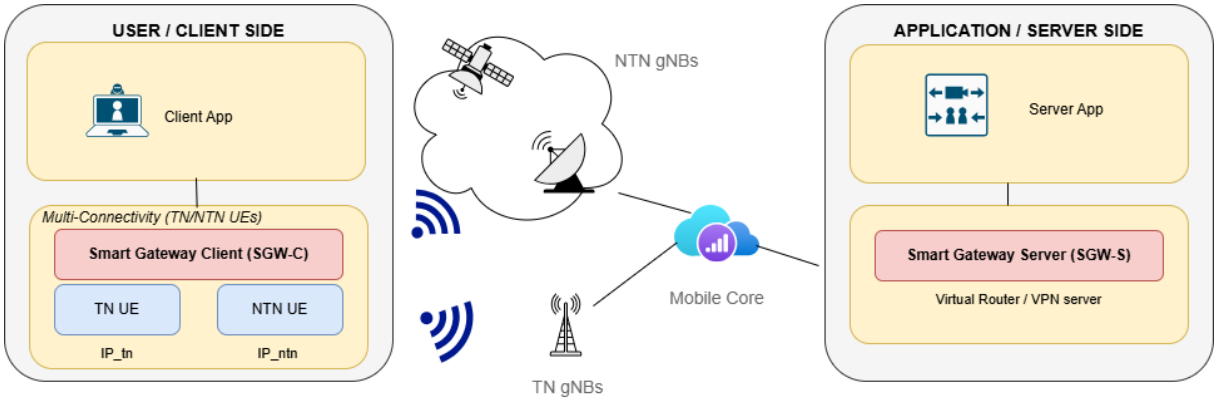

Figure 4-5: SGW concept

More specifically, dynamic switching is obtained with the cooperation of SGWs on both sides, allowing the application server and the application client to communicate using the same

private IP addresses while the actual routing decisions are abstracted by the SGW infrastructure (i.e., transparent to the application on both sides). The key to this seamless communication is the use of VPN tunnels, which are set up between the SGW-C and SGW-S, featuring:

- Site-to-Site VPNs: These tunnels ensure secure, encrypted communication between the application server and client without exposing the underlying IP addresses of the UEs (i.e., TN, NTN). The VPNs also abstract the complexity of the TN and NTN networks, allowing for uninterrupted connectivity during vertical handover.
- Relay Node IPs: IP addresses assigned to the TN UE and NTN UE are used as relay addresses between the client and server applications. These IP addresses are employed only as intermediate points for routing without being exposed to the application layers.

This architecture ensures that the server and client maintain consistent communication by always targeting the same IP addresses, simplifying the routing process while also providing certain degree of security. The SGWs make it safer and easier to switch between networks, as the application server only needs to send data back to the application client private IP, and the decision on which path to use (terrestrial or satellite) is handled by the SGW-S.

4.3.7.2 Support to Integrated Services

The Traffic engineering for TN-NTN switching component is addressing the provision of alternate mobile network (i.e., 5G) access connectivity to TN through a fallback active NTN connection in case of failure or leaving the coverage area. Based on that, the traffic steering is an element for a network deployment prepared for handling emergencies and disasters, e.g., the Emergency Network Service. In case of a disaster, the end user will still be able to have cellular connectivity if the end user is equipped with the SGW-C and the network deployment has allocated the appropriate satellite resources and creating the appropriate NTN mobile network components (e.g., NTN DU) running in parallel to TN components.

4.3.8 App-08: Semantic Domain Switching for XR Streaming over TN-NTN Networks

4.3.8.1 Definition

XR streaming places strict and continuous demands on the underlying network: latency must remain below perceptual thresholds (extensive experimentation results have shown that typical latency values should not exceed 50 ms) to prevent motion sickness and maintain pose responsiveness, and these requirements must be met even as network conditions fluctuate. It should be noted that relevant values of the delay jitter will be investigated in the upcoming rounds of testing. In multi-domain deployments where both TN and NTN connectivity are available, such as emergency response, remote industrial inspection, or disaster recovery scenarios, the system must decide in real time which network path best satisfies the XR application's QoS requirements. As shown in Figure 4-6, the Semantic Domain Switching component for XR streaming provides application-layer-driven vertical handover between TN and NTN network domains, extending the Smart Gateway (SGW) architecture described in App-07 with an intelligence layer that uses the XR stream's own latency signal as the sole decision criterion. The component operates at the interface between the XR application layer

and the SGW infrastructure, enabling proactive domain selection without modifying the encoder, the rendering pipeline, or the underlying network stack, as shown in Figure 4-6.

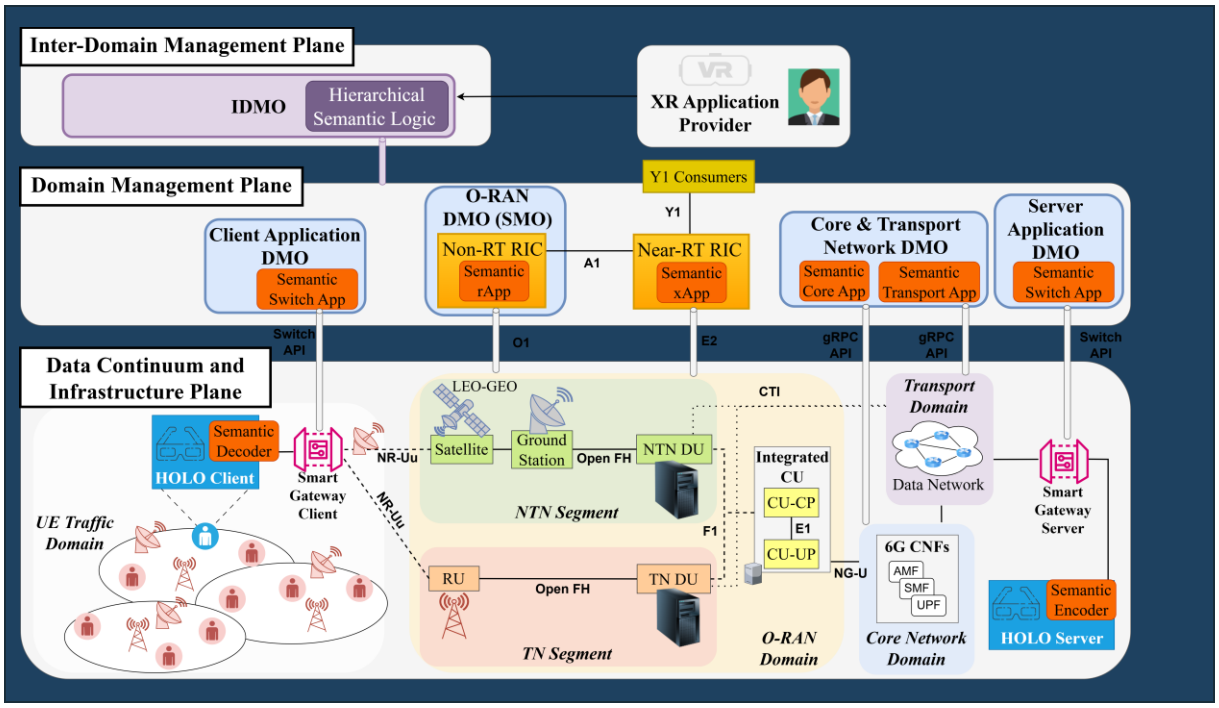

Figure 4-6: Semantic-aware O-RAN UNITY-6G architecture for TN-NTN

The component provides the following functionalities:

- *Latency monitoring:* the XR client continuously measures end-to-end stream latency at the application layer and computes a rolling average over a short observation window (5 samples).
- *Congestion regime classification:* the semantic controller classifies the current network condition into one of three regimes, high-load, medium-load, or low-load, corresponding to the capacity characteristics of the active domain.
- *Proactive handover triggering:* when the rolling average latency exceeds a configurable warning threshold (e.g. default 45 ms), the controller issues a handover request to a less congested domain before the latency violation becomes perceptible to the user. Escalation to the most constrained domain is triggered at a severe threshold (e.g. default 80 ms). Recovery to a higher-capacity domain is initiated once latency stabilises below a recovery threshold (e.g. default 38 ms) for a minimum hold period.
- *Session continuity:* handover decisions are forwarded to the SGW-Server (SGW-S), which activates the corresponding VPN tunnel, TN or NTN, transparently to the application layer. The XR client and rendering server retain the same private IP addresses throughout, ensuring uninterrupted session state during the domain transition.
- *Encoder independence:* the XR encoder maintains its target bitrate of 80 Mbps regardless of the active domain. The semantic response is exclusively a network path selection decision, not an encoder or compression rate adjustment.

The component interfaces with the SGW-Client (SGW-C) on the user side, which exposes the application-layer latency measurements to the semantic controller, and with the SGW-S on the application server side, which executes the VPN tunnel activation corresponding to the selected domain. The decision logic resides within the SGW-C, keeping the application server agnostic to the handover mechanism. The three domain states and their corresponding network capacity levels are summarized in Table 1 below.

Table 1: Semantic domain switching states and handover triggers.

Domain state	Network path	Trigger
High-load	TN (highest capacity)	Latency < 38 ms (recovery)
Medium-load	TN or NTN (intermediate)	Latency ≥ 45 ms (warning)
Low-load	NTN (fallback)	Latency ≥ 80 ms (severe)

4.3.8.2 Support to Integrated Services

The Semantic Domain Switching component addresses the provision of resilient XR streaming connectivity in scenarios where TN infrastructure is degraded or unavailable, by enabling the XR application to autonomously select the most appropriate network domain based on observed QoS conditions. This makes the component directly applicable to integrated services that combine TN and NTN access, including the Emergency Network Service and crisis management deployments described in App-07.

In an emergency or disaster scenario, TN infrastructure may be partially damaged, resulting in increased congestion and latency on terrestrial links. The semantic controller detects this degradation through the XR latency signal and proactively initiates a handover to the NTN domain before the stream quality deteriorates to a perceptually unacceptable level. This is a key distinction from conventional SGW operation, which relies on network-layer triggers such as link failure or coverage loss: the semantic approach also acts on application-layer observations and therefore responds earlier and more precisely to conditions that affect XR user experience.

Evaluation on real XR streaming measurements collected under controlled network congestion conditions demonstrates the effectiveness of the approach. At the 50 ms latency target, the domain switching controller reduces the violation rate from 28.7% to 6.3%, a factor of 4.5 improvement. For severe violations above 80 ms, which correspond to motion sickness and pose instability in XR, the reduction is from 16.1% to 1.8%, a factor of 8.8. The system spends 28.9% of session time on the highest-capacity domain at the default warning threshold of 45 ms, rising to 45-46% when the threshold is relaxed to 50-55 ms at the cost of higher violation rates. The sensitivity of the controller to the warning threshold setting, providing operators with a configurable tradeoff between QoS strictness and domain utilisation efficiency. The component is particularly suited to the following integrated service scenarios: Remote XR-assisted emergency response: field teams using XR headsets for remote expert guidance maintain stream quality during transitions between degraded TN and NTN coverage without manual intervention.

- *Digital twin visualisation in disaster zones:* real-time 3D environment streaming for situational awareness remains below perceptual latency thresholds even as network conditions change dynamically.

- *Humanitarian aid coordination:* XR-based training and collaboration tools maintain session continuity during vertical handovers driven by infrastructure availability rather than application-layer awareness.

4.3.9 App-09: Probabilistic forecasting app in TN-NTN

4.3.9.1 Definition

The purpose of this probabilistic forecasting rApp is to provide smart and proactive resource management in TN-NTN by forecasting future resource demand in a probabilistic manner. This application is motivated by the evolution of satellite communication systems toward fully flexible digital payloads, which allow operators to dynamically reconfigure satellite resources according to changing user demand, traffic profiles, beam-level load variations, and operational constraints.

Unlike traditional single-point forecasting methods that estimate only one expected value of future demand, this application aims to characterize the possible range of future resource requirements. It estimates the uncertainty associated with future demand and provides probabilistic outputs such as prediction intervals, confidence ranges, and quantiles. This enables the system to understand not only the most likely future demand, but also the level of risk associated with that prediction. These probabilistic outputs further support forecasting-based allocation by indicating how much resource should be reserved, adjusted, or redistributed under uncertain and highly variable traffic conditions.

In TN-NTN systems, these probabilistic forecasts may also be used by a common control and coordination layer to support resource management across TN and NTN segments. The application intends to forecast future demand together with uncertainty information and allows the system to decide how resources should be distributed, reserved, or adjusted between TN and NTN. By jointly considering these criteria, it can support proactive and risk-aware resource allocation across integrated TN-NTN systems.

The probabilistic forecasting rApp expected to provide key functionalities for receiving authorized network measurements, model training and retraining, probabilistic prediction, performance monitoring. It would process the received data and generates univariate and multivariate probabilistic forecasts of future resource demand and expose the output to other components and applications to support their tasks.

4.3.9.2 Support to Integrated Services

The integration of the probabilistic forecasting rApp within the integrated service framework aims to move beyond a standalone prediction capability and establish a complete prediction-to-decision workflow for practical service management. To support integration within an end-to-end O-RAN-based ecosystem, the rApp is expected to be containerized and deployed within the Non-RT RIC framework, where it can interact with monitoring sources, analytics functions, decision-support modules, and resource allocation mechanisms through dedicated interfaces

The integration framework utilizes Swagger/OpenAPI to provide a structured and user-friendly interface for interacting with the rApp backend and other network entities. Different components can query the application endpoints to retrieve rApp functionalities, including status monitoring, prediction information, performance metrics, and model weights as assisted information. By exposing these capabilities, the probabilistic forecasting rApp would be incorporated into a broader operational framework where network observations can be

converted into predictive insights and subsequently used to guide resource-management actions for other applications.

4.3.10 App-10: Conflict Management between Open RAN and IDMO

4.3.10.1 Definition

The framework provides a hierarchical conflict resolution algorithm for multi-domain O-RAN environments. It consists of conflict resolution modules deployed in DMOs and a supervisory module in the IDMO. Communication between modules is implemented using Apache Kafka. The framework detects both intra-domain and inter-domain conflicts and resolves them based on network parameters, KPIs, or through negotiation between DMO and IDMO components. More details can be found in [5].

4.3.10.2 Support to Integrated Services

The Conflict Management App can be used across the four described integrated services, since conflicts can arise inside the O-RAN domain (in case that multiple intelligent applications are operating simultaneously with conflicting objectives), but also between the O-RAN domain and the IDMO. The latter inter-domain conflict management supports multi-tenant scenarios, since the IDMO may provide conflicting objectives to the O-RAN domain or between different O-RAN domains. For instance, an additional deployed network slice for emergency communications may be prioritized from the IDMO perspective, while xApps in the O-RAN domain that accommodates the normal traffic will experience conflicts and reduced performance.

4.3.11 App-11: Trust Modelling to enable trustworthy interactions between IDMO/DMO and intra DMO components using DLT

4.3.11.1 Definition

This app introduces a trust modelling approach that enables trustworthy interactions across administrative domains (IDMO-DMO), addressing the limitations of implicit trust assumptions in multi-operator, multi-vendor 6G orchestration environments. Leveraging distributed ledger technology, it provides a foundation for establishing verifiable trust between interacting entities without relying on a centralized authority. This approach improves accountability and traceability of cross-domain interactions, increases resilience against non-compliant or malicious participants, and supports scalable, automated trust decisions aligning with the broader security expectations of next-generation network orchestration.

4.3.11.2 Support to Integrated Services

The trust modelling App acts as a foundational, cross-cutting capability that other inter-domain orchestration services can rely on to assess the trustworthiness of participating domains and components before engaging in collaborative actions. By providing a verifiable and auditable trust basis, it supports services such as multi-tenant resource allocation in making more informed and accountable decisions when interactions span multiple administrative domains, reducing the risk of relying on unverified or compromised peers.

4.3.12 App-12: LLM based monitoring app for O-RAN

4.3.12.1 Definition

We propose an adaptive monitoring system for O-RAN that integrates intelligent query handling mechanisms using Large Language Models (LLMs). The system aims to streamline data visualization and adapt to dynamic network conditions and user requirements. Our approach dynamically converts user intent into low-level O-RAN component APIs and other infrastructure tools, validating the system through deployments with both software-based and RF front ends. The results demonstrate the system's capability to inject advanced intelligence into various network tasks, optimizing performance and efficiency.

LLMs have emerged as powerful text processing tools that inject intelligence into various domains, including natural language understanding, generation, and translation. Recent versions of LLMs have expanded beyond text to support multiple modalities such as video, audio, and voice, enhancing their versatility and applicability. In the context of O-RAN, the majority of interactions between different components are managed through configurations in YAML files, software specifications, and infrastructure running on cloud-native tools like Kubernetes. By integrating LLMs, O-RAN and future 6G networks can benefit from advanced intelligence, optimizing configurations, automating processes, and improving overall network performance. This infusion of AI at the core of network operations promises to enhance efficiency, adaptability, and the user experience in next-generation mobile networks, as presented in Figure 4-7.

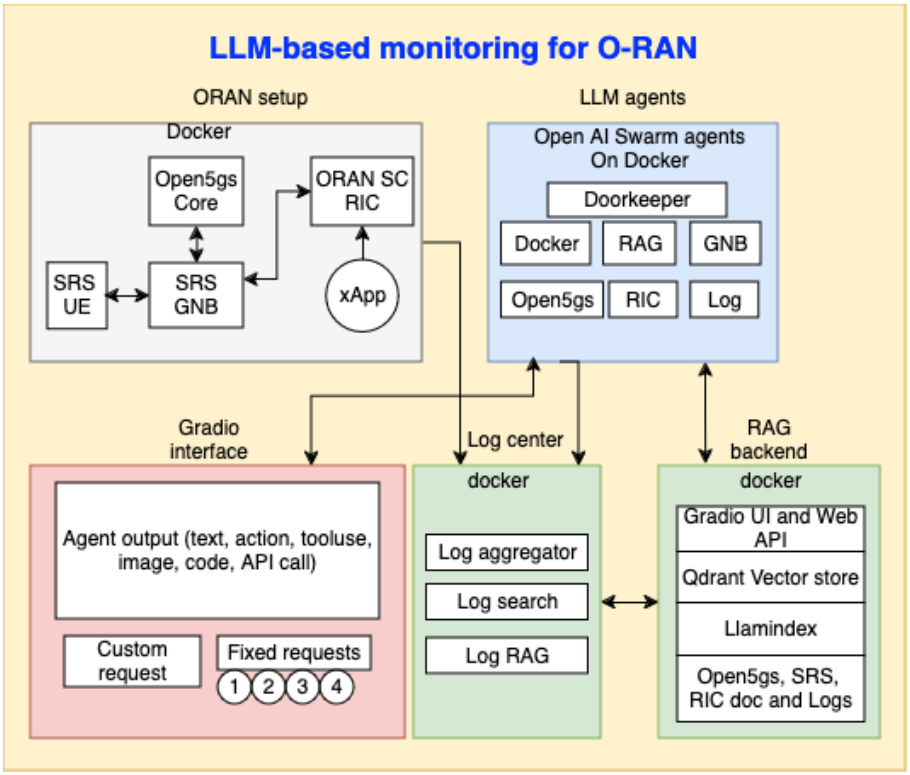

Figure 4-7: LLM-based monitoring for O-RAN

LLMs currently lack an understanding of their environment, so deploying them in an actual O-RAN scenario requires a grounded approach. This can be achieved through careful system

message design and prompt engineering, incorporating relevant O-RAN deployment details during LLM inference. Additionally, complex tasks cannot be accomplished in a single attempt, even by an intelligent human, necessitating the breakdown of tasks. An agent-based workflow has emerged as a powerful paradigm, decomposing problems into various components, providing tools for the LLM, and offering feedback from the environment. This approach effectively bridges the gap between the LLM's lack of grounding and the complex dynamics of systems like O-RAN. In the context of O-RAN systems, digital twins represent a promising technology that could support the development of LLM-based agent workflows. By providing a realistic representation of the network and exposing relevant contextual information and performance indicators, digital twins could offer valuable feedback during the reasoning process. Such capabilities may facilitate the evaluation and refinement of agent decisions in a controlled environment before they are deployed in the operational network, reducing risks while accelerating the development of intelligent O-RAN applications..

4.3.12.2 Support to Integrated Services

Our proposed monitoring and controlling system leverages the parallel processing capabilities of LLMs to extend its functionality across a wide variety of tasks within the O-RAN system. By exploiting the large context windows offered by recent LLMs, combined with tool integrations, Retrieval-Augmented Generation (RAG), and agentic workflows, the system achieves enhanced efficiency and adaptability. This system (implementation principles are depicted in Figure 4-8) connects with various O-RAN components (whether on bare metal, Docker, or Kubernetes), centralizes logs for analysis, and employs an agent-based workflow enhanced with RAG capabilities to diagnose issues, generate documentation, and coordinate all tasks automatically.

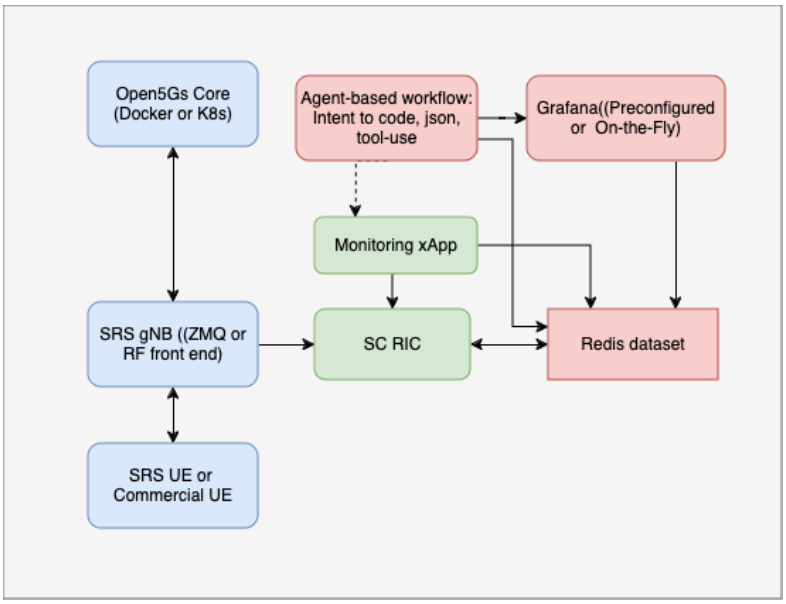

Figure 4-8: Implementation considerations of the LLM-based monitoring app

4.3.13 App-13: xApp for KPI monitoring

4.3.13.1 Overview and Use Case Scenario

The KPI monitoring xApp is developed by DLR to support intelligent resource management in an O-RAN enabled Non-Terrestrial Network (NTN) environment. In this type of network, the radio access condition can change dynamically due to satellite movement, varying channel quality, long propagation delay, Doppler effects, and limited onboard radio resources. These characteristics make continuous KPI monitoring essential before any optimization or scheduling decision can be applied. The xApp therefore focuses on collecting and analyzing relevant RAN performance indicators such as Signal-to-Noise Ratio (SNR), user throughput, latency, Physical Resource Block (PRB) availability, PRB utilization, PRB fragmentation, blocking probability, and QoS satisfaction.

Figure 4-9 illustrates the position of the KPI monitoring xApp within the O-RAN NTN-TN architecture and shows how RAN measurements are collected, processed, and exposed for monitoring and future optimization.

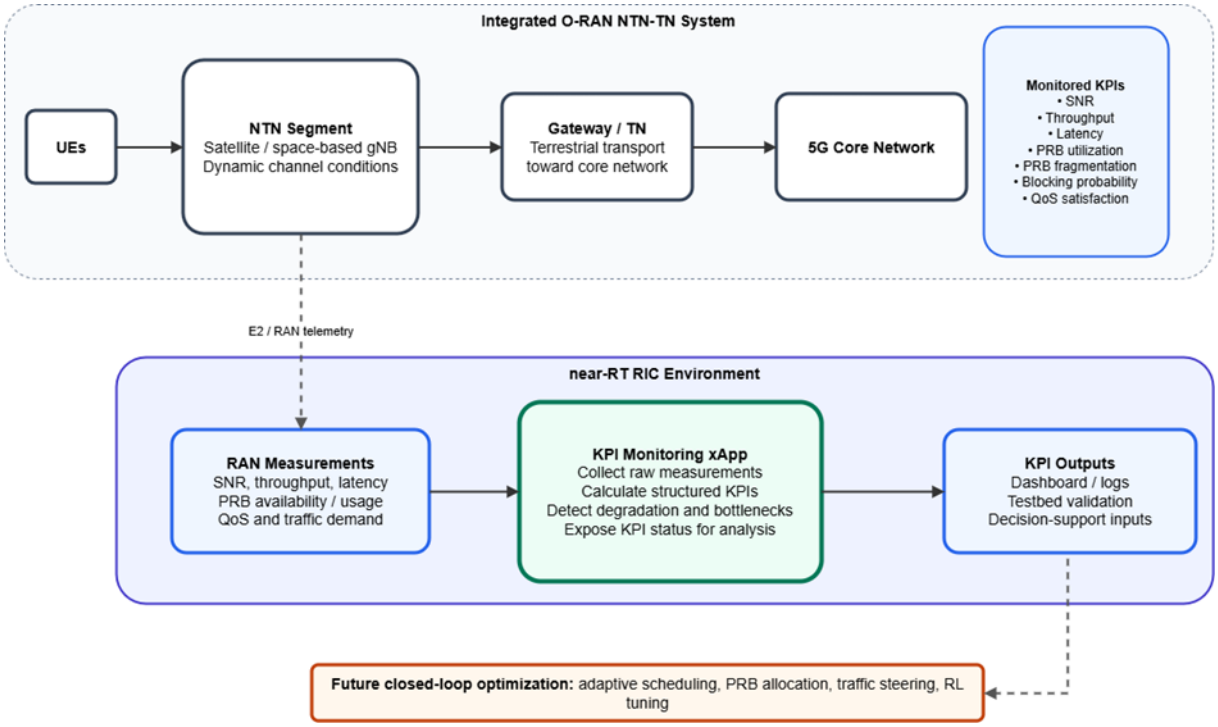

Figure 4-9: KPI monitoring xApp in an integrated O-RAN NTN-TN architecture

The development of the xApp follows a near-real-time monitoring approach within the O-RAN architecture. The xApp receives measurements from the RAN through the near-RT RIC environment and processes them into meaningful KPI information that represents the current network state. SNR is used to observe the instantaneous channel condition of each user, meanwhile throughput and latency are used to evaluate service performance. PRB related KPIs, such as PRB availability, utilization, and fragmentation are monitored to understand how efficiently radio resources are being allocated. QoS satisfaction are also considered to ensure that the monitoring function does not only focus on system level performance, but also captures the service quality experienced by individual users.

In the first stage, the KPI monitoring xApp will mainly operate as an observation and evaluation component. It will collect raw measurements, calculate KPI values, and present them in a structured format for analysis. This allows the network operator to identify performance degradation, inefficient resource usage, high-latency periods, and possible bottlenecks in the NTN RAN.

A practical use case is the monitoring of service performance for latency sensitive and bandwidth demanding applications over an O-RAN NTN-TN architecture. For example, when multiple users or traffic flows are served through the NTN access path, the KPI monitoring xApp can detect whether specific users experience reduced throughput, increased latency, inefficient PRB allocation, or unfair resource distribution. This information is useful for evaluating whether the NTN-TN system can maintain stable service quality under changing radio conditions.

The long-term goal of the KPI monitoring xApp is to provide the foundation for closed-loop optimization in O-RAN-enabled NTN systems. The monitored KPIs can later be used as input for adaptive scheduling, intelligent PRB allocation, traffic steering, or reinforcement-learning-based parameter tuning. In this way, the xApp is not limited to passive monitoring, but can also evolve into a decision-support function that enables more efficient radio resource management. The expected outcome is improved throughput, reduced latency, lower blocking probability, better PRB utilization, reduced resource fragmentation, and improved fairness among users in dynamic NTN scenarios.

4.3.13.2 Support to Integrated Services

The KPI monitoring xApp can be integrated into the O-RAN NTN-TN system by deploying it inside the near-real-time RIC as a modular xApp. In this position, the xApp can receive RAN measurements from the O-RAN nodes through the E2 interface and process them within the near-RT RIC control loop. The monitored data can include user-level radio measurements, throughput reports, latency-related indicators, PRB usage, and resource availability information. After collecting these measurements, the xApp processes them into structured KPIs that can be visualized, logged, and forwarded to other functions. This integration enables the near-RT RIC to act as the central intelligence point for monitoring the NTN-TN RAN behavior, while keeping the xApp flexible enough to support future closed-loop control functions such as adaptive scheduling, PRB allocation, or AI/ML-based resource management.

4.3.14 App-14: Traffic steering and policy-based load balancing

4.3.14.1 Definition

The primary goal of the traffic steering algorithm is to optimize the overall network performance and user experience. It could be achieved by load balancing that dynamically moves a part of traffic from one cell/sector/carrier/domain into another to use underused nodes (to meet traffic demand) and lighten load of overloaded nodes. Policy-based load balancing exposes a set of parameters that can affect load balancing processes in the form of a policy set. A policy can be defined per specific time, specific type of users, specific types of traffic etc., and can modify the load balancing process, e.g. by moving a low-traffic users (IoT) into the second/third closest cell, allowing the high priority users like emergency increase QoE.
In general, the process of load balancing can be realized if the algorithm has access to the

data used in handover procedure, i.e. power reports. Additionally, it has to monitor the average load of cells, their configuration (carrier frequency, handover thresholds). Moreover, algorithms need to be able to control handover of specific users – directly, or indirectly through influence on handover procedures parameters. All specified data and control can be used to optimize resource utilization in the network (improve QoE depending on policy and current network status).

4.3.14.2 Support to Integrated Services

This application relates to Multi-RAT Time-Sensitive Service. A traffic steering algorithm will be implemented and used in the integrated scenario where IEEE 802.11, transport, and RAN will be utilized. The application needs access to traffic channel condition metrics (like RSRP, 5QI), cell load estimation of the domain/node/cell, and capability of making handover (directly or indirectly). Such application improves overall spectrum efficiency by locating and associating UEs to cells based more on their needs (traffic type, device class, etc.) than only based on their actual position (received power). When changing traffic conditions or channel degradations prevent a specific radio access technology from meeting strict Quality of Service (QoS) and time critical boundaries for traffic flows, the optimization application computes real-time policy updates. These traffic steering and policy-based load balancing decisions initiate cross-RAT mobility and traffic redirection mechanisms. To preserve end-to-end bounded latency constraints, the application triggers target domain preparation to proactively reserve infrastructure resources and exchange user mobility contexts before traffic redirection. The Infrastructure Domain Management (IDM) entities across the RAN, non-3GPP, and transport domains are then leveraged to dynamically apply runtime routing changes and execution rules. Continuous synchronization across all domains is maintained via cross-layer status notifications, allowing the steering framework to seamlessly handle both proactive and reactive load balancing.

4.3.15 App-15: Radio Access Technology Monitoring

4.3.15.1 Definition

This app goal is to provide fine-grained monitoring telemetry data of the RATs that are included in the unified network architecture. While for different RATs the internal APIs to retrieve certain data are RAT dependent, the exposure of such data will utilize the E2 interface regardless of the specific RAT. The monitored data are categorized into three groups: (i) the statistics per traffic flows, (ii) channel quality related metrics and (iii) key-performance indicators (KPIs) related to latency and reliability for each traffic flow metrics. In the first group the app will monitor metrics such as: number of correctly Tx/Rx packets, number of retransmitted packets, number of correctly acknowledged packets, etc. All metrics related to statistics of traffic flow-basis as well as network node-bases will be collected in the first group. The second group will collect information like: channel state information (CSI), channel quality information (CQI), experienced interference using SINR, RSSI levels for packets received etc. In the third group metrics related to latency such as channel access delay, queue buffer status, queue fillings, TB latency, TB error rate, HARQ retransmissions etc. will be monitored. The third group of metrics will collect information like ingress and egress timestamps of the packets of certain traffic flow and report calculated end-to-end latency of the packets and jitter.

4.3.15.2 Support to Integrated Services

The generic RAT monitoring App is integrated into the multi-RAT time-sensitive service. This App allows for real-time monitoring of the traffic load and how good the application

requirements in terms of latency and jitter are maintained. In any control loop involved in multi-RAT management, monitoring is the initial step in the message flow before the controllers (like DMO and IDMO) will take decisions to change/update RAT configuration.

4.4 INTEGRATION AND TESTING OF COMPONENTS FOR SERVICES AND APPLICATIONS

This section introduces a conceptual framework for the integration and testing of the services and applications defined in Section 4.2 and Section 4.3. In alignment with the SDLC defined in Section 2, it outlines how service components are envisaged to be integrated into the overall SBA described in Section 3, and subsequently validated within the Digital Twin (DT) network. In addition, this section also proposes the definition of dedicated and automated testing procedures intended to verify the correctness, interoperability, and operational behaviour of the services and applications across heterogeneous 6G environments.

The concept aims to establish a structured and scalable integration and testing framework capable of supporting the use cases, reusable services, and distributed network domains described in this deliverable, while ensuring traceability between services, sub-modules, and validation activities.

Proposed Methodology

The proposed conceptual methodology envisions a top-down, use case-driven integration and testing approach aligned with the UNITY-6G SDLC, SBA and intent-based networking principles described in Sections 2 and 3. If adopted, it would enable systematic decomposition, integration, validation, and traceability of services and applications defined in this deliverable.

Under this concept, the approach would begin at the use case layer, where each use case described in Section 4.2 may involve multiple services and applications spanning Open RAN, transport, cloud-edge, NPN, satellite, orchestration, AI, and Digital Twin domains. Services and components contributed by partners would subsequently be decomposed into smaller sub-modules or functional features, representing the intended granularity for integration and testing activities.

The methodology is based on the following principles:

- One use case can contain multiple services and applications.
- One service or component can be decomposed into multiple sub-modules or features.
- Sub-modules may serve multiple use cases.
- Integration and testing activities are performed at sub-module level.
- Each test case is traceable to a sub-module, service/component, and use case.

Under this conceptual approach, it is envisaged that partners would decompose their services into sub-modules and identify the corresponding interfaces, dependencies, telemetry flows, and operational behaviors. This is intended to enable precise traceability and systematic validation across the complete service chain while supporting modularity, scalability, and reuse across different use cases.

Integration activities would focus on ensuring interoperability between services, orchestration components, monitoring systems, AI functions, Digital Twin entities, and heterogeneous network domains within the UNITY-6G SBA environment.

To support standardized validation across different services and applications, the methodology proposes common testing and evaluation categories that could include:

- Functional validation
- Interface and interoperability testing
- Performance and latency evaluation
- Scalability and adaptivity testing
- Energy-efficiency assessment
- Robustness and resilience validation

It is envisaged that evaluation and validation activities would be carried out using the UNITY-6G Digital Twin infrastructure to support automated testing, controlled experimentation, continuous integration workflows, and iterative refinement prior to deployment in operational 6G environments.

Overall, the proposed conceptual methodology is intended to provide a structured mechanism for coordinating partner contributions, managing reusable service components, and achieving comprehensive integration and validation coverage across the use cases and network domains described in this deliverable. The specific implementation of this framework will be refined and confirmed as the project progresses. The conceptual framework described in this section is intended to serve as a roadmap for the integration and testing activities that will be further developed and demonstrated in the upcoming project period.

5 CONCLUSIONS

This deliverable is an introduction to the work of WP5 about Services and Applications and how these exploit the capabilities of the UNITY-6G architecture. As 6G networks evolve, they become more heterogeneous with multiple domains that need to be integrated as a unified system. In this context, this document presents a new approach of service management by introducing SDLC. It covers the definition of services lifecycle, from design and implementation to deployment, operation, and optimization, while also establishes the foundations for SDLC management within the UNITY-6G framework. This framework builds upon SBA to support seamless interoperability among heterogeneous domains, including Open RAN, transport networks, non-terrestrial networks, AI-driven subsystems, digital twins, and cloud-edge infrastructures. For this reason, the principles of SBA are also discussed and some new modules, such as IDMO, cross-domain service-based management bus, and exposure mechanisms, are introduced as the architectural basis for flexible, programmable, and automated end-to-end service delivery. Based on the above, a wide portfolio of integrated services and intelligent applications is presented related to different scenarios about emergency communications, semantic services, time-sensitive networking, and AI-assisted management solutions. The analysis is focused on how domain-specific and cross-domain capabilities can be orchestrated to realize advanced business-oriented services aligned with future 6G requirements.

Future work within WP5 will focus on detailed definition of the functional components, communication flows, interfaces, and protocols of the proposed SBA, aiming to support end-to-end specifications and service orchestration across the UNITY-6G ecosystem. Following, the presented services will be integrated and tested in the projects' use cases and proof-of-concepts, contributing to the realization of a sustainable, intelligent, and automated 6G system.

REFERENCES

[1] Ericsson Blog, Advanced core network autonomy with cApps and Ericsson Intelligent Automation Platform, 2026, https://www.ericsson.com/en/blog/2026/6/intelligent-core-automation-with-eiap-and-capps

[2] UNITY-6G: Deliverable 2.1 Techno-economic analysis of the 6G environment, use case requirements and KPIs

[3] UNITY-6G: Deliverable 2.3 1st Release of UNITY-6G Architecture, March 2026.

[4] UNITY-6G: Deliverable 3.1 Intermediate Report on Network Implementation and Exposure including Open RAN, xHaul, Satellite, RIC components and Digital Twin Application Specifications and Design, May 2026.

[5] UNITY-6G: Deliverable 4.1 Intermediate report on distributed intelligence solutions for resource provisioning, edge-cloud continuum and conflict resolution for integrated components, June 2026.

[6] ETSI GR NFV-MAN 001 V1.2.1 (2021-12); Network Functions Virtualisation (NFV); Management and Orchestration; Report on Management and Orchestration Framework.

[7] ETSI TS 128 530 V15.2.0 (2019-10); 5G; Management and orchestration; Concepts, use cases and requirements (3GPP TS 28.530 version 15.2.0 Release 15).

[8] TM Forum, Open Digital Architecture, https://www.tmforum.org/open-digital-architecture/

[9] Villa, Davide, et al. "Colosseum as a digital twin: Bridging real-world experimentation and wireless network emulation." *IEEE Transactions on Mobile Computing* 23.10 (2024): 9150-9166.

[10] 3GPP, “3GPP TS 23.501 (2017.12)-System Architecture for the 5G System, Release 15, Technical Specification TS 23.501, Third Generation Partnership Project (3GPP), 2017.

[11] 3GPP, “3GPP TS 23.502 (2021.01)-Procedures for the 5G System (5GS), Release 16, Technical Specification TS 23.502, Third Generation Partnership Project (3GPP), 2021.

[12] 3GPP, “3GPP TS 29.500 (2022.07)-5G; 5G System; Technical Realization of Service Based Architecture, Release 17, Technical Specification TS 29.500, Third Generation Partnership Project (3GPP), 2022.

[13] 3GPP, “3GPP TS 28.533 (2025.10)-5G; Management and Orchestration; Architecture Framework, Release 19, Technical Specification TS 28.533, Third Generation Partnership Project (3GPP), 2025.

[14] O-RAN Alliance, "O-RAN Architecture Description v12.01," Technical Specification O-RAN.WG1.TS.OAD-R005-v16.00, 2026

[15] M. A. Uusitalo, M. Ericson, B. Richerzhagen, E. U. Soykan, P. Rugeland, G. Fettweis, D. Sabella, G. Wikström, M. Boldi, M.-H. Hamon, “Hexa-X the European 6G flagship project, in 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit). IEEE, 2021, pp. 580–585.

[16] EU SNS JU 6G-BRICKS Project, Building Reusable testbed infrastructures, https://6g-bricks.eu/ [Accessed: 2026-04-23]

[17] EU SNS JU 6G-CLOUD Project, Service-oriented 6G Network Architecture, https://www.6g-cloud.eu/ [Accessed: 2026-04-23]

[18] T. Chen, S. Kuklinski, E. Pateromichelakis, K. Samdanis, A. Kourtis, N. Nikaein, A.

Skarmeta, “Service-oriented Architecture Evolution towards 6G Networks”, in IEEE Conference on Standards for Communications and Networking (CSCN), Munich, Germany, Nov. 2023.

[19] E. Zeydan, J. Mangues-Bafalluy, J. Baranda, M. Requena, Y. Turk, “Service based Virtual RAN architecture for next generation cellular systems”, in IEEE Access, pp. 9455-9470, 2022.

[20] T. Chen, 6G-Cloud Service-Oriented Architecture, in 6G Series Workshop by Hexa-X-II, Feb 2025. Available at: https://hexa-x-ii.eu/wp-content/uploads/2025/02/6G-Cloud-Service-oriented-architecture_HX2-workshop_Tao_12.2.2025_v1.1.pdf [Accessed: 2026-04-23]

[21] EU SNS JU ORIGAMI Project, Optimized Resource Integration and Global Architecture for Mobile Infrastructure for 6G, https://sns-origami.eu/ [Accessed: 2026-04-23]

[22] EU SNS ORIGAMI Project, D2.2: Initial architecture of ORIGAMI, new business models and refinement of requirements and KPIs, April 2025.

[23] L. E. Chatzieleftheriou *et al*., "6G Standardization Potential of the Origami Novel Architectures and Use Cases," *2025 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Poznan, Poland, 2025, pp. 387-392.

[24] EU SNS JU 6Green Project, Green Technologies For 5/6G Service-Based Architectures, https://www.6green.eu/ [Accessed: 2026-04-23]

[25] EU SNS JU 6Green Project, D3.3: The 6Green 5/6G Service Based Architecture, April 2026.

[26] EU SNS-JU Hexa-II Project, European level 6G Flagship project, https://hexa-x-ii.eu [Accessed: 2026-04-23].

[27] EU SNS-JU Hexa-II Project, D3.5: Final architectural framework and analysis, February 2025.

[28] HE EUSPA, 5G-GOVSATCOM Project, 2024, [Online]. Available: https://5g-govsatcom.eu/

[29] HE EUSPA, 5G-HUB Project, 2024, [Online]. Available: https://www.5g-hub.eu/

[30] K. Ito and N. Izuka, "Proposal of Client-Server Based Vertical Handover Scheme Using Virtual Routers for Edge Computing in Local 5G Networks and WLANs," 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2023.

[31] EU SNS-JU UNITY-6G Project, D3.1: Intermediate report on network implementation and exposure including Open Ran, xhaul, Satellite, RIC components and digital twin application specifications and design, May 2026.

[32] R. Greidi, A. Ben Ari, T. Maimon, G. Kedar, and K. Cohen, “DPIR: Deep Predictive Interference-Aware Routing for Wireless Networks,” *2025 61st Allerton Conference on Communication, Control, and Computing Proceedings*, Allerton Conference on Communication, Control, and Computing, Sept. 17, 2025. Handle: https://hdl.handle.net/2142/130262

[33] 3GPP, ”3GPP TS 28.312 (2026.03) V19.5.0 Management and orchestration; Intent driven management services for mobile networks (Release 19). Technical Specification TS 28.312, Third Generation Partnership (3GPP), 2026

[34] TM Forum, “Intent in Autonomous Networks” (IG1253; v1.3.0), TM Forum, 08/2022

[35] TM Forum Specification; “Service Order Management API User Guide (TMF641)” v4.1.1, TM Forum 12/2020

[36] TM Forum Specification; “Resource Order Management API User Guide (TMF652)” v4.0.0, TM Forum 07/2020

[37] Haxhibeqiri, J., Campos, P. A., Havinga, T., Moerman, I., & Hoebeke, J. (2025, December). Distributed Multi-link Operation (MLO) for Frame Replication in Wireless Time-Sensitive Networking. In *2025 8th International Conference on Advanced Communication Technologies and Networking (CommNet)* (pp. 1-7). IEEE

[38] Fontanesi, G., Wilhelmi, F., & Giordano, L. G. (2024, September). Continuous multi-link operation: A contention-free mechanism for the unlicensed spectrum. In *2024 IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)* (pp. 766-770). IEEE.

[39] E. J. Birrane, N. Kuhn, Y. Qu, R. Taylor, and L. Zhang, ‘Time-Variant Routing (TVR) Use Cases’, no. 9657. RFC Editor, Oct-2024.

APPENDIX A

This Appendix summarizes the project KPIs that are associated with WP5. The measurement methodology of these KPIs defined in [2] is not repeated here; however, a brief explanation of how each KPI is linked with WP5 is provided in Table 2.

Table 2: KPIs linked with WP5.

#	KPI	WP5 association
#1 #32	Deliver four distributed management entities, namely monitoring (MS), analytics (AE), decision engines (DE) and actuators (ACT) deployable in each heterogeneous domain. Deliver two proof-of-concepts of the architecture: (i) TN – NTN and (ii) a time-sensitive NPN integrating both 3GPP and IEEE 802.11 technology.	These functional KPIs aim to deliver the deployed tuples and demonstrate them in two PoCs. WP5 defines the service lifecycle management and the IDMO capabilities for service orchestration, components that are critical enablers for the UNITY-6G architecture.
#6 #30	Decrease service creation and termination time (for PoC #2, ES #3). Decrease service creation and termination time (PoC#1, ES#3).	These KPIs are linked with the service orchestration and the SLCM using DT validation, i.e., how to initiate or terminate service creation requests, as well as pre-validate service configurations, predict resource requirements, and simulate deployment before physical instantiation using the DT.
#18	Reduction in the deployment time of heterogeneous domain services.	This KPI is directly linked with the IDMO capabilities, i.e. AI-native service-based orchestrator with cross-domain resource abstraction from the UNITY-6G heterogeneous domains.
#19	Reduce time to manage network resources and management overhead and the reaction time by taking proactive actions in the network (time from identification to resolution via appropriate reconfigurations) with integrated design.	This KPI is directly linked with the IDMO management capabilities that enable integrated design (AI-assisted proactive control, DT validation)
#22	Allocate 20% more network service LCM tasks	The network service LCM tasks include instantiation, scaling, termination, healing, updating, and monitoring of services and network slices, therefore directly associated with the lifecycle management of services defined in WP5.

#	KPI	WP5 association
#24	Decrease end-to-end message delivery latency and increase maximum message publish throughput in SBA service bus in integrated network scenarios.	The message delivery latency directly depends on the SBMA and the SBI, showcasing how the integrated design improves e2e responsiveness and publish capacity.
#26	Increase the ratio of service management and LCM tasks resolved by local AE/DE components.	This KPI measures the amount of service-management and LCM tasks that are autonomously completed at the local AE/DE level, linked with the operation of DMOs.
#31	Reduce service recovery time below 180s.	KPI #31 is linked with the service lifecycle management provided by the IDMO, since it measures the duration from the moment a service failure is detected until the moment the service is fully restored
#36	Guarantee 99.99999% service availability and reliability based on effective monitoring and LCM	This KPI is linked with the availability of the XR integrated service, including its monitoring over an extended time period with varying conditions (load, disturbances, and failures), as well as the measurement of failure and recovery metrics
#37	Guarantee 99.99999% service continuity through AI-driven proactive management of cloud-native services.	AI-driven (anomaly detection, DT-enabled forecasting) proactive management by the IDMO to enable automated mitigation (e.g., pre-emptive scaling, service migration, reconfiguration).
#38	Reduce Operation expenditure (OPEX) by 30% due to the automation of service management and increase in link utilization (via planning annual frequency licensing).	AI-driven orchestration for the service lifecycle management, i.e., service creation/termination, fault management and healing, performance optimization, SLA monitoring and adjustment, cross-domain coordination.