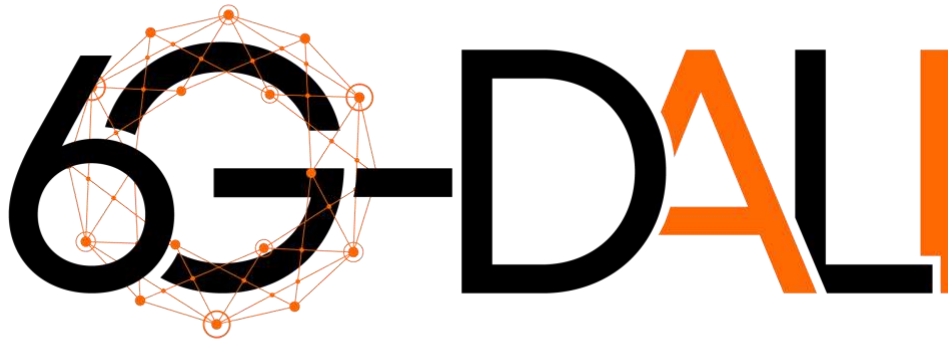


3rd March, 2026



Initial Framework Architecture

Deliverable 2.3, Work Package 2

Project Identifier	HORIZON-JU-SNS-2024-STREAM-B-01-08, Project 101192750		
Project Name	6G Data and ML operations automation via an end-to-end AI framework		
Acronym	6G-DALI		
Start Date	01 January 2025	End Date	31 December 2027
Project URL	www.6gdali.eu		
Deliverable	D2.3 Initial Framework Architecture		
Work Package	WP2 '6G-DALI Framework Architecture and Business Model'		
Contractual due date	28/02/2026	Actual submission date	03/03/2026
Type	R	Dissemination Level	Public
Lead Beneficiary	Intracom Telecom		
Responsible Author	Vasiliki Parousidou (ICOM)		
Contributors	Marco Ruta (NXW), Giacomo Bernini (NXW), Luisa Schuhmacher (KUL), Panagiotis Marantis (ISI/ATH), Yuandou Wang (TUD), Michalis Rantopoulos (OTE), Ioannis Chochliouros (OTE), Srikumar Venugopal (IBM), Dimitrios Amaxilatis (SPW), Themistoklis Sarantakos (SPW), Menuka Perera (EUR), Natalia Grigoriadou (ICOM)		
Peer reviewer(s)	Christian Pinto (IBM), Mahdi Sharara (VIA)		
Version	Date	Authors	Comments
1.0	03/03/2026	Vasiliki Parousidou (ICOM)	Submitted Version

Disclaimer

The content of this document reflects only the author's view. Neither the European Commission nor the Smart Networks and Services Joint Undertaking (SNS JU) are responsible for any use that may be made of the information it contains.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the 6G-DALI consortium make no warranty of any kind regarding this material including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neither the 6G-DALI Consortium nor any of its members, their officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the 6G-DALI Consortium nor any of its members, their officers, employees or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

©6G-DALI Consortium. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both. Reproduction is authorised provided the source is acknowledged.

Table of Contents

Table of Contents.....	3
Glossary of terms and abbreviations used	7
Executive Summary.....	10
1 Introduction	12
1.1 Mapping 6G-DALI outputs.....	12
1.2 Deliverable overview and report structure.....	13
2 6G-DALI Project Overview.....	14
2.1 6G-DALI Vision	14
2.2 Mapping with Relevant Research Initiatives	14
2.2.1 SUNRISE-6G.....	15
2.2.2 6G-BRICKS	15
2.2.3 6G-SANDBOX	16
2.2.4 ADROIT6G	17
2.2.5 6G4SOCIETY	18
2.3 Compliance with Relevant Standards	19
2.3.1 3GPP	19
2.3.2 Gaia-X.....	20
2.3.3 ETSI	22
2.3.4 O-RAN	23
3 6G-DALI High Level Architecture	25
3.1 User Requirements.....	25
3.2 High Level Logical Architecture.....	26
4 Detailed Architecture of 6G-DALI.....	29
4.1 DataOps	29
4.1.1 Functional Architecture	29
4.1.2 DataOps User Interface	30
4.1.3 Intent-based Data Search and Generation.....	32
4.1.4 DataOps Orchestrator	33
4.1.5 Experimentation On Demand	35
4.1.6 DataOps Tools Stack.....	35
4.1.7 Interfaces.....	36
4.2 MLOps	36
4.2.1 Functional Architecture	36
4.2.2 MLOps User Interface	37
4.2.3 MLOps Orchestrator	39
4.2.4 Experimentation on Demand	41
4.2.5 MLOps Tools Stack	41
4.2.6 Interfaces.....	44
4.3 6G Data Space	44
4.3.1 Functional Architecture	45
4.3.2 Data Space User Interface	46
4.3.3 Data Quality & Governance	48
4.3.4 Federated Catalogue of Data Services & Models.....	50
4.3.5 Metadata Storage & Indexing.....	53
4.3.6 Discovery Services.....	54
4.3.7 Interfaces.....	55
4.4 Adaptation Layer.....	56
4.4.1 Functional Architecture	56
4.4.2 Common Set of APIs.....	57
4.4.3 Data Space Connector	58

4.4.4	Testbed Adapter	60
4.5	Testbeds	60
4.5.1	Functional Architecture	60
4.5.2	Local MLOps Tools Stack	61
4.5.3	Short-Term Raw Data Storage	61
4.5.4	6G Enabler	61
4.6	Digital Twin Testbed	61
4.6.1	Functional Architecture	61
4.6.2	DT Management and Orchestration	62
4.6.3	DT components	62
4.6.4	DT Testbed Adapter	62
4.6.5	Interfaces	62
4.7	6G Data Lake	62
4.7.1	Functional Architecture	63
4.7.2	Short-term Raw Data Storage	64
4.7.3	Long-term Final Data Storage	64
4.7.4	Data Space Connector	64
5	Architectural Process Perspective	66
5.1	End-to-end DataOps Workflows	66
5.1.1	Cold Data Search and Retrieval	66
5.1.2	Hot Data Generation via Chatbot	67
5.1.3	Hot Data Generation via Experimentation on Demand	68
5.1.4	Data Transformation and Augmentation Services	68
5.2	End-to-end MLOps Workflow	69
5.3	End-to-end Digital Twin Workflow	70
5.3.1	Data Generation Workflow	70
5.3.2	RLOps Validation Workflow	71
6	Conclusions	72
7	References	73
Annex I:	6G-DALI Testbeds	75
I.1	KUL Testbed	75
I.1.1	Functional Architecture	75
I.1.2	Local MLOps Tools Stack	76
I.1.3	Short-Term Raw Data Storage	76
I.1.4	Interfaces	76
I.2	ISI/ATH Testbed	76
I.2.1	Functional Architecture	76
I.2.2	Local MLOps Tools Stack	77
I.2.3	Short-Term Raw Data Storage	78
I.2.4	Interfaces	78
I.3	EUR Testbed	78
I.3.1	Functional Architecture	78
I.3.2	Local MLOps Tools Stack	79
I.3.3	Short-Term Raw Data Storage	80
I.3.4	Interfaces	80

List of Figures

Figure 1: SUNRISE-6G Overall Reference Architecture [1]	15
Figure 2: 6G-BRICKS Overall Architecture [2]	16
Figure 3: 6G-SANDBOX Facility [3]	17
Figure 4: ADROIT6G Architecture [4]	18
Figure 5: AIMLE Enablement Layer	19
Figure 6: AI/ML Management Model Lifecycle	20
Figure 7: 6G-DALI Overall Architecture	26
Figure 8: 6G-DALI DataOps architecture	29
Figure 9: MLOps Functional Blocks	37
Figure 10: MLOps Northbound Interface	38
Figure 11: MLOps Management and Orchestration Layer	40
Figure 12: High-level overview of the Trustworthy AI tools across the MLOps and RLOps frameworks	43
Figure 13: 6G-DALI Data Space architecture	45
Figure 14: Unified Data Governance Architecture	50
Figure 15: Adaptation layer architecture	56
Figure 16: Testbed functional architecture	60
Figure 17: The Digital Twin Architecture	61
Figure 18: 6G Data Lake Storage High Level Architecture	63
Figure 19: DataOps workflow for cold data search and retrieval	66
Figure 20: DataOps workflow for hot data generation via chatbot	67
Figure 21: DataOps workflow for hot data generation via experimentation on demand	68
Figure 22: DataOps workflow for data transformation services	69
Figure 23: MLOps E2E Workflow sequence diagram	70
Figure 24: Data generation workflow on DTT	70
Figure 25: RLOps Validation Workflow on DTT	71
Figure 26: KU Leuven 6G-Bricks Testbed Schematic Representation	75
Figure 27: ATH 5G/6G testbed facility overview	77
Figure 28: EUR Testbed architecture	78

List of Tables

Table 1: Adherence to 6G-DALI GA Deliverable & Tasks Descriptions.....	12
Table 2: User Requirements regarding 6G-DALI overall architecture.....	25
Table 3: 6G-DALI architecture global interfaces.....	28
Table 4: MLOps API Categories	38
Table 5: 6G-DALI MLOps Service Management Functionalities	40
Table 6: MLOps Interfaces.....	44
Table 7: Adaptation Framework APIs developed in the SUNRISE-6G project	57
Table 8: KU Leuven 6G-Bricks Testbed Interfaces, as depicted in Figure 7.	76
Table 9: KPIs provided by EUR testbed.....	79

Glossary of terms and abbreviations used

Abbreviation / Term	Description
3GPP	3rd Generation Partnership Project
AI	Artificial Intelligence
AIMLE	AI/ML Enablement Layer
API	Application Programming Interface
AppD	Application Descriptor
BDIx	Belief-Desire-Intention extended
BWPs	Bandwidth Parts
CEC	Cloud Edge Continuum
CN	Core Network
CNF	Container Network Function
CoTL	Cooperative Transfer Learning
COR	Communication Overhead Reduction
CU	Central Unit
CRUD	Create, Read, Update and Delete
DMO	Domain Management and Orchestration
DTT	Digital Twin Testbed
DU	Distributed Unit
ELT	Extract Load Transform
EPPE	Enhancement in Privacy-Preserving Efficiency
eMBB	enhanced Mobile Broadband
FL	Federated Learning
FPGA	Field-Programmable Gate Array
GHz	Gigahertz
GPS	Global Positioning System
HPO	Hyperparameter Optimization
HTTP	Hypertext Transfer Protocol
IDLM	Intent Driven Lifecycle Management
IDSA	International Data Spaces Association
I/Q	In-phase/Quadrature
IR	Infrared
JSON	JavaScript Object Notation

K8s	Kubernetes
LLM	Large Language Model
MANO	Management and Orchestration
MARL	Multi-Agent Reinforcement Learning
MCP	Model Context Protocol
MEC	Multi-access Edge Computing
MHz	Megahertz
MIMO	Multiple-Input and Multiple-Output
ML	Machine Learning
MLKTL	ML-Knowledge-based Transfer Learning
MLOps	Machine Learning Operations
mMTC	massive Machine-Type Communications
mmWave	Millimeter Wave
Mpixel	Megapixel
Msp/s	Megasamples per second
NSD	Network Service Descriptors
NEF	Network Exposure Function
NFV	Network Functions Virtualization
NFVO	Network Functions Virtualization Orchestrator
NR	New Radio
NTN	Non-Terrestrial Networks
NWDAF	Network Data Analytics Function
OAI	Open Air Interface
OpenOP	Open Operator Platform
O-RAN	Open Radio Access Network
RAN	Radio Access Network
REST API	Representational State Transfer Application Programming Interface
RF	Radio Frequency
RIC	Real-Time Intelligent Controller
RLOps	Reinforcement Learning Operations
RU	Radio Unit
SDG	Software Development Group
SM	Service Model
SoTA	State-of-the-art

SSH	Secure SHell
TaaS	Testbed-as-a-Service
TDD	Time Division Duplex
TF	Transfer Learning
UE	User Equipment
USRP	Universal Software Radio Peripheral
URLLC	Ultra-Reliable Low-Latency Communications
VAL	Vertical Application Layer
VNF	Virtual Network Functions
YAML	Yet Another Markup Language
ZSM	Zero touch network & Service Management

Executive Summary

This deliverable contributes to the detailed definition of the **6G-DALI architecture**, providing a structured and standards-aligned blueprint for enabling end-to-end AI experimentation in 6G systems. The 6G-DALI project addresses the critical need for an integrated AI framework capable of supporting the evolution toward AI-native 6G networks, where data-driven intelligence is deeply embedded across network functions, services, and vertical applications. As 6G systems become AI-driven, there is a growing need for coordinated DataOps and MLOps mechanisms that ensure scalable experimentation, reproducibility, and trustworthy data exchange.

At its core, 6G-DALI introduces a novel architectural approach built upon two synergistic pillars: (i) AI experimentation as a service via **MLOps**, and (ii) structured data collection, transformation, and provisioning via **DataOps**. These pillars are interconnected through a governed 6G Data Space that enables secure, sovereign, and interoperable sharing of datasets, AI models, and data-driven services across federated testbeds. By aligning with International Data Spaces (IDS) and Gaia-X principles, the 6G-DALI architecture fosters trusted data sharing and ensures compliance with European data governance frameworks.

This deliverable builds upon the project vision, objectives, and user requirements, as well as an analysis of the state of the art in relevant research initiatives and standardization activities. It systematically translates high-level project goals into a coherent architectural framework. The document first defines the high-level logical architecture, identifying the primary architectural layers and blocks, and then provides detailed logical architectures for each component. In doing so, it bridges conceptual design with practical implementation considerations.

The 6G-DALI architecture is structured into four main layers: the **Interface Layer**, the **Orchestration Layer**, the **Adaptation Layer**, and the **Infrastructure Layer**. Within these layers, the main architectural blocks, i.e., **DataOps**, **MLOps**, **6G Data Space**, **Adaptation Layer**, **Testbeds**, **Digital Twin Testbed**, and **6G Data Lake**, are defined along with their roles and logical interactions.

- The **Interface Layer** acts as the primary entry point for experimenters. It provides user-facing capabilities for dataset discovery, triggering dataset generation experiments, AI model management, and access to the federated catalogue. This layer abstracts infrastructure complexity and enables seamless interaction with DataOps and MLOps services.
- The **Orchestration Layer** functions as the coordination backbone of the framework. It manages end-to-end workflows, interprets experiment requests, allocates resources for data generation experiments, and ensures coherent interactions among DataOps, MLOps, and 6G Data Space components. Through this layer, the framework enables end-to-end automation of data processing pipelines and AI lifecycle operations while maintaining governance and traceability.
- The **Adaptation Layer** provides standardized interfaces that connect the core framework to heterogeneous 6G testbeds. It ensures interoperability and seamless integration of data generation and model deployment processes across diverse infrastructures.
- The **Infrastructure Layer** encompasses the distributed experimentation environments and storage systems that support the framework. It includes physical 6G testbeds, a Digital Twin Testbed for virtualized large-scale experimentation, and the 6G Data Lake for persistent data storage. Importantly, while testbeds may include advanced computational resources, such resources remain under local governance and are not exposed as general-purpose GPU-as-a-Service infrastructure. The 6G-DALI framework focuses on orchestration, lifecycle management, and controlled experimentation across 6G testbeds.

For each architectural block, the deliverable provides a detailed logical architecture specifying internal sub-components, responsibilities, and interfaces. Emphasis is placed on interoperability, extensibility, and standards compliance. The architecture is designed to remain technology-agnostic while aligning with relevant 3GPP standards (e.g., AI/ML management functions and NWDAF), ETSI initiatives, and

data space governance frameworks. This alignment ensures that 6G-DALI remains interoperable with emerging 6G and AI standardization efforts.

In addition to the structural architectural definition, the deliverable introduces architectural process perspectives that illustrate how the defined components interact in realistic experimentation scenarios. These process views demonstrate **end-to-end workflows**, including:

- Cold data discovery through federated catalogue queries within the 6G Data Space.
- Hot data generation through orchestrated experiments on 6G testbeds.
- Data transformation and augmentation pipelines managed through DataOps processes.
- AI/ML lifecycle workflow execution through the MLOps components.
- Synthetic data generation through the Digital Twin Testbed (DTT).
- Reinforcement learning algorithms validation prior to real-world deployment through DTT.

These workflows illustrate how the different architectural layers collaborate to support reproducible experimentation, cross-domain coordination, and secure data exchange. They also demonstrate how DataOps and MLOps processes are automated and orchestrated in a coherent and governed manner.

The main conclusion of this work is that an integrated, modular, and standards-aligned architectural framework is essential to enable trustworthy and interoperable AI experimentation in 6G environments. The proposed 6G-DALI architecture provides a solid foundation for the subsequent implementation, integration, and validation phases of the project, while remaining flexible to incorporate feedback and evolving requirements.

Overall, the purpose of this deliverable is to bridge the gap between high-level project objectives and concrete system realization by defining a clear and comprehensive architectural framework. This framework will guide the development of the 6G-DALI system and support its contribution to the evolution of AI-native 6G systems.

1 Introduction

The goal of this section is to provide a brief outline of the **objectives** of the specific 6G-DALI Deliverable, how those are aligned and relevant with the overall project, and what was the approach followed in order to achieve them.

1.1 Mapping 6G-DALI outputs

The purpose of this section is to map 6G-DALI Grant Agreement commitments, both within the formal Deliverable and Task description, against the project’s respective outputs and work performed. This is described in Table 1.

Table 1: Adherence to 6G-DALI GA Deliverable & Tasks Descriptions

Deliverable description			
D2.3 ‘Initial Framework Architecture’: ‘The initial version of the 6G-DALI framework architecture’			
6G-DALI Task No. and Title	6G-DALI GA Task Description	Respective document chapter(s)	Justification
Task 2.3 Framework architecture	The task provides a detailed specification of the 6G-DALI AI framework aligned with the requirements defined in T2.2. The framework is composed of a set of interoperable components that collaboratively enable the discovery, generation, preparation, and management of 6G datasets by leveraging multiple 6G testbeds and a Digital Twin, while operating within the 6G data space. Core components include the Northbound API, MLOps/ROps, data generation and preparation mechanisms, the adaptation layer, and testbed-specific adapters. For each component, the task specifies the functional building blocks, required inputs, and expected outputs. Particular emphasis is placed on the interaction between the adaptation layer and the adapters, as these interfaces	Section 2 – 6G-DALI Project Overview	Section 2 provides the contextual foundation for the framework architecture by presenting the 6G-DALI vision, and mapping to the SoTA and relevant standards. This ensures that the AI framework design is aligned with existing research initiatives and standardization efforts.
		Section 3 – 6G-DALI High Level Architecture	Section 3 translates the project vision and requirements into a high-level architectural design by listing user requirements and the main architectural blocks and their interactions. It establishes the structural blueprint of the 6G-DALI framework.
		Section 4 – Detailed Logical Architecture of 6G-DALI	Section 4 delivers the detailed specification of the AI framework by describing internal components, functional blocks, interfaces, and interactions of the MLOps, DataOps, 6G Data Space, and Infrastructure architectural blocks. It also details the adaptation layer and adapter design to support interoperability and future extensibility.

	enable communication with external systems and heterogeneous testbeds. The design of adapters is carefully specified to facilitate the seamless integration of new testbeds into the 6G-DALI framework and to ensure interoperability with the 6G Data Space.	Section 5 – Architectural Process Perspective	Section 5 illustrates the practical realization of the AI framework through end-to-end architectural workflows covering DataOps, MLOps, and Digital Twin processes. These workflows demonstrate how the defined architectural components interact to support the 6G-DALI objectives in operational scenarios.
--	---	---	--

1.2 Deliverable overview and report structure

This deliverable is structured to provide the architectural specification of the 6G-DALI framework, detailing its design principles, architectural components, and operational perspectives.

- **Section 1** introduces the deliverable, establishing the context within the broader 6G-DALI project outputs and outlining the document structure.
- **Section 2** presents an overview of the 6G-DALI project, including its vision, a mapping to the state of the art of relevant research initiatives, and an overview of applicable standards.
- **Section 3** introduces the high-level architecture of 6G-DALI, defining the user requirements and presenting the high-level logical architecture and main architectural blocks.
- **Section 4** provides the detailed logical architecture of the 6G-DALI framework, organized around the main architectural blocks. Section 4.1 describes the DataOps architecture, analysing its internal components, functionalities, and interfaces. Section 4.2 presents the MLOps architecture, while section 4.3 describes the 6G Data Space architecture, covering the components responsible for data generation, transformation, and provisioning. Finally, Sections 4.4-4.7 present the Adaptation Layer, that provides the crucial interface between the framework and the testbeds, the 6G-DALI testbeds and the Digital Twin Testbed (DTT), as well as the 6G Data Lake, which supports data storage.
- **Section 5** presents the architectural process perspectives, illustrating end-to-end workflows that demonstrate how the defined architectural components interact to support data generation, AI/ML lifecycle management, and data space operations. In particular, section 5.1 describes four DataOps workflows for cold data search, hot data generation, and data transformation and augmentation services, section 5.2 illustrates an end-to-end MLOps workflow, while section 5.3 presents two Digital Twin-based workflows.
- **Section 6** concludes the deliverable by summarizing the main architectural outcomes and outlining key insights and future directions.

2 6G-DALI Project Overview

This section provides an overview of the 6G-DALI project, outlining its vision and positioning within the broader research and standardization landscape. The 6G-DALI project represents a cutting-edge European initiative that aims to design and implement a comprehensive end-to-end framework for DataOps and MLOps in 6G networks, bridging the domains of artificial intelligence, data management, and telecommunications. By enabling automated, intelligent, and trustworthy data-driven workflows, 6G-DALI supports the development and validation of AI-enabled 6G systems.

Through its innovative architecture, 6G-DALI enables large-scale experimentation, advanced data analytics, and the integration of digital twin technologies to accelerate the evolution of AI-driven 6G infrastructures. The project aligns with major European research initiatives, ensuring consistency with the principles of Gaia-X, 3GPP and O-RAN. This alignment guaranteed data sovereignty, interoperability, and secure data exchange, while contributing to open innovation ecosystems and global standardization efforts.

2.1 6G-DALI Vision

The vision of 6G-DALI is to build a novel end-to-end AI framework that seamlessly connects 6G data with vertical industries, ML developers and experimenters, leveraging a federation of advanced 6G testbeds. To realize this vision, 6G-DALI brings together three communities: experts in 6G system design and experimentation, experts in AI and MLOps, and experts in DataOps and Gaia-X technologies. Their collaboration enables the development of an efficient, realistic, and trustworthy framework for end-to-end AI/ML experimentation in 6G environments. The envisioned framework is open and extensible, providing all the functionalities required by ML developers to train, validate, and deploy ML models for 6G systems. Central to this vision is the adoption of Gaia-X data management mechanisms for handling the complete data lifecycle, with data collected from physical 6G testbeds as well as a Digital Twin Testbed (DTT). This approach enables the creation of a 6G data space that supports secure, interoperable, and sovereign data sharing among 6G stakeholders.

To fulfil this vision, 6G-DALI is structured around two independent pillars: (i) AI experimentation as a service through **MLOps**, and (ii) data collection, preparation, and storage through **DataOps**. The MLOps pillar enables the training, finetuning, deployment, and continuous monitoring of AI/ML models on 6G testbeds, including drift detection and re-training using newly generated data. The DataOps pillar, in turn, provides mechanisms for discovering, generating, transforming, and curating high-quality datasets that are stored within the 6G Data Space and made available for AI experimentation. These processes are supported by adapters and connectors that link the framework to the testbeds and the 6G Data Space, ensuring seamless integration and automation.

2.2 Mapping with Relevant Research Initiatives

6G-DALI is aligned with key European 6G initiatives including SUNRISE-6G¹, 6G-BRICKS², 6G-SANDBOX³, ADROIT6G⁴ and 6G4SOCIETY⁵. Excluding 6G4SOCIETY, which focuses on the societal and sustainable values, the other projects focus on works in infrastructure, testbeds, and experimentation. It is here where 6G-DALI steps in and builds a complementary layer centered on the lifecycle management of AI and data-driven intelligence. By integrating MLOps, DataOps capabilities and digital twins, 6G-DALI enhances the usability of 6G experimentation platforms by enabling AI-native experimentation for future networks.

¹ <https://sunrise6g.eu/>

² <https://6g-bricks.eu/>

³ <https://6g-sandbox.eu/>

⁴ <https://adroit6g.eu/>

2.2.1 SUNRISE-6G

SUNRISE-6G is the flagship SNS C Phase 2 project and, as shown in Figure 1, aims to deliver a scalable federation solution for EU testbeds. The SUNRISE-6G facility seeks to provide access to heterogeneous resources, including 6G devices, 6G datasets and AI models made available by 14 European testbeds. This is achieved through decentralised paradigms. SUNRISE-6G proposes a scalable, open and standard-compliant approach to experimentation in a pan-European Federation of 6G infrastructures.

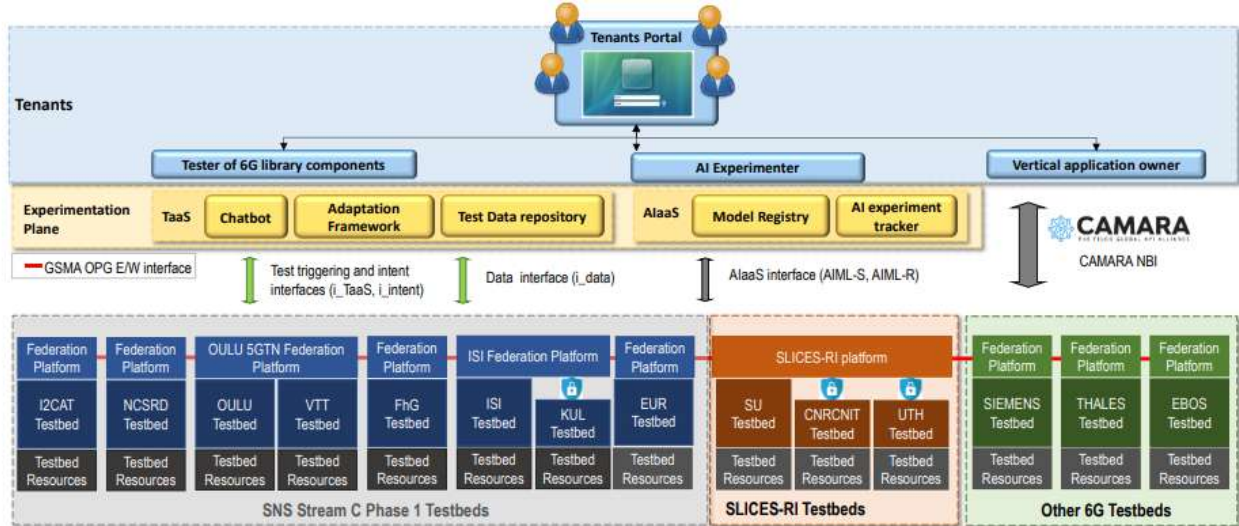


Figure 1: SUNRISE-6G Overall Reference Architecture [1]

SUNRISE-6G offers multiple “as a service” functions to experimenters as part of an Experimentation Plane through a Tenant Web Portal. In summary the SUNRISE-6G facility architecture consists of the following:

- **Tenant Portal and Experimentation plane:** SUNRISE-6G reflects a future multi-stakeholder 6G ecosystem. A Tenant web portal serves as an entry point for all interested stakeholders. SUNRISE-6G introduces a groundbreaking approach to Testing as a Service (TaaS) through the Intent-Driven Lifecycle Management (IDL) paradigm.
- **Testbed Federation plane:** SUNRISE-6G makes distinction between the Federation platform and the Resource domain in line with the GSMA Open Gateway initiative. As shown in Figure 1 each Federation Platform deployment can be leveraged to interconnect more than one testbed in the SUNRISE-6G federation, offering the flexibility to federate testbeds that lack E2E capabilities.
- **Testbed infrastructures with a library of user-facing 6G enablers:** 6G enablers are exposed becoming accessible to experimenters outside the project. Moreover, new complementary enablers are implemented and incorporated into the 6G Library, extending the testbed’s existing capabilities. The available enablers are externally accessible by experimenters through the SUNRISE-6G system portal.

Alignment with 6G-DALI: SUNRISE-6G delivers two key assets that will be leveraged in 6G-DALI: Experimentation as a Service and the AI-native plane. In 6G-DALI, these assets will be further enhanced through DataOps and the 6G Data Space to establish a closed-loop workflow that enables experiment description, deployment, and data collection. Moreover, 6G-DALI’s three testbeds are part of SUNRISE-6G. SUNRISE-6G builds a network-of-networks, whereas 6G-DALI focuses on data and AI support. In other words, 6G-DALI provides the data pipelines, 6G Data Space and digital twin environments needed to train and validate the models on the federated testbeds.

2.2.2 6G-BRICKS

The 6G-BRICKS facility introduces architectural innovations such as a disaggregated Management Plane and Operations Support System to support extendibility, evolvability and multi-tenancy.

The reference architecture as shown in Figure 2 introduces the following architectural tiers:

- **The Experimentation Plane:** acts as the entry point to the facility, where mature frontend elements as the System Portal, the Service Catalogue and intent engine will be leveraged and extended delivering intent-driven human-in-the-loop experimentation functionality.
- **The Management & Orchestration Plane:** is deployed at each facility site as a unified controllability framework, via the Domain Management and Orchestration (DMO) Framework. This layer provides explainable feedback to the experimenters for potential SLA breaches.
- **6G RAN infrastructure domain:** In this domain, breakthrough 6G RAN technologies are integrated as self-contained modules with O-RAN interfaces to ensure the openness and reusability of the developed components.
- **Alignment with 6G-DALI:** The Intent Translation engine initially developed within 6G-BRICKS will be leveraged in 6G-DALI to extend LLM knowledge and translate data requests into experiment. Moreover, 6G-BRICKS developed a solution for leveraging AI/ML algorithms for the automated scaling and placement of microservices at the Cloud Edge Continuum (CEC). These models will be packaged with MLOps pipelines as part of the 6G-DALI e2e AI framework. Furthermore, 6G-BRICKS's obtained CSI datasets for localisation that will be developed and integrated within 6G-DALI's framework focusing on trustworthiness. In more general terms, 6G-DALI acts complementary to 6G-BRICKS by leveraging the platform's measurements into its DataOps and MLOps pipeline for model training and drift monitoring.

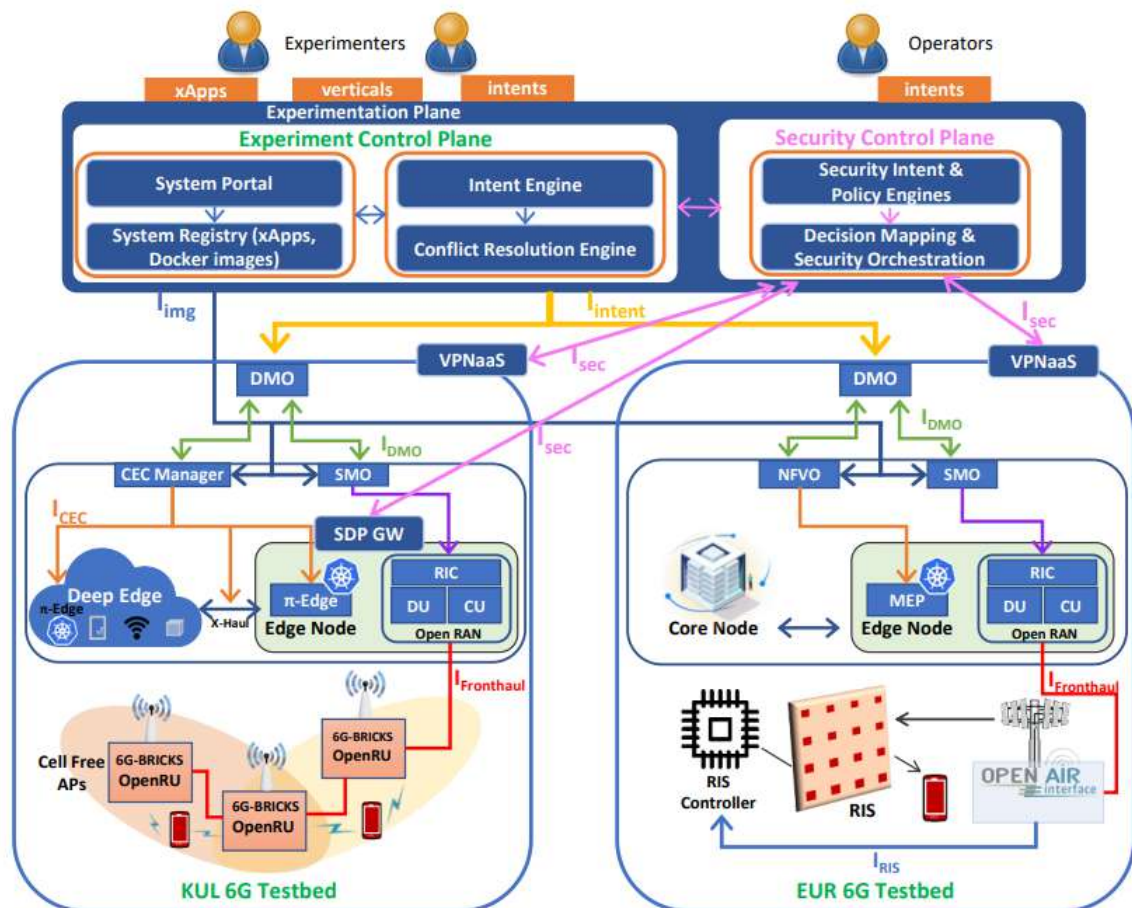


Figure 2: 6G-BRICKS Overall Architecture [2]

2.2.3 6G-SANDBOX

6G-SANDBOX's architecture emerges from experience gained through previous 5GPPP projects and is structured into three layers, as shown in Figure 3:

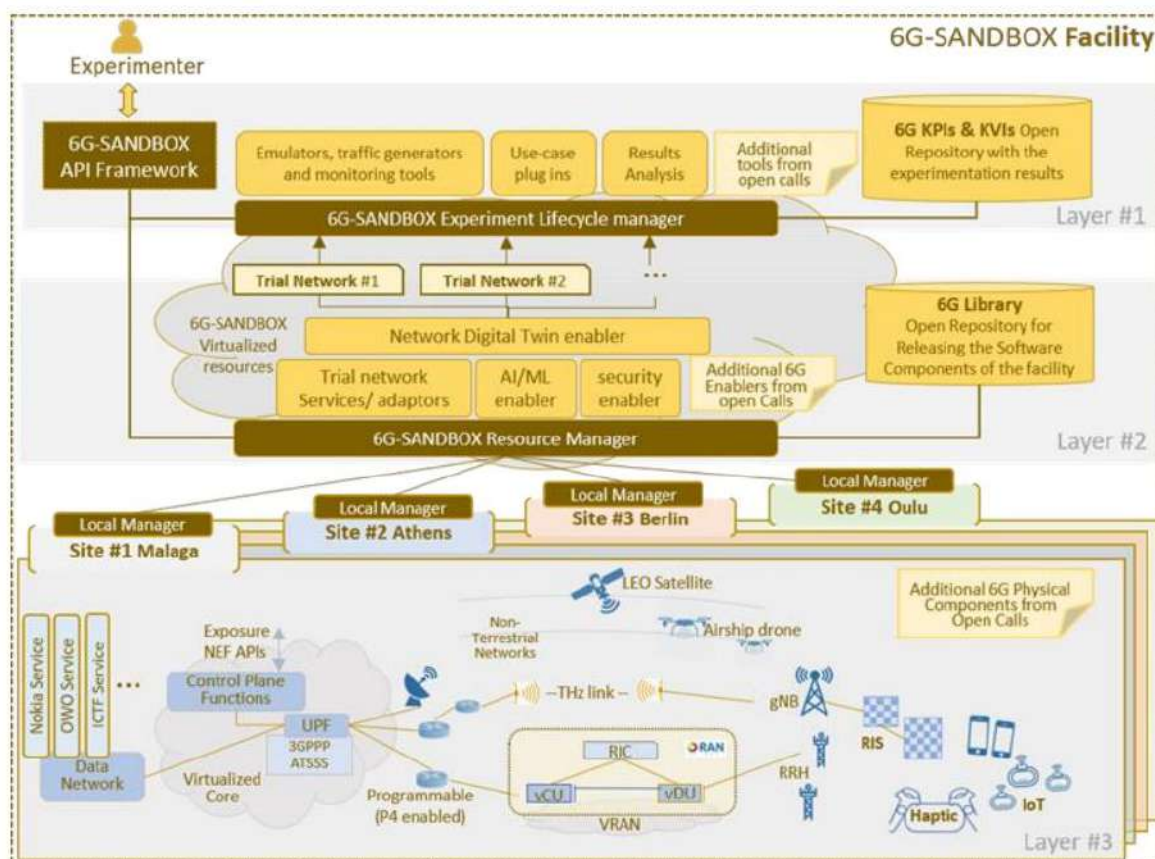


Figure 3: 6G-SANDBOX Facility [3]

- **The Physical Connectivity Layer (Layer#3):** it consists of the four diverse yet interoperable platforms in Malaga, Athens, Berlin and Oulu and provides the physical connectivity required for the Trial Networks.
- **The Resource Management Layer (Layer#2):** is responsible for the management of the virtualised resources within the facility. It also provides the necessary infrastructure for the creation and programmability of the Trial Networks.
- **The Experimentation Lifecycle (Layer#1):** this layer serves as the entry point for experimenters to conduct experiment-related activities.

Alignment with 6G-DALI: 6G-DALI acts complementary to 6G-SANDBOX, since it adds a higher layer of data intelligence by providing the data lakes, analytics engines and digital twin services that 6G-SANDBOX's framework requires for AI-driven optimisation. Moreover, 6G-SANDBOX (via its O-RAN testbeds) could supply data that feed into 6G-DALI's models training.

2.2.4 ADR0IT6G

ADR0IT6G implements a 6G architecture with a management and control framework based on distributed and secure CrowdSourcing AI. It addresses key challenges such as energy efficiency, the integration of extreme-edge devices, and the adoption of Non-Terrestrial Networks (NTN) [4].

The ADR0IT6G architecture, depicted in Figure 4, is built upon three main frameworks, operating over a programmable inter-computing infrastructure:

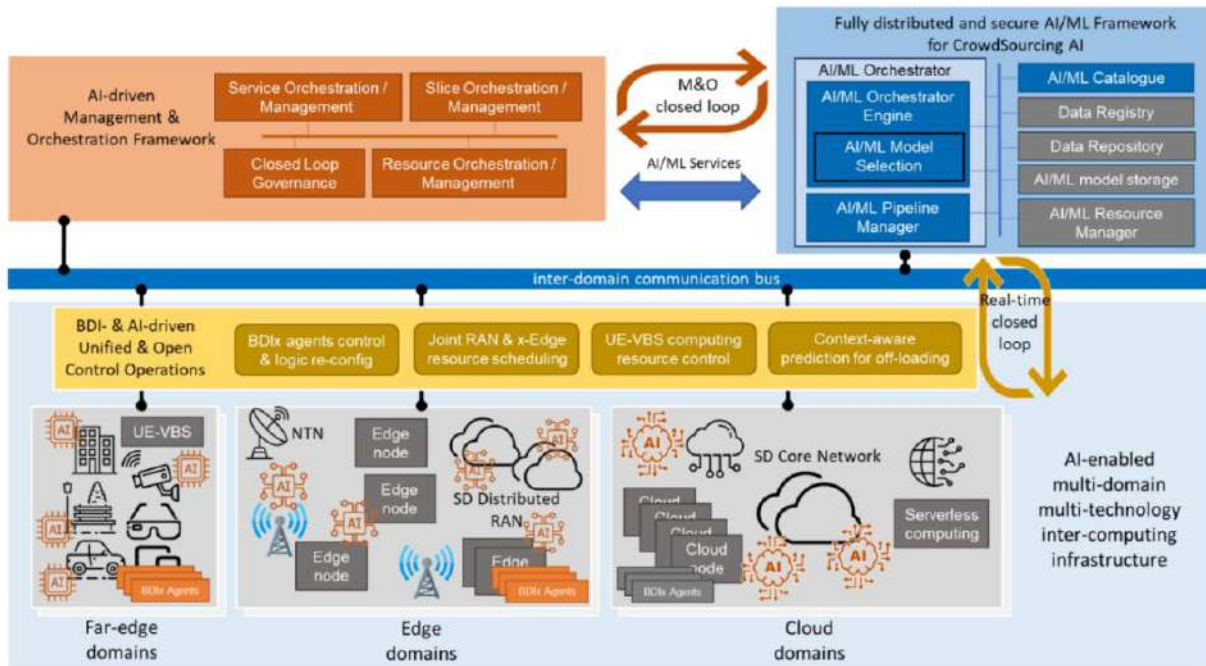


Figure 4: ADROIT6G Architecture [4]

- AI-driven Management & Orchestration Framework:** handles end-to-end automated management of the infrastructure including functionalities for services, slices, and resources orchestrations. Moreover, it includes a Closed-loop (CL) Governance module that automates the lifecycle of AI-driven closed loop for zero-touch automations and optimizations.
- Fully distributed and secure AI/ML Framework for CrowdSourcing AI:** core AI engine of ADROIT6G that provides a single point for Machine Learning Operations (MLOps) management allowing nodes to share AI models and resources. It supports advanced operations as model selection based on performances/resources trade-off.
- BDI- & AI-driven Unified & Open Control Operations:** manages real-time closed loops using Belief-Desire-Intention extended (BDIx) agents that reside on User Equipment’s (UE) and network nodes, making autonomous decision for resource control and offloading based on local beliefs and desires.

Alignment with 6G-DALI: The ADROIT6G architecture and the concepts at its core provide a foundational baseline for the 6G-DALI project, particularly regarding the establishment of a framework for the orchestration of distributed intelligence. 6G-DALI advances these concepts by introducing specialized operation methodologies to enhance the quality, trust, and efficiency of AI/ML models, integrating Reinforcement Learning Operations (RLOps), Transfer Learning (TL), automated Hyperparameter Optimization (HPO), Explainable AI (XAI) and Robust AI methodologies. Moreover, 6G-DALI, extends the concepts of MLOps orchestration into meta-orchestration, coordinating pre-existing MLOps stacks across distributed testbeds to deploy and manage AI/ML pipelines.

2.2.5 6G4SOCIETY

The 6G4Society project is a Coordination Action SNS JU project which aims to ensure that societal and sustainable values are properly embedded into the development of 6G technology, bringing a sociological perspective to technological development.

Alignment with 6G-DALI: 6G4Society can influence the architectural requirements for 6G-DALI’s in terms of fairness metrics and sustainable optimisation. Moreover, the values in 6G4Society can be translated into quantifiable goals for 6G-DALI.

2.3 Compliance with Relevant Standards

The 6G-DALI aligns with relevant standardization activities and frameworks that are shaping the evolution of AI-enabled 6G systems. This section highlights how 6G-DALI leverages and complements key standards from 3GPP, ETSI, and Gaia-X to ensure interoperability, data sovereignty, and secure AI lifecycle management.

2.3.1 3GPP

3rd Generation Partnership Project⁶ (3GPP) provides a global, collaborative, and industry-led environment for the development of technical specifications for mobile telecommunications. This section reports concepts from three of the most relevant to the 6G-DALI, Technical Specifications (TS) by 3GPP regarding the standardization of AIaaS service exposure (Section 2.3.1.1), AI/ML management capabilities (Section 2.3.1.2), and the evolution of AI/ML network data analytics functions (Section 2.3.1.3).

2.3.1.1 AIMLE Enablement Layer

The 3GPP AIML Enablement (AIMLE) Layer [5], as depicted in Figure 5, provides a functional architecture to support AI/ML functionalities exposure within the Vertical Application Layer (VAL). This framework facilitates the interactions between the AIMLE Client (residing on the UE) and the AIMLE Server (located on the network side) via the AIML-UU interface, enabling AI/ML capabilities as model training, deployment, and inference. Key point of the architecture is the ML Repository, connected to the AIMLE Server through the AIML-R, which provides an endpoint for management of Models and ML participants.

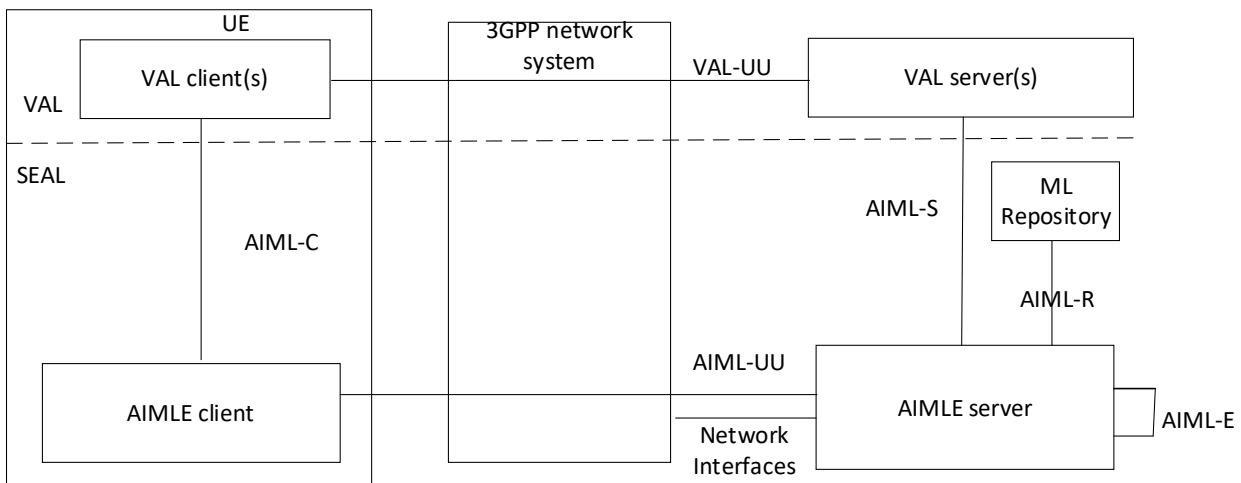


Figure 5: AIMLE Enablement Layer

Key concepts of this framework will be used as reference to implement the AIaaS functionalities envisioned in the 6G-DALI architecture, defining endpoints for AI/ML models management as model retrieval, training, models performance monitoring, and updated. While AIMLE provides information about the signaling framework, 6G-DALI will enhance this with operational methodologies as RLOps, XAI, and automated HPO, which are not explicitly detailed in the 3GPP standard. Moreover, while AIMLE focuses on 3GPP-specific domain orchestration, 6G-DALI introduces the concept of multi-domain meta-orchestration to coordinate across multi-testbed and potentially non-3GPP environments.

2.3.1.2 AI/ML Management

The 3GPP TS 28.105 [6] defines a management and orchestration framework for AI/ML in 5G/6G networks, treating AI/ML capabilities as managed services, allowing operators to standardize how they request, monitor, and update models across the network. The standard defines a cyclical lifecycle for

⁶ <https://www.3gpp.org/>

AI/ML models, depicted in Figure 6, that ensures that models are continuously managed during their lifecycle, from the training to the deployment for inference. The standard goes beyond simple learning methodologies, providing support to complex operations as ML-Knowledge-based Transfer Learning (MLKTL), Federated Learning (FL), Reinforcement Learning (RL), and Distributed Learning.

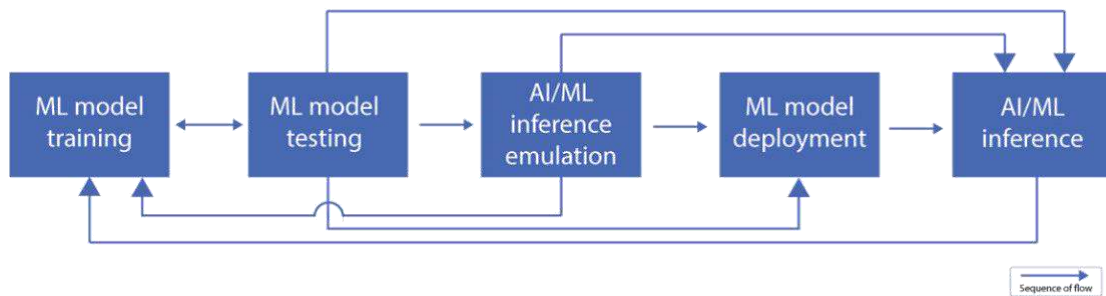


Figure 6: AI/ML Management Model Lifecycle

6G-DALI will adopt the TS 28.105 Information Model as foundational model for the 6G-DALI MLOps framework. To ensure standards compliance and interoperability with future 6G-networks, the standard's base entities will be strictly maintained (e.g., MLModel, TrainingProcess, DeploymentProcess). Meanwhile, the 6G-DALI MLOps information model will extend the TS28.105 framework by introducing new, complementary entities. These additions will support advanced MLOps paradigms such as Trustworthy AI, and Meta-orchestration features that are currently gaps in the standard.

2.3.1.3 NWDAF

The Network Data Analytics Function (NWDAF) is a key architectural component of the 3GPP Service-Based Architecture (SBA), responsible for collecting, processing, and analysing data from 5G Core Network Functions (NFs) to generate analytics that support intelligent network operation. NWDAF enables both reactive and proactive decision-making by providing statistical insights, predictions, and recommendations to authorized consumers via standardized service-based interfaces. Introduced in 3GPP Release 15 as a foundational analytics function, NWDAF initially focused on descriptive and predictive analytics with limited support for machine learning lifecycle considerations. Subsequent releases progressively enhanced its capabilities: Release 16 extended analytics consumption and data collection mechanisms, while Release 17 introduced a more mature functional decomposition, including the explicit definition of the Model Training Logical Function (MTLF) and the Analytics Logical Function (AnLF). This evolution reflects 3GPP's intent to operationalize machine learning within the core network by clearly separating model training from inference and analytics exposure, while enabling service-based interaction with other NFs and management systems.

Within the 6G-DALI, we aim to assess and demonstrate NWDAF compliance with 3GPP Release 17 by addressing the practical realization of MLOps workflows inside the NWDAF architecture. This enables standardized model training, versioning, storage, deployment, and monitoring, aligned with the logical separation and service exposure principles introduced in Release 17. As a result, 6G-DALI provides a unified and scalable NWDAF MLOps framework that aligns 3GPP architectural specifications with operational best practices, establishing a concrete foundation for automated, intelligent analytics in the transition from advanced 5G systems toward 6G.

2.3.2 Gaia-X

Gaia-X represents a European initiative establishing a federated, secure, and trustworthy data infrastructure ecosystem that enables data sovereignty while promoting data sharing and interoperability across organizational and national boundaries. The 6G-DALI Data Space architecture leverages Gaia-X principles and technical specifications to ensure data sovereignty across the federated

testbed infrastructure while enabling secure, policy-controlled data exchange supporting collaborative 6G research.

The 6G-DALI Data Space directly implements core Gaia-X architectural principles as foundational design choices [7] :

- **Federation for Testbed Autonomy:** Gaia-X defines federation as enabling independent participants to maintain sovereignty while participating in collaborative ecosystems. The 6G-DALI federated catalogue architecture leverages this principle, allowing each testbed of the facility to maintain autonomous control over metadata publication workflows, quality standards, and access policies while contributing to unified dataset discovery. This federation approach respects the diverse governance requirements of international research partners while enabling seamless data sharing across consortium boundaries.
- **Sovereignty-Preserving Data Control:** Gaia-X mandates that data providers retain control over usage rights, access policies, and processing conditions. The 6G-DALI Data Governance component leverages this principle through policy-based access control enabling testbed-specific governance rules. Dataset metadata includes explicit license terms, access restrictions, and usage policies that are enforced when users discover or retrieve data, ensuring that testbed operators maintain full authority over how their experimental data is accessed and utilized.
- **Trust Through Transparency:** Gaia-X emphasizes verifiable trust through transparent, auditable mechanisms. The 6G-DALI architecture leverages this principle by implementing cryptographically signed metadata ensuring integrity, comprehensive provenance tracking using W3C PROV-O vocabulary capturing complete data lineage from raw experiments through processing pipelines, and automated quality assessment providing reliability indicators. This transparency enables researchers to verify dataset authenticity, understand data provenance, and assess data quality before committing computational resources to analysis or AI/ML training.
- **Interoperability Through Standards:** Gaia-X requires standardized protocols and vocabularies enabling seamless interaction between heterogeneous systems. The 6G-DALI Data Space leverages DCAT-AP version 3.0.1 [8] for metadata structures ensuring compatibility with European data infrastructure, W3C SPARQL for semantic queries enabling federated catalogue searches, and OAuth 2.0/OIDC for authentication supporting federated identity across institutional boundaries. This standards-based approach ensures that 6G-DALI datasets can be discovered and accessed through broader European data portals including data.europa.eu.

While leveraging Gaia-X principles, the 6G-DALI architecture complements standard Gaia-X building blocks with innovations addressing specific 6G research infrastructure requirements:

- **Federated Catalogue with AI/ML Model Integration:** Gaia-X specifies federated catalogues for data asset discovery using standardized metadata. The 6G-DALI Federated Catalogue complements this baseline by extending DCAT-AP to include AI/ML Models as first-class catalogue entities alongside datasets, capturing model architecture descriptions, training configurations, performance metrics, and explicit relationships to training datasets through *trainedOn* properties. This extension positions the Data Space as a unified discovery interface for both training data and trained intelligence, addressing AI lifecycle management requirements not fully specified in baseline Gaia-X catalogues focused primarily on data assets.
- **Sovereign Data Exchange with Experiment Automation:** Gaia-X mandates controlled data sharing through connectors enforcing usage policies [9] . The 6G-DALI Data Space Connectors complement this by implementing Eclipse Data Space Components (EDC) [10] technology aligned with Gaia-X specifications while adding experiment-triggered automation. Rather than requiring manual data sharing initiation, 6G-DALI connectors automatically detect experiment completion, transfer dataset files to the Data Lake, and register metadata in the Data Space without human intervention. This automation addresses the high-throughput experimental

workflows characteristic of 6G testbed environments where manual publishing would create unacceptable delays between data generation and research utilization.

Gaia-X conformance positions 6G-DALI for interoperability with broader European research data infrastructure. The architecture's DCAT-AP compliance enables potential federation with data.europa.eu and national open data portals across EU member states. OAI-PMH harvesting interfaces allow external catalogues to discover 6G-DALI datasets, extending research impact beyond consortium boundaries. Alignment with IDSA Reference Architecture Model components ensures compatibility with other data space initiatives in telecommunications, manufacturing, and mobility sectors, facilitating potential cross-domain research collaborations. As Gaia-X evolves and additional technical specifications emerge, the 6G-DALI Data Space's foundational alignment ensures the project can adopt new capabilities while maintaining compatibility with the European data space ecosystem, providing long-term sustainability for 6G research data infrastructure.

2.3.3 ETSI

ETSI (European Telecommunications Standards Institute) provides an open, inclusive and collaborative environment for the development of globally applicable ICT standards⁷.

2.3.3.1 Technical Committee Data

In response to the growing strategic importance of data technologies, TC DATA⁸ was established to reinforce ETSI's role in data solutions. TC DATA supports the deployment and operation of distributed data solutions across connectivity (data in transit), storage (data at rest) and compute (data in process), while promoting open technologies, interoperability, privacy and security, and addressing key European policy and regulatory frameworks such as the European Data Act and the data-related aspects of the European AI Act.

TC DATA serves as a centre of expertise for data infrastructures, services and applications, developing and maintaining technical standards that enable data interoperability and semantic interoperability. Its work is closely coordinated with other ETSI technical bodies and international standardization organizations, as well as with relevant open-source initiatives. Key activities include reference implementations and interoperability testing, the maintenance and evolution of widely adopted specifications such as SAREF⁹ and NGSI-LD¹⁰, and the integration of outputs from oneM2M¹¹, ensuring harmonization and avoiding duplication across the data standardization landscape.

The objectives of 6G-DALI are strongly aligned with the mission and scope of TC DATA, as the project focuses on the design and implementation of an end-to-end DataOps and MLOps framework for AI-enabled 6G networks. By enabling automated, intelligent and trustworthy data-driven workflows, 6G-DALI supports the creation of a 6G Data Space that ensures secure, interoperable and sovereign data sharing across the complete data lifecycle. This approach directly leverages TC DATA principles on distributed data solutions, interoperability, privacy and regulatory compliance.

In this context, a new Work Item (WI)¹² has recently been approved within TC DATA to address data sovereignty, governance and trust in cross-domain scenarios. It will define a reference governance architecture and specify machine-readable data contracts, together with federated authentication and fine-grained authorization mechanisms.

⁷ <https://www.etsi.org/about>

⁸ <https://www.etsi.org/committee/2519-data>

⁹ <https://saref.etsi.org/>

¹⁰ <https://ngsi-ld.org/>

¹¹ <https://www.onem2m.org/>

¹² https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=78208

2.3.3.2 OpenOP

ETSI Software Development Group OpenOP (SDG OOP¹³) is an open-source operator platform developed within ETSI that enables federation of operator networks and testbeds, along with standardised capability exposure APIs. Originating from foundational work in the EU-funded SUNRISE-6G project, OpenOP aligns with GSMA requirements and APIs for federation and supports the exposure of CAMARA service APIs, while enabling integration with a wide range of southbound platforms, such as Kubernetes, 3GPP Network Exposure Function (NEF), ETSI MEC, and Network Functions Virtualization (NFV) orchestration frameworks. OpenOP is designed as a neutral, non-commercial, and standardised experimentation framework, fostering collaboration among research organizations, SMEs, and telecom stakeholders. Through close interaction with standardization bodies and open-source communities, including GSMA, LF CAMARA, and 3GPP, OpenOP contributes to the evolution of interoperable, AI-native network platforms and complements the broader ETSI open-source ecosystem, including initiatives such as OpenCAPIF¹⁴, OpenSlice¹⁵, Open-Source MANO¹⁶, MEC¹⁷, NFV¹⁸, and ZSM¹⁹.

In the context of 6G-DALI, OpenOP provides a key, standards-aligned foundation for AI service exposure, orchestration, and federation across domains. 6G-DALI aligns with OpenOP concepts to support the secure and sovereign exchange of AI assets, including datasets, models, and data processing pipelines, across heterogeneous 6G testbeds. In particular, **OpenOP's AI service exposure layer** and its meta-orchestration framework are closely aligned with the 6G-DALI MLOps and DataOps pillars, enabling intent-driven access to AI services through unified exposure APIs and governed data space services compliant with International Data Spaces Association (IDSA) and Gaia-X principles. This alignment enables 6G-DALI to support cross-domain AI workflows, covering data preparation, model discovery, training, update, and deployment across telecom, cloud/edge, and vertical ecosystems, while preserving data sovereignty, policy enforcement, and interoperability. By bridging user intents, standardised APIs, and AI asset exposure, OpenOP directly reinforces the 6G-DALI objective of establishing an AI Plane Federation across networks and 6G facilities, accelerating trustworthy and scalable AI experimentation for 6G systems.

2.3.4 O-RAN

O-RAN promotes the use of standardized, open interfaces, allowing for more flexible and efficient deployment and management of RAN infrastructure. A central element of the O-RAN architecture is the Real-time Intelligent Controller (RIC), which optimizes RAN performance and flexibility by providing intelligent, real-time control over network functions.

The RIC consists of two layers: the near-real-time RIC and the non-real-time RIC. The near-real-time RIC operates with low latency (typically within 10 milliseconds) and manages dynamic functions such as handovers, interference management, and resource allocation to ensure optimal performance and user experience. In contrast, the non-real-time RIC operates at higher latency and handles tasks that require data aggregation and machine learning, such as long-term network optimization, traffic forecasting, and policy management.

O-RAN specifies the E2 interface, which enables the RIC to control the logic exposed by E2 nodes (including gNB, DU, and CU). The E2 interface is a logical interface consisting of two protocols: the E2 Application Protocol (E2AP) and the E2 Service Model (E2SM). E2AP is a procedural protocol that coordinates communication between the near-real-time RIC and E2 nodes, providing a basic set of

¹³ <https://oop.etsi.org/>

¹⁴ <https://ocf.etsi.org/>

¹⁵ <https://osl.etsi.org/>

¹⁶ <https://osm.etsi.org/>

¹⁷ <https://www.etsi.org/technologies/multi-access-edge-computing>

¹⁸ <https://www.etsi.org/technologies/nfv>

¹⁹ <https://www.etsi.org/technologies/zero-touch-network-service-management>

services. E2AP messages can carry different E2SMs, which implement specific functionalities such as reporting RAN metrics or controlling RAN parameters.

Each E2 node exposes a set of RAN functions, representing the services or capabilities it supports. For example, DUs from different vendors may offer various control options, depending on which parameters and functionalities can be adjusted, as well as their ability to collect and report different performance metrics. The data models exchanged between E2 nodes and the RIC via the E2 interface are called Service Models (SMs).

xApps are applications that run on top of the RIC, which acts as a relay between the xApps and the E2 nodes. Messages from xApps to E2 nodes are called E2 control messages, while those from E2 nodes to xApps are called E2 indication messages. These messages incorporate the SMs available at the RIC and the E2 nodes. Several SM models are standardized by O-RAN, including two key models: E2SM-KPM (Key Performance Measurement) and E2SM-RC (RAN Control). The E2SM-KPM model is used for monitoring and reporting KPIs, such as throughput, latency, and cell load, providing insights into network performance. Meanwhile, E2SM-RC allows the RIC to control specific RAN functions, including beamforming, scheduling, and handover management, to optimize resource utilization. xApps are applications that operate on top of the RIC, which serves as an intermediary between the xApps and the E2 nodes. Communication from xApps to E2 nodes takes the form of E2 control messages, while messages sent from E2 nodes to xApps are known as E2 indication messages. These messages utilize the Service Models (SMs) available at both the RIC and the E2 nodes.

O-RAN has standardized several SM models, with two key examples being E2SM-KPM (Key Performance Measurement) and E2SM-RC (RAN Control). The E2SM-KPM model is designed for monitoring and reporting key performance indicators (KPIs) such as throughput, latency, and cell load, thereby providing valuable insights into network performance. In contrast, the E2SM-RC model enables the RIC to control specific RAN functions, including beamforming, scheduling, and handover management—to optimize resource utilization.

In 6G-DALI, the O-RAN architecture is utilized by integrating the RIC (OAI FlexRIC) with DALI's Digital Twins (DT) using the E2 interface. This setup enables the deployment of xApps for two main purposes:

- DataOps: Run experiments, collect RAN KPIs using the xApp KPM SM, and upload the collected KPIs to DALI's DataSpace via an adapter.
- RLOps: Deploy and evaluate Reinforcement Learning (RL) agents as xApps on the Digital Twins. Collect RAN data using KPM SM and execute agent actions using RC SM.

This integration streamlines both data operations and reinforcement learning workflows within the 6G-DALI framework.

3 6G-DALI High Level Architecture

This section provides a high-level overview of the 6G-DALI overall architecture. Building upon the project context, objectives, and SoTA discussed in section 2, it presents an architectural view that captures the fundamental building blocks of the system and their interactions. The subsections that follow first detail the user requirements that guide the architectural design. Subsequently, the high-level logical architecture of 6G-DALI is introduced, identifying the main architectural blocks and their roles within the overall system. These blocks constitute the foundation of the architecture and are further refined and analysed in detail in section 4.

3.1 User Requirements

The 6G-DALI requirement elicitation methodology, outlined in Deliverable 2.2, captured the project stakeholders' requirements identified in Deliverable 2.1. The subset of requirements that directly serve as guiding principles for defining the architectural blocks and their interactions is presented in Table 2.

Table 2: User Requirements regarding 6G-DALI overall architecture

UR ID	UR Description	Architectural Block
UR1	6G-DALI shall support the end-to-end management of ML model lifecycles across multiple tools and environments.	MLOps
UR2	The system shall enable framework-agnostic integration of MLOps components, including experiment tracking, artifact storage, and model serving.	MLOps
UR3	The system shall allow secure and policy-driven access to models and orchestration resources.	MLOps
UR4	The system shall provide a common set of API tools that ensure seamless integration between MLOps, DataOps, Data Space and Infrastructure components.	MLOps, DataOps, Data Space, Infrastructure
UR5	6G-DALI shall allow experimenters to express data requests to the 6G-DALI user interface written in human language.	DataOps
UR6	6G-DALI shall use machine learning algorithms to clean, enrich and transform the generated datasets to improve their quality.	DataOps
UR7	6G-DALI shall be able to trigger on-demand experiments on testbeds and DTT for data generation.	DataOps
UR8	6G-DALI shall allow users to use the Gaia-X compatible federated catalogue to search and extract datasets, data transformation and augmentation services, and ML models.	Data Space
UR9	6G-DALI shall include rich metadata annotation for all datasets, ML models and services in the Data Space.	Data Space
UR10	The system shall enforce identity, trust, and security policies based on predefined user roles and data set classifications.	Data Space

UR11	The adaptation layer shall provide a common set of APIs to interact with the adapters deployed on various testbeds, including Digital Twin, to request experiment execution and collect datasets.	Infrastructure
UR12	6G-DALI shall define a consistent data model that the Digital Twin will adhere to and ensure compliance with the requisite Data Spaces and Catalogues.	Infrastructure
UR13	The adaptation layer shall facilitate storing data collected from testbeds into the 6G data lake and the corresponding metadata to the 6G data space through the adapters.	Infrastructure

3.2 High Level Logical Architecture

This section introduces the high-level logical architecture of 6G-DALI. It identifies the main architectural blocks, their roles, and the logical relationships between them, offering an abstract yet coherent view. The high-level logical architecture is designed to support the user requirements defined in section 3.1 while remaining consistent with the relevant standards. Rather than providing an exhaustive description of internal components, this section focuses on the roles and interfaces of the main architectural blocks and on how they collectively enable the 6G-DALI objectives and functionalities. Each of these blocks is further elaborated in section 4, where detailed logical architectures are presented, including internal sub-components and their interactions.

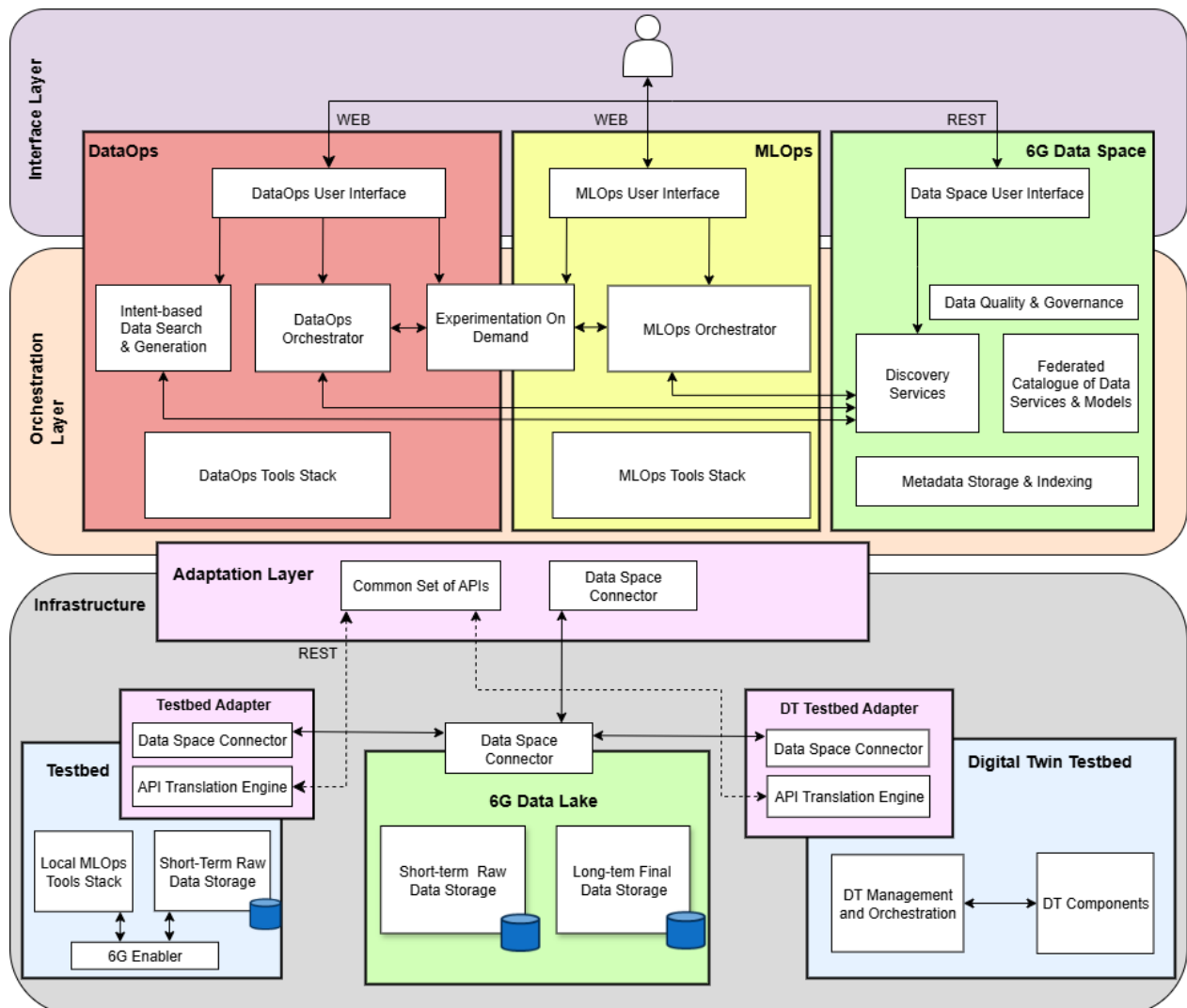


Figure 7: 6G-DALI Overall Architecture

The 6G-DALI architecture, shown in Figure 7, is structured into four main layers: the **Interface Layer**, the **Orchestration Layer**, the **Adaptation Layer**, and the **Infrastructure Layer**. These layers define the 6G-DALI's core structure and facilitate the interaction between various components. Within these layers, we identify seven architectural blocks: **DataOps**, **MLOps**, **6G Data Space**, **Adaptation Layer**, **Testbeds**, **Digital Twin Testbed** and **6G Data Lake**. The detailed architectures of these components, including their sub-components and high-level interactions, are presented in sections 4 and 5, correspondingly.

1. **Interface Layer**. At the top of the architecture lies the **Interface Layer**, which serves as the primary point of interaction for experimenters. This layer is responsible for providing seamless communication between the user-facing components and the underlying orchestration and infrastructure layers. Within the interface layer the **DataOps**, **MLOps**, and **6G Data Space** components play crucial roles in managing and enabling the flow of data, ML models, and data-driven services.
 - **The DataOps** components enable experimenters to submit data requests and request data transformation or augmentation algorithms, which trigger Extract – Load – Transform (ELT) pipelines.
 - **The MLOps** components bridge the gap between model development and deployment, allowing experimenters to request available models, trigger hyperparameter optimization (HPO) and reinforcement learning (RL) experiments, or perform transfer learning (TL) tasks.
 - **The 6G Data Space** components allow authorized users to query the federated catalogue and retrieve specific data, models, or transformation services.
2. **Orchestration Layer**. The **Orchestration Layer** sits between the Interface and Infrastructure Layers, coordinating complex workflows and operations across the system. It acts as the central brain of the architecture, enabling resource allocation and orchestrating interactions between **DataOps**, **MLOps**, and **6G Data Space** components.
 - **The DataOps components** interpret user requests (including those in natural language) and query the 6G Data Space catalogue. They also trigger data generation experiments on testbeds and extract datasets from the 6G Data Lake. Additionally, they apply data transformation and augmentation algorithms to datasets on-demand.
 - **The MLOps components** manage the entire ML model lifecycle, ensuring framework-agnostic integration of MLOps tools, providing secure access to models, and triggering HPO experiments.
 - **The 6G Data Space components** enable secure and compliant, with Gaia-X and IDSA principles, sharing of data resources. These include a federated catalogue with Gaia-X complaint metadata for datasets, models, and services. They also handle user authentication, authorization, metadata matching, and access decisioning.
3. **Adaptation Layer**. The **Adaptation Layer** interfaces the 6G-DALI framework with testbeds, enabling data generation experiments through a common set of APIs and seamless integration of models and data through data space connectors.
4. **Infrastructure Layer**. The **Infrastructure Layer** provides the foundational physical and virtual resources and includes all necessary components to support experiments that generate data, as well as for its local storage and processing. The key architectural block in this layer is Infrastructure, which encompasses the following sub-components:
 - **Testbed**: The testbeds run experiments that provide the requested data and may also provide MLOps tools.
 - **Digital Twin Testbed (DTT)**: The DTT is a virtual environment that mirrors real 6G testbeds, enabling large-scale experiments.
 - **6G Data Lake**: The 6G Data Lake serves as the data repository, storing both raw and processed (final) data.

It should be clarified that although participating testbeds may include computational resources (e.g., GPUs), such resources are managed locally and are not exposed as on-demand training or inference infrastructure to external experimenters.

Each of these blocks plays a critical role in realizing the 6G-DALI system's objectives. An overview of the global interfaces enabling interactions among the main architectural components is provided in Table 3. The relationship between these blocks, along with their interactions, are illustrated in the following sections.

Table 3: 6G-DALI architecture global interfaces

Interface ID	Interface Description	Involved Components
MLOps User Interface	Exposes AI/ML services to experimenters (e.g., model discovery, hyperparameter optimization, transfer learning, etc.)	Experimenter, MLOps User Interface
DataOps User Interface	Enables experimenters to express data-related intents, including data transformation or augmentation requests.	Experimenter, DataOps User Interface
Data Space User Interface	Enables authorized experimenters to discover, query, and access datasets, models, and services via the federated 6G data space catalogue.	Experimenter, Data Space User Interface
Experiment Execution Interface	Enables triggering data generation experiments to testbeds.	Adaptation Layer, Testbeds
Data Space Connector Interface	Supports secure, sovereign, and interoperable data transfer between the data lake and the testbeds.	Data Space Connector

4 Detailed Architecture of 6G-DALI

This section presents the detailed architecture of the 6G-DALI framework. Based on the high-level architectural view introduced in section 3, it elaborates on the main architectural blocks, namely MLOps, DataOps, 6G Data Space, Adaptation Layer, Testbeds, DTT, and 6G Data Lake, by analysing their internal structure, key functionalities, and interactions. This detailed architectural description supports the architectural process perspectives and end-to-end workflows presented in section 5. DataOps

4.1 DataOps

The role of DataOps in 6G-DALI is to establish the foundational mechanisms for managing, processing, and optimizing data workflows across distributed 6G environments. It serves as a core architectural block enabling seamless orchestration of data collection, transformation, and provisioning process that underpin AI experimentation and data-driven analytics within the 6G-DALI framework. DataOps unifies essential data management capabilities, including dataset discovery and generation, data quality management, data transformation and augmentation, and data provisioning, ensuring that high-fidelity and context-aware data can be efficiently accessed and utilized for model development and validation.

4.1.1 Functional Architecture

This section presents the detailed functional architecture of the **6G-DALI DataOps block**, elaborating on its internal components and their roles in supporting the end-to-end data lifecycle. Figure 8 depicts the detailed DataOps architecture, illustrating how the DataOps functionalities are realized through a set of interconnected components that collectively enable intent-driven data discovery, generation, transformation, and provisioning across distributed 6G environments. The DataOps architecture follows an Extract-Load-Transform (ELT) paradigm tailored to 6G experimentation, integrating dataset discovery and generation mechanisms, secure and interoperable data ingestion from heterogeneous testbeds, and advanced data processing pipelines for cleaning, augmentation, validation, and quality insurance. The functional architecture support both historical “cold” datasets accessed through the 6G Data Space and on-demand generated “hot” datasets produced by ongoing experiments, enabling flexible data management strategies. The individual DataOps components and their interactions are described in the following subsections, where each component’s functionality and contribution to the overall DataOps workflow are analysed in detail.

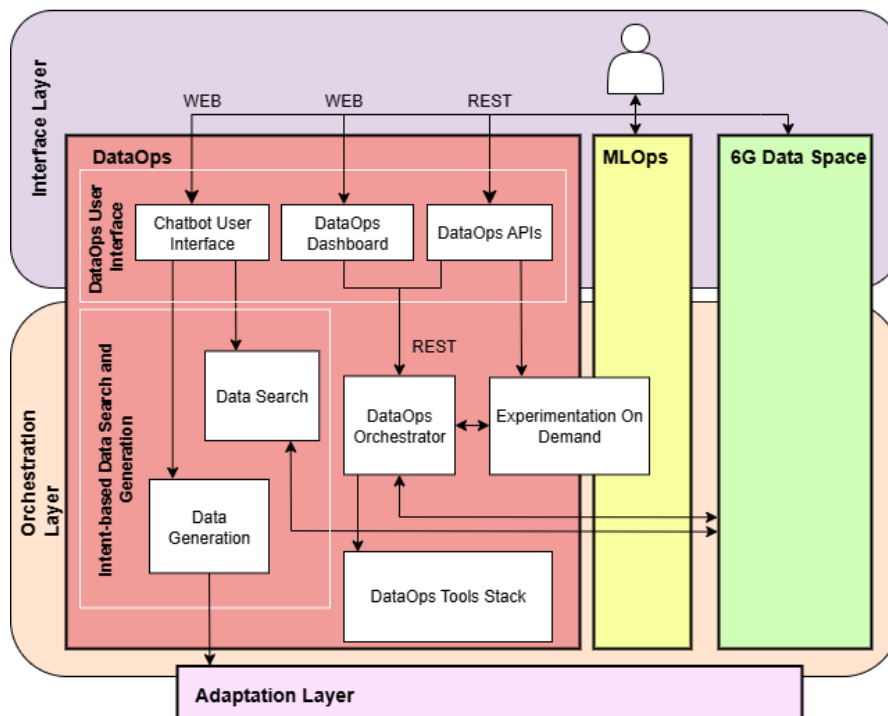


Figure 8: 6G-DALI DataOps architecture

4.1.2 DataOps User Interface

The DataOps User Interface serves as the primary human interaction point for experimenters and data engineers to request data or design, configure, and execute data processing pipelines within the 6G-DALI framework. It includes three blocks: the chatbot user interface, the dataops dashboard, as well as the dataops APIs. These blocks are described and analysed in the following sub-sections.

Baseline: The DataOps User Interface builds on top of Apache Airflow's API and programmatic DAG definition capabilities as its foundational technology. Apache Airflow provides the underlying workflow orchestration engine, supporting DAG-based pipeline specification, task dependency management, distributed execution across worker nodes, and comprehensive monitoring and logging capabilities. The baseline includes Airflow's native interfaces for Python-based DAG construction, task operators for various data processing operations, and web-based monitoring dashboards showing execution status and logs.

Innovation: The 6G-DALI implementation extends these baseline functionalities through several key innovations addressing the gap between Apache Airflow's developer-centric programmatic interface and the needs of 6G experimenters who require accessible data processing capabilities without deep workflow programming expertise. Specifically, it offers:

- **Data Space-Integrated Service Discovery:** The interface dynamically retrieves available Data Processing Services from the 6G Data Space catalogue rather than presenting a static, hard-coded list of operations. This integration means that as new data processing capabilities are developed and registered in the Data Space, they automatically become available for pipeline composition without interface modifications. Service metadata including input format requirements, output format guarantees, and processing semantics inform the interface's validation logic, preventing incompatible task connections and providing contextual guidance during pipeline construction.
- **Dataset-Aware Pipeline Configuration:** Integration with Data Space dataset discovery enables users to select input data through metadata-based queries rather than manually specifying file paths or storage locations. Users can construct input selections which the interface translates into Data Space queries, retrieving matching datasets and automatically configuring pipeline input tasks with appropriate data access parameters. This abstraction shields users from storage infrastructure details while enabling sophisticated data selection based on rich metadata attributes.

4.1.2.1 Chatbot User Interface

The Chatbot Agent serves as the primary user-facing entry point of the agentic AI framework. Its main role is to interact with users through free-form natural language, allowing them to express high-level intents without requiring technical expertise or predefined query structures. Leveraging a local large language model (LLM), the Chatbot Agent captures and interprets user inputs, extracts domain-specific entities and keywords, and evaluates whether the expressed intent is sufficiently complete to be processed downstream. In cases where ambiguity, missing parameters, or underspecified requirements are detected, the agent proactively initiates an intent clarification dialogue to refine the user request. Once the intent is deemed complete, the agent confirms the extracted information with the user and hands over control to the Search Agent. All conversational context and intermediate reasoning results are maintained within an AgentState structure, which records user messages, agent responses, extracted entities, and intent-completeness status, ensuring continuity and traceability across interactions.

4.1.2.2 DataOps Dashboard

This component sits atop the DataOps Orchestrator and Tool Stack, translating user needs into executable Apache Airflow DAGs (Directed Acyclic Graphs) that orchestrate data transformation workflows across the distributed testbed infrastructure.

The interface enables experimenters to construct data processing pipelines through DAG composition, selecting processing tasks from a catalogue of available Data Processing Services that are registered in the 6G Data Space. These services represent reusable data transformation operations including data cleaning, format conversion, aggregation, augmentation, quality assessment, and domain-specific processing tailored to 6G network data. Users compose pipelines by connecting these services into logical sequences, defining data flow dependencies and transformation steps without requiring direct knowledge of Apache Airflow syntax or Python DAG construction.

Pipeline configuration capabilities enable experimenters to specify input datasets by querying the 6G Data Space catalogue, selecting source data based on metadata attributes such as testbed origin, temporal coverage, quality scores, and thematic categories. Output specifications define where processed datasets should be stored in the Data Lake and what metadata should be registered in the Data Space upon pipeline completion. Parameter configuration for individual processing tasks allows users to customize transformation behaviour, adjusting thresholds, selecting algorithms, and specifying output formats appropriate to their experimental objectives.

Pipeline execution management provides interfaces for submitting configured pipelines to the Apache Airflow cluster, monitoring execution progress through real-time status updates, viewing logs from individual task executions, and handling execution failures through retry mechanisms or manual intervention. The interface translates user-defined pipeline specifications into Apache Airflow DAG definitions dynamically at submission time, generating Python code that instantiates the appropriate task operators, configures dependencies, and sets execution parameters.

The component maintains a library of reusable pipeline templates representing common data processing patterns in 6G research such as time-series aggregation workflows for network performance metrics, data cleaning sequences for experimental measurements, and multi-source data fusion pipelines combining datasets from different testbeds. Users can instantiate these templates, customize parameters, and save modified versions as new templates for future reuse.

Integration with the 6G Data Space enables the interface to present available Data Processing Services dynamically, retrieving service metadata including input/output specifications, processing capabilities, and performance characteristics directly from the Data Space catalogue. This integration ensures that only compatible services are offered when users construct pipelines, preventing configuration errors where task outputs don't match subsequent task inputs.

4.1.2.3 DataOps APIs

The DataOps APIs provide a comprehensive REST-based interface layer enabling programmatic interaction with all DataOps capabilities within the 6G-DALI framework. These APIs serve as the integration point connecting external 6G-DALI components, automated workflows, and third-party applications with DataOps services for data processing pipeline management, Data Processing Service registration, and pipeline execution orchestration.

The DataOps APIs implement a unified RESTful architecture following standard HTTP conventions for resource addressing, method semantics, and status code usage. All API endpoints require OAuth 2.0 authentication through bearer tokens issued by the project's Keycloak identity provider, with authorization scopes controlling access to specific operations. Request and response payloads utilize JSON format for structured data exchange, with consistent error response schemas providing detailed diagnostic information when operations fail.

The API architecture separates concerns across two primary functional domains corresponding to the major DataOps capabilities: the **Data Operation Management API** handles registration, versioning, and lifecycle management of Data Processing Services that constitute reusable pipeline building blocks, while the **Pipeline Management API** enables submission, execution control, and monitoring of data transformation workflows orchestrated through Apache Airflow. This separation ensures that service

developers can manage processing capabilities independently from experimenters who compose and execute pipelines consuming those services.

4.1.3 Intent-based Data Search and Generation

The 6G-DALI Intent-based Data Search and Generation component enable users to obtain datasets that match high-level experimental objectives expressed in natural language. It supports two complementary operational modes:

1. **Intent-driven dataset discovery (Data Search block):** The system interprets user intent and searches the 6G-DALI data space for existing datasets that satisfy the requested characteristics.
2. **Intent-driven data generation through experimentation (Data Generation block):** When no suitable dataset is available, the system initiates an experiment execution workflow to generate the required data on appropriate testbed infrastructures.

Through these capabilities, the component provides an end-to-end, intent-driven lifecycle for data search and generation. This lifecycle encompasses the interpretation of user data requirements, the discovery of relevant datasets, the planning and orchestration of experiments when necessary, and the creation and registration of newly generated data to enable future reuse. By abstracting infrastructure complexity and enabling high-level interaction with distributed data and experimentation resources, the component reduces the technical barriers associated with data-driven 6G research. At the same time, it promotes the efficient reuse of existing datasets and supports the systematic production of new experimental data.

Baseline. Existing approaches to intent-driven experimentation, such as those developed in projects like SUNRISE-6G, primarily aim to enable users to express experimental requirements through declarative or natural-language intents that are automatically translated into executable experiment configurations across distributed testbeds. These systems typically incorporate mechanisms for intent interpretation, orchestration of experiment deployment across testbeds, and testbed-specific execution support implemented through coordinated agents or orchestration services. They also provide natural-language interaction interfaces designed to simplify experiment management and user interaction.

Overall, their primary objective is to facilitate remote experimentation by abstracting infrastructure complexity and automating the configuration and execution of experiments across heterogeneous platforms. However, these approaches mainly address experiment lifecycle management, assuming that data must be produced through experimentation, and do not explicitly integrate dataset discovery within a unified intent-driven workflow.

Innovation. The 6G-DALI Intent-based Data Search and Generation components extend existing intent-driven experimentation paradigms by introducing a unified architecture that integrates data discovery and data generation within a single intent-driven process. Rather than treating dataset retrieval and experiment execution as separate operational domains, the component brings them together within a coherent framework that manages both activities in response to user intent.

A key innovation lies in its unified data acquisition strategy. The system prioritizes the reuse of existing datasets by first attempting to satisfy user requirements through available data resources. Experimentation is triggered only when no suitable dataset is found, thereby promoting efficient utilization of existing data while avoiding unnecessary experimental deployment.

The architecture also provides integrated orchestration across both data space and experimentation testbeds. Dataset search and experiment execution are treated as complementary mechanisms operating within the same framework, effectively bridging data management systems and distributed

experimental platforms. This integration enables seamless transitions between data discovery and data generation processes.

In addition, the component implements an automated decision workflow that determines whether data should be retrieved or generated based on intent interpretation and current data availability. By automating this decision process, the system reduces the need for manual intervention in experiment planning and resource selection. The framework further supports the complete data lifecycle. Newly generated datasets are systematically registered within the data space, ensuring their discoverability and enabling future reuse by other users and applications.

Through these capabilities, 6G-DALI moves beyond traditional intent-driven experiment orchestration toward a broader paradigm of intent-driven data provisioning, in which users are provided with the most appropriate data source - whether existing or newly generated - through a single unified interaction model.

4.1.4 DataOps Orchestrator

The DataOps Orchestrator serves as the intelligent middleware layer between user-defined data processing requirements and the Apache Airflow execution infrastructure, managing the **complete lifecycle of DataOps pipelines** within the 6G-DALI framework. This component bridges the gap between the high-level pipeline specifications created through the DataOps User Interface and the low-level execution details required by Apache Airflow, handling dynamic DAG generation, Data Space integration, resource coordination, and execution monitoring.

The orchestrator's primary responsibility is dynamic DAG generation, translating abstract pipeline specifications into concrete Apache Airflow DAG definitions. When users submit pipeline configurations through the DataOps User Interface, the orchestrator receives structured pipeline descriptions specifying task sequences, data dependencies, and processing parameters. It transforms these specifications into valid Python code defining Airflow DAG structures, instantiating appropriate task operators for each Data Processing Service, configuring inter-task dependencies to enforce correct execution ordering, and setting execution parameters including retry policies, timeout values, and resource requirements. This generated DAG code is then registered with the Apache Airflow scheduler for execution.

Data Space integration constitutes a critical orchestration function, enabling seamless interaction between DataOps pipelines and the 6G Data Space catalogue. The orchestrator resolves dataset references in pipeline specifications by querying the Data Space discovery API, translating metadata-based dataset selections into concrete storage locations and access credentials. When pipelines specify inputs, the orchestrator executes Data Space search queries, retrieves matching dataset metadata, extracts Data Lake storage URLs and access tokens, and injects these concrete references into the generated DAG tasks. Similarly, when pipelines complete and produce output datasets, the orchestrator automatically registers new dataset metadata in the Data Space, creating DCAT-AP compliant records that capture provenance information linking outputs to inputs and transformation processes.

Data Processing Service resolution enables the orchestrator to translate service references in pipeline specifications into executable Airflow task operators. When users compose pipelines from Data Processing Services catalogued in the Data Space, the orchestrator retrieves service metadata including implementation details, container images or execution environments, input/output schemas, and resource requirements. It maps these service specifications to appropriate Airflow operators such as KubernetesPodOperator for containerized services, PythonOperator for Python-based transformations, or custom operators for specialized processing capabilities. This mapping abstraction ensures that experimenters can focus on processing logic without needing detailed knowledge of Airflow's operator ecosystem.

Execution coordination manages the submission and monitoring of generated DAGs within the Apache Airflow cluster. The orchestrator interfaces with Airflow's REST API to trigger DAG runs, monitor

execution progress through task state transitions, collect logs from individual task executions, and handle execution events such as task failures, retries, and completions. When tasks fail, the orchestrator evaluates failure conditions and applies configured retry strategies or notifies users for manual intervention. Upon successful pipeline completion, the orchestrator executes post-processing workflows including output dataset registration, quality score computation triggering, and provenance metadata finalization.

Resource management capabilities enable the orchestrator to optimize pipeline execution across distributed infrastructure. Based on Data Processing Service resource requirements and available cluster capacity, the orchestrator schedules tasks to appropriate worker nodes, manages data locality to minimize transfer overhead by scheduling tasks near their input data, and implements execution policies that balance throughput with resource utilization. For pipelines processing large datasets across multiple testbeds, the orchestrator can implement federated execution strategies where tasks run on testbed-local Airflow workers to avoid transferring raw data across network boundaries.

Provenance tracking throughout pipeline execution captures comprehensive lineage information using PROV-O vocabulary. The orchestrator records which datasets served as inputs, which Data Processing Services performed transformations with what parameter configurations, which intermediate datasets were generated during multi-stage pipelines, and which final output datasets resulted from execution. This provenance graph is registered in the Data Space alongside output datasets, enabling reproducibility verification, quality impact analysis, and data lineage visualization showing complete derivation chains from raw experimental data to processed analytical datasets.

Baseline: The DataOps Orchestrator builds upon Apache Airflow's core orchestration capabilities including DAG-based workflow specification, distributed task execution across worker pools, dependency management ensuring correct task ordering, monitoring and logging infrastructure, and REST API for programmatic interaction. Apache Airflow provides the foundational workflow engine with proven scalability and reliability in data engineering applications. The baseline also incorporates standard Airflow patterns for dynamic DAG generation using DAG factories and programmatic DAG construction rather than static DAG files.

Innovation: The 6G-DALI implementation extends this baseline through several new functionalities specifically addressing the requirements of federated 6G research infrastructure and experiment-driven data processing workflows:

- **Data Space-Native Orchestration:** Unlike traditional Airflow deployments where data sources and processing logic are tightly coupled through hard-coded file paths and database connections, the 6G-DALI orchestrator implements Data space-native orchestration where all data references are resolved through metadata queries. This abstraction means experimenters specify what data they need rather than where it resides, with the orchestrator handling storage location resolution, access credential management, and data transfer orchestration transparently. When testbed storage infrastructure changes or datasets migrate between storage tiers, pipelines continue functioning without modification because data access is mediated through stable Data Space metadata rather than brittle storage paths.
- **Service-Oriented Task Composition:** The orchestrator implements a service-oriented architecture where Data Processing Services catalogued in the Data Space serve as first-class pipeline building blocks rather than requiring custom task operator development for each processing capability. When new data processing algorithms are developed, they are packaged as services with standardized metadata describing inputs, outputs, and execution requirements, then registered in the Data Space. The orchestrator automatically discovers these services and generates appropriate Airflow task operators dynamically based on service metadata, eliminating the traditional Airflow requirement for pre-installed custom operators. This innovation dramatically accelerates the integration of new processing capabilities into available pipeline tooling.

4.1.5 Experimentation On Demand

Specifically, for DataOps, the Experimentation on Demand (EoD) component provides the functionality required to identify gaps in existing datasets that prevent the effective training or evaluation of ML models. It provides capabilities to perform structured assessments of the available data assets, determine whether key variables, operating conditions, or coverage regions are missing, and flag cases where the existing datasets are insufficiently representative. To support this process, the component maintains a partial characterization of the dataset-generation space, describing which parts of the domain have been explored, which measurement conditions have been covered, and where uncertainties or sparsity remain. This enables systematic determination of what additional data must be generated to support model development.

Innovation: The main innovation in the DataOps space is the ability to generate large datasets with the use of surrogate models. During an experimental campaign, users define the characteristics of the dataset to be generated. Normally, this would constitute a configuration space of experiments to be executed on the testbeds. EoD instead, strategically samples only a subset of the overall experimental space on the actual 6G testbeds. This subset is then used to train surrogate models capable of approximating the underlying system behaviour. Once a user-agreed confidence threshold is reached, these surrogate models can accurately generate the remaining part of the experimental space that reflect the expected behaviour of the real testbed. This enables the training of AI models without requiring exhaustive measurements from physical experiments.

This approach dramatically reduces the time and resources needed to obtain comprehensive experimental data.

4.1.6 DataOps Tools Stack

The DataOps tools stack is designed to offer a range of novel algorithms and developed tools that support data quality improvement and augmentation. These algorithms or tools are packaged as microservices, APIs, or interfaces that users can call and integrate into an ELT pipeline, serving as building blocks to facilitate users in cleaning, transforming, and augmenting the raw data.

According to the 6G-DALI DataOps technical usage scenarios, the tools stack shall address several data quality issues, such as outliers, missingness, uniqueness, handle format change, dataset reduction and aggregation, as well as time-series imputation, and/or data augmentation. Before performing data cleaning, an important step is to identify data quality issues. These tools work under each identical quality issue, and the transformed or augmented datasets can be validated by the governance and data quality module of the 6G data space.

Baseline. For the data quality and validation module, we consider the open-source framework called Great Expectations (GX)²⁰. GX enables systematic data quality checks to assess and improve data reliability across diverse data assets and sources, including SQL tables in databases, file-based datasets stored in object stores, and in-memory DataFrames. This flexibility is essential in distributed 6G research environments, where heterogeneity is inherent across data formats, storage backends, and processing stacks. GX also supports built-in methods for (time-series) data distribution analysis, handles data freshness, integrity, missingness, uniqueness in data quality management, and validates data schemas and unstructured data. For data augmentation, we consider the join-path-based data augmentation methods for cleaned (tabular) datasets, like AutoFeat²¹. Given a base table with a target variable and a repository of tabular datasets, AutoFeat helps to discover relevant features (such as column names) for augmentation among the tables from the data repository. Besides, for time-series data, we consider diffusion models to generate missing data; such imputed series can enrich training instances, which can also be useful for the time-series data augmentation task.

²⁰ https://github.com/great-expectations/great_expectations

²¹ <https://github.com/delftdata/autofeat>

Innovation and extension. The 6G-DALI implementation extends the existing tools by integrating novel AI-based time-series generation algorithms or methods for tabular data imputation into the DataOps tools stack, tailored to address missing-data challenges in 6G time-series and tabular data within DataOps workflows or ELT pipelines. The 6G-DALI implementation also extends augmentation algorithms tailored to data augmentation of 6G DataOps workflows.

4.1.7 Interfaces

The DataOps component exposes a set of interaction interfaces that enable human users and external system components to access data processing and orchestration capabilities. The detailed functionality of each interface is described in Section 4.1.2, while the following summarizes the exposed interaction points and their integration role within the 6G-DALI architecture.

- **DataOps Dashboard (WEB)** is a browser-based interface providing visual pipeline design, configuration, and execution capabilities. It exposes the functionality described in Section 4.1.2.2, including DAG composition, dataset selection via Data Space integration, parameter configuration, template management, and execution monitoring. Authentication is handled through OAuth 2.0 / OIDC. This interface serves as the primary human-access layer for data transformation workflow orchestration.
- **Chatbot User Interface (WEB)** is a conversational interface providing natural-language access to data discovery and experiment-driven dataset generation capabilities, as detailed in Section 4.1.2.1. It exposes dataset search, experiment request formulation, and user guidance functionalities through a dialogue-based interaction model. This interface targets accessibility and lowers the barrier to entry for non-technical users.
- **Data Operation Management API (REST)** is a REST-based interface for registering, updating, versioning, and managing Data Processing Services. It enables service developers and external components to programmatically contribute new processing capabilities to the DataOps ecosystem. Registered services are catalogued in the 6G Data Space and become available for pipeline composition through the DataOps User Interface.
- **Pipeline Management API (REST)** is a programmatic interface enabling external systems to submit, monitor, and control pipeline executions without using the graphical interface. It supports structured pipeline specification submission, execution status retrieval, log access, and lifecycle management. Authentication and authorization are enforced via OAuth 2.0 access tokens with scoped permissions.

4.2 MLOps

The MLOps block/component in 6G-DALI is designed to provide comprehensive lifecycle management for AI/ML functionalities, serving as a centralized point for hosting and managing all AI/ML intelligence developed in 6G-DALI. It unifies essential AI/ML capabilities, including model cataloguing and versioning, model training, model deployment, and continuous performance monitoring. Crucially, the 6G-DALI MLOps will extend these baseline functionalities with the support for advanced paradigms such as Hyperparameter Optimization (HPO), Federated Learning (FL), Reinforcement Learning (RL), Transfer Learning (TL), and Trustworthy AI (TAI), operating in a dual modality: centralized and distributed across the 6G-DALI testbeds. In the centralized approach, the 6G-DALI facility will support AI/ML workloads directly deployed and managed in a centralized environment, while, in the decentralized approach, AI/ML workloads will be deployed and managed in a fully distributed manner via meta-orchestration, coordinating and enhancing independent MLOps stacks running across the heterogeneous 6G-DALI testbeds.

4.2.1 Functional Architecture

The MLOps functional architecture, depicted in Figure 9, is composed of two different layers, that operate across MLOps Tools Stacks deployed in both a centralized facility and in distributed testbeds.

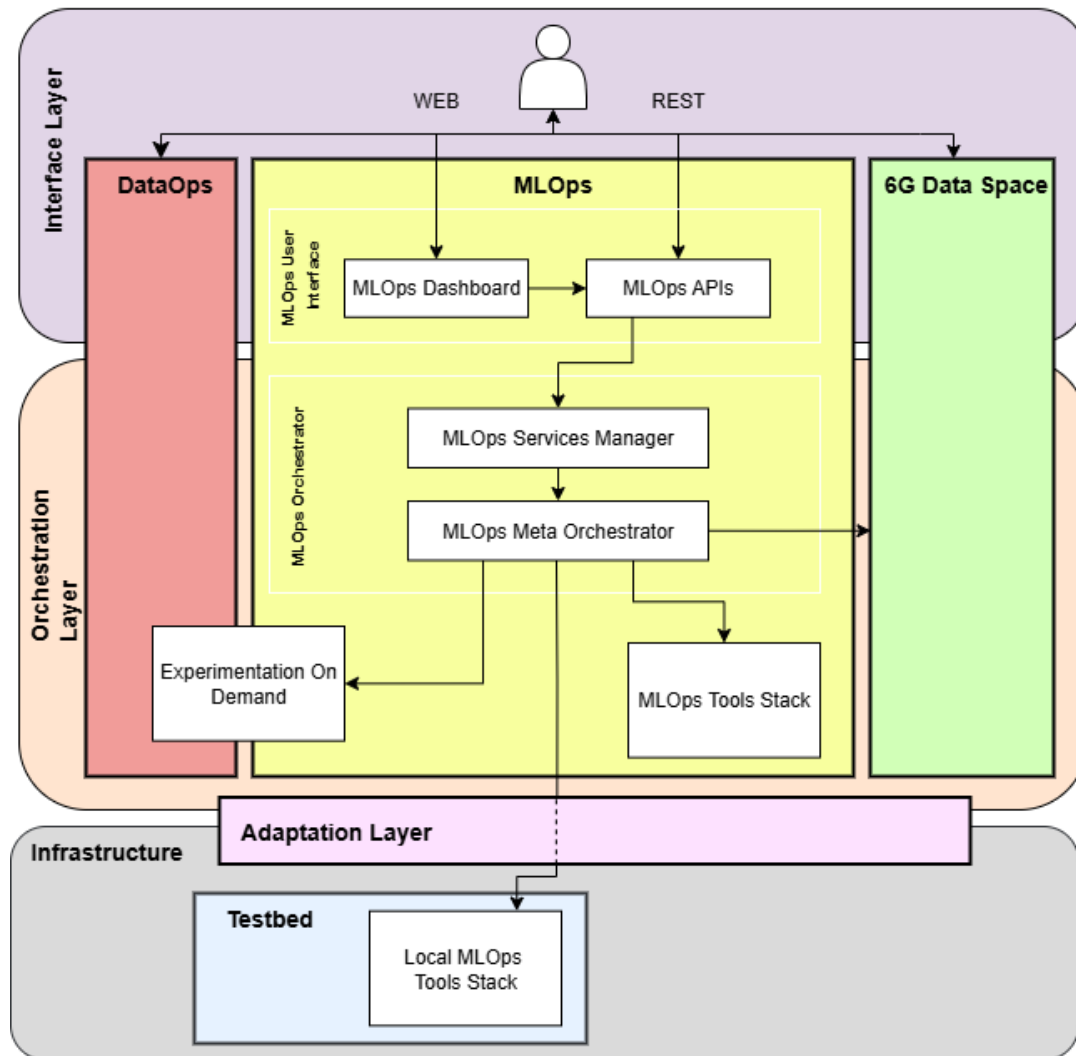


Figure 9: MLOps Functional Blocks

These layers are defined as follows:

- **MLOps User Interface:** provides a human-centric interface and programmatic API designed for experimenters, enabling them to interact with the 6G-DALI MLOps stack.
- **MLOps Orchestrator:** manages the system logic through an abstract framework that handles high-level concepts and entities which are then realized and implemented by the underlying centralized and testbed's tools stacks.

Sections 4.2.2 and 4.2.3 respectively report about the two layers, including their baseline solutions and core functionalities. Section 4.2.5 and its subsections detail the specific tools envisioned for the MLOps Centralized Tools Stack and the Testbed Tools Stacks.

4.2.2 MLOps User Interface

The MLOps User Interface serves as the primary gateway for interacting with the 6G-DALI MLOps workflows. It provides a unified, high-level abstraction that decouples user objectives from the complexity of the underlying infrastructure and execution environments. To accommodate both human operators and automated systems, this interface is structured into two complementary layers: a graphical MLOps Dashboard, described in Section 4.2.2.1, and programmatic MLOps APIs, described in Section 4.2.2.2.

4.2.2.1 MLOps Dashboard

The MLOps Dashboard provides a human-centric, graphical interface for experimenters to manage AI/ML workflows in a technology-agnostic manner. Architecturally, the Dashboard acts as a frontend

client that operates by consuming the underlying MLOps APIs to execute user commands. From a functional perspective, the dashboard exposes visual access to core MLOps concepts such as workflow and pipeline definitions, executions, artifacts, and metadata. It facilitates reproducibility and lifecycle transparency by allowing users to easily navigate experiment tracking and model versioning, and the inspection of execution history and lineage.

4.2.2.2 MLOps APIs

The MLOps APIs layer, depicted in Figure 10, exposes the AlaaS functionalities envisioned in 6G-DALI. This layer uses the 3GPP AIMLE APIs (described in [5] as baseline, extending them to support the set of MLOps operations developed for the 6G-DALI Framework. These APIs are primarily envisioned as RESTful interfaces. Furthermore, this layer will provide a Model Context Protocol (MCP) Server to facilitate interaction external for autonomous agents. While the implementation of agents interacting with the MLOps MCP server is not part of the 6G-DALI work on MLOps, exposing this endpoint ensures that the MLOps AlaaS is aligned with current trends and is future-proof.

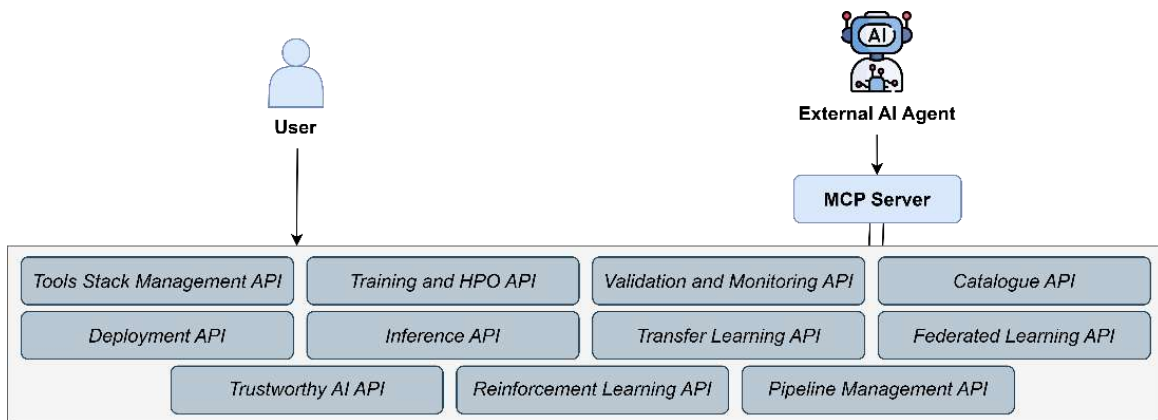


Figure 10: MLOps Northbound Interface

Table 4 provides a categorization of these APIs. It is important to emphasise the crucial role of the Tools Stack Management API in the meta-orchestration of MLOps tasks provided by 6G-DALI MLOps. This API enables the registration, updating or removal of tool stacks, providing the necessary capabilities to execute MLOps tasks. The other APIs enable specific MLOps functionalities that can be performed in 6G-DALI, either in the centralised Tools Stack or in the Tools Stack provided by the testbeds.

Table 4: MLOps API Categories

API Category	High-level Description
Tools Stack Management API	Enables the CRUD operations for MLOps stack, allowing the registration/de-registration of new tools stacks and the modification of their existing capabilities.
Training and HPO API	Enables the CRUD operations for training and HPO processes, allowing the selection of models, datasets (previously selected from the DALI data space), and specific training parameters.
Validation and monitoring API	Enables the CRUD operations for one shot validation and continuous verification of trained models.
Catalogue API	Enables the management of the lifecycle and versioning of available AI/ML models within the DALI ecosystem.

Deployment API	Enables the automated deployment of trained models onto target execution environments and the management of their operational status.
Inference API	Enables the execution of inference requests over deployed models, allowing for real-time or batch data processing and prediction endpoint retrieval.
Transfer Learning API	Enables the processes where knowledge is extracted from existing models and ensembled or fine-tuned into new models.
Federated Learning API	Enables the deployment and management of federated clusters, as well as the execution and coordination of distributed training tasks.
Trustworthy AI API	Enables the endpoints for assessing and ensuring model robustness, explainability, and ethical compliance.
Reinforcement Learning API	Enables the management of RL agents, environments, and reward function configurations.
Pipeline Management API	Enables the CRUD operations for AI/ML Pipeline composed of several MLOps tasks.

4.2.3 MLOps Orchestrator

The MLOps Orchestrator layer (Figure 11) is built upon the information model defined in [6], which is structured around three primary entities: Models, Processes (involving those models), and Reports (generated as outcomes of the processes). By adopting this model, the MLOps Orchestrator layer operates at a level of abstraction above the individual tools. The DALI MLOps Orchestrator layer utilizes the ADROIT6G AI/ML framework for crowdsourcing AI as its baseline, extending it by integrating the 3GPP-based information model and moving away from direct tool interaction toward a concept-based orchestration. To enable the MLOps meta-orchestration functionalities, this layer describes MLOps operations and tool capabilities as generic processes, allowing experimenters to request tasks and access reports in a completely tool-agnostic manner. The three primary functional components of this layer are:

- **MLOps Services Manager:** the role of this component is to manage at an abstract level the MLOps processes. It hosts various Management components (reported in 4.2.5) each containing the logic and data model required to manage its respective MLOps processes (e.g. Training, Validation, HPO).
- **MLOps Meta-Orchestrator:** this component receives abstract MLOps tasks description (defined in the mlops northbound interface) from the MLOps Services Manager and its role is to analyse the requirements and identify the most suitable Tools Stack (either centralized or in a testbed), and coordinate with them to implement and execute the task.

The decoupling between the MLOps Services Manager and the MLOps Meta-Orchestrator is the fundamental enabler of the tools-agnostic 6G-DALI meta-orchestration. It maintains an abstract information model at the upper layer while utilizing a meta-orchestrator capable of translating these concepts into actual deployments across the tool stacks.

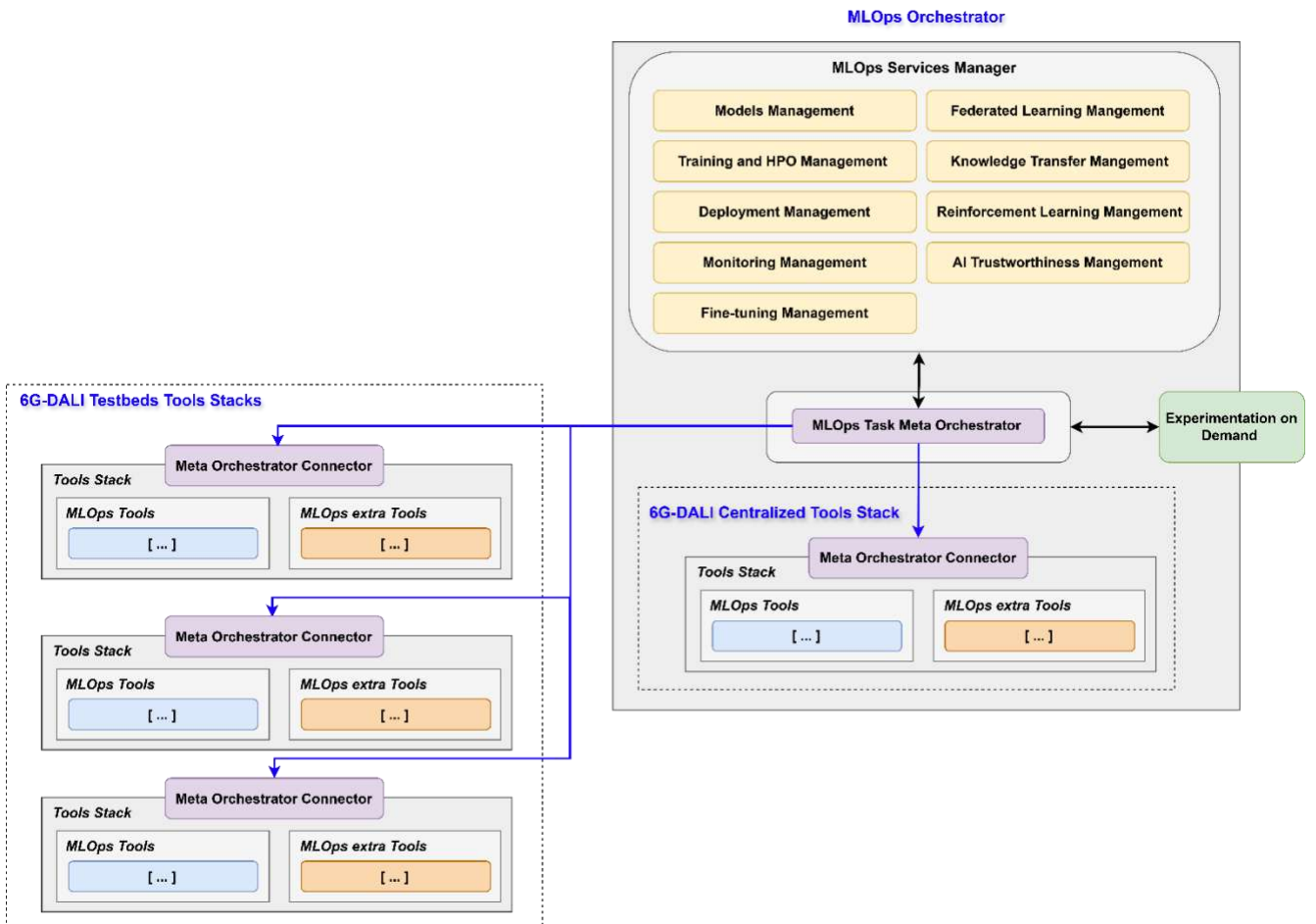


Figure 11: MLOps Management and Orchestration Layer

Table 5: 6G-DALI MLOps Service Management Functionalities

Service Management Functionalities	High-level Description
Models Management	Responsible for the lifecycle and versioning of the <i>MLModel</i> entities and their associated metadata.
Training and HPO Management	Responsible for the logic of <i>MLTrainingProcesses</i> and <i>HyperparameterOptimizationProcesses</i> .
Deployment Management	Responsible for the transition of models into an operational state via <i>MLModelDeploymentProcesses</i> .
Monitoring Management	Responsible for the oversight of model performance through <i>MLModelMonitoringProcesses</i> and the generation of performance Reports.
Fine-tuning Management	Responsible for the logic of <i>MLModelFineTuningProcesses</i> to adapt existing models to new data or domains.
Federated Learning Management	Responsible for the coordination of <i>FederatedLearningProcesses</i> across distributed participants.
Knowledge Transfer Management	Responsible for <i>KnowledgeTransferProcesses</i> involving the extraction and reuse of insights from source to target models.

Reinforcement Learning Management	Responsible for the logic of <i>RLTrainingProcesses</i> involving agents, environments, and reward feedback loops.
AI Trustworthiness Management	Responsible for <i>TrustworthinessEvaluationProcesses</i> and the generation of associated compliance Reports.

4.2.4 Experimentation on Demand

This component provides automated execution support for compute-intensive MLOps tasks, most notably hyperparameter optimisation, by executing tasks "translated" by the meta-orchestrator for a specific tool stack. Its optimisation engine evaluates the hyperparameters search space, selects algorithms suited to the model and dataset characteristics, and orchestrates parallel or sequential trials to efficiently explore parameter configurations. Throughout execution, it tracks intermediate states, reuses cached computations where possible, and exposes structured outputs back to the MLOps Services Manager for reporting, model registration, and downstream lifecycle actions.

Innovation. EoD provides a significant innovation in how experimental campaigns are executed and optimised, particularly given the current limitations of 6G testbeds, which remain few in number and for which dataset generation through physical experimentation is both time-consuming and resource-intensive. EoD enables experimenters not only to run experiments but also to apply optimisation algorithms that automatically explore the configuration space and identify system setups that best satisfy user-defined objectives, such as reducing energy consumption, increasing throughput, or supporting a greater number of concurrent users. By intelligently navigating the experimental space, EoD helps users converge quickly on configurations that match their operational goals.

Also, by leveraging the data generation via surrogate models described in Section 4.1.5, EoD supports experimenters in analysing and understanding how the system behaves in regions that have not been physically tested.

4.2.5 MLOps Tools Stack

As can be noted from Figure 11, both the centralized and the testbeds tools stacks envisioned in the 6G-DALI ecosystem can host a variety of MLOps and MLOps-extra tools. The scope of this section is to provide insights into the MLOps-extra tools developed in 6G-DALI in the context of MLOps, capable of offering MLOps-extra functionalities that build on top of existing MLOps functionalities provided by state-of-the-art MLOps Tools. The following sections report about the baseline for these tools and which are the extensions envisioned in 6G-DALI to go beyond the actual state-of-the-art.

4.2.5.1 Federated Learning Tools

The Federated Learning (FL) tools module is designed to provide a range of novel FL algorithms, consisting of global models and federated learning aggregation strategies to contribute to innovation area 7. It targets improvements in privacy-preserving efficiency and reduction in communication overhead by leveraging MLOps and NWDAF operations for 6G privacy and efficiency.

Baseline. Since these privacy-preserving FL models or tools are dedicated to 6G time-series datasets, we consider forecasting and classification tasks. Besides, to fortify defences against adversarial attacks within a distributed learning environment, secure aggregation strategies are considered. FLAME [11] is a resilient aggregation framework for FL to defend against backdoor attacks by filtering out malicious model updates without sacrificing the accuracy of the global model. However, the raw model updates from clients to the server are not hidden. To achieve EFFE and COR, several encoding technologies, such as random projection, noise-shaping, and efficient communication protocols, such as quantization, sparsification, and model compression, are considered. Besides, to reduce communication overhead, communication-efficient federated distillation approaches, for example FedBoost [12] are also considered. The novel FL algorithms will build atop robust baselines to improve privacy-

preserving efficiency while reducing communication overhead, tailored to 6G-DALI time-series datasets.

Innovation. Complementing the algorithmic innovations, the 6G-DALI FL Tools will specifically address the current operational gaps in Federated Learning Operations (FLOps) by providing a Federations Management Framework. While state-of-the-art FL Frameworks as Flower²² and OpenFL²³ provide robust environments for the management of FL training, there remains a significant challenge in the management and orchestration of FL Nodes lifecycle and the dynamic management of Federations. To fill this gap, 6G-DALI introduces a FLOps Management Tool for the automated deployment and LCM of FL nodes. This tool acts as an automation layer that enables the registration, health monitoring, and decommissioning of FL nodes across distributed testbeds. By integrating this management capability, the framework enables full FLOps automation. This ensures that the advanced algorithms previously described can be deployed at scale with the same level of operational agility found in MLOps pipelines.

4.2.5.2 Knowledge Transfer Tools

The **Knowledge Transfer Tool** is a centralized MLOps service that builds upon the **Cooperative Transfer Learning (CoTL)** paradigm introduced in [13], which demonstrated that independently trained ML models can selectively exchange algorithm-specific knowledge to improve class-level performance without requiring centralized data sharing or full model aggregation. In the **baseline** work, CoTL is realized through decentralized, pairwise cooperation between models, where knowledge transfer decisions are taken locally based on observed performance characteristics, and the approach is primarily illustrated in algorithmic level.

Baseline. In 6G-DALI, we **extend this baseline** by transforming CoTL from a conceptual and algorithmic-centric mechanism into a fully managed, centralized MLOps service that enables scalable and systematic cooperative learning across distributed 6G experimenters and testbeds. The proposed Knowledge Transfer Tool operationalizes CoTL by acting as a coordination and decision-making engine that governs how and between which models' knowledge is exchanged.

Innovation. Unlike the baseline approach, where cooperation emerges from direct interactions between models, the Knowledge Transfer Tool introduces a **centralized knowledge management layer**. Experimenters contribute model-level and data-level metadata, including model type, performance metrics, data characteristics, and class-level errors, which are used by the tool to identify *logical neighbours*, i.e., models trained on similar data distributions or exhibiting complementary performance profiles. This metadata-driven neighbour discovery enables multi-directional and many-to-many cooperation, going beyond the pairwise interactions considered in the baseline.

A key extension over the original CoTL formulation is the introduction of a **Knowledge Catalogue**, which persistently stores reusable knowledge units extracted from participating models. Knowledge extraction follows algorithm-specific strategies, such as deriving decision structures from ensemble-based models (e.g., decision trees from Random Forests). As new models and updates are contributed, the catalogue is continuously enriched, allowing knowledge to be reused across time and across multiple experimenters rather than exchanged only transiently. During the knowledge transfer phase, the tool performs **selective and targeted knowledge propagation**. Model weaknesses, such as poorly performing classes, are matched against relevant knowledge units originating from stronger models. For example, in the case of Random Forests, high-quality decision trees extracted from neighbouring models can selectively replace weaker components in a target model, following a partition-and-merge strategy similar in spirit to the baseline CoTL approach, but applied under centralized control and with global visibility of available knowledge.

Overall, the Knowledge Transfer Tool extends the original CoTL concept by providing a scalable, reusable, and MLOps-integrated realization suitable for operational 6G environments. By embedding

²² <https://flower.ai/>

²³ <https://github.com/securefederatedai/openfederatedlearning>

CoTL within the 6G-DALI MLOps framework, alongside model training, validation, deployment, and monitoring, the tool allows experimenters to benefit from CoTL through high-level interactions, without direct exposure to the underlying algorithmic mechanisms.

4.2.5.3 Reinforcement Learning Tools

The proposed RLOps framework establishes a comprehensive operational paradigm for deploying RL agents in collaboration with DTs, addressing the gap between RL research and operational deployment in Open-RAN systems. The framework defines a standardized lifecycle encompassing Design, Development, and Operation phases, enabling systematic validation, deployment, and lifecycle management of RL-driven control applications. By structuring RL workflows around operational constraints and O-RAN compliance, the framework advances beyond ad hoc experimentation toward reproducible, production-ready intelligence for next-generation radio access networks.

Innovation. This work extends beyond the current state of the art by delivering an end-to-end integration of widely adopted MLOps and RLOps platforms within a unified and validated workflow. Consistent with Deliverable D2.2, MLflow [14] Ray Tune[15] and RLlib [16] constitute the technological basis of the framework. These technologies are considered from an architectural and workflow integration perspective. Their combined use enables pipeline orchestration, experiment tracking, distributed training, and policy lifecycle management within a coherent RLOps environment.

A key architectural contribution lies in the tight coupling between the RLOps framework and DT platforms. The novelty of the proposed framework resides in the interoperability, lifecycle alignment, and operationalization of RL components. By transforming standalone RL tooling into an integrated system, the framework supports continuous integration, evaluation, and controlled deployment of RL policies in DT-driven Open-RAN scenarios. This integration enables safe training, validation, and performance assessment of RL agents in high-fidelity DT environments prior to live deployment, while supporting scalable and collaborative RL workflows across distributed DT instances.

4.2.5.4 AI Trustworthiness Tools

The 6G-DALI Trustworthy AI Tools are designed as modular, model-agnostic tools integrated into the MLOps and RLOps pipelines to ensure that AI models trained and tested through the 6G-DALI facility are reliable, transparent, and robust.

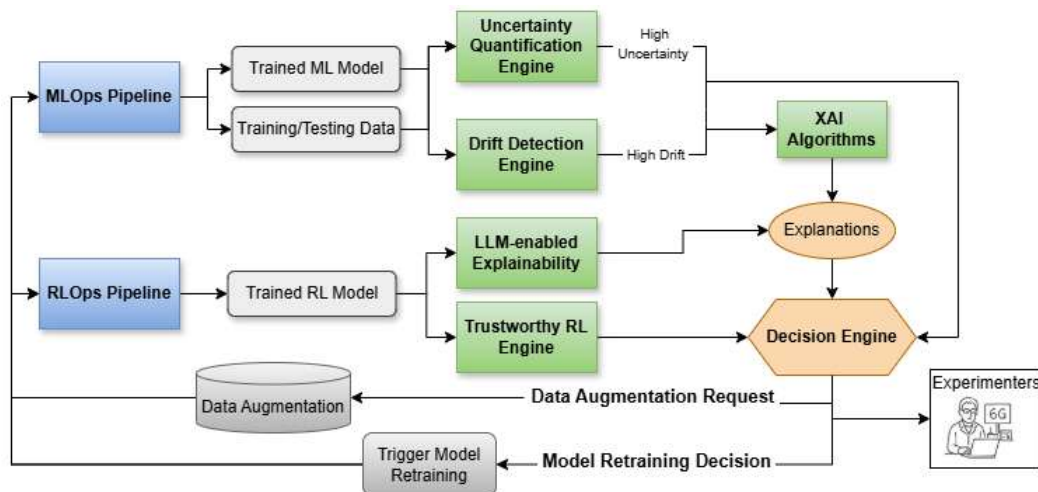


Figure 12: High-level overview of the Trustworthy AI tools across the MLOps and RLOps frameworks

Innovation. They comprise several key components that collectively provide beyond-state-of-the-art functionality to handle the challenges of training trustworthy AI models for 6G networks and datasets. A high-level overview of the trustworthy AI tools is depicted in Figure 12 structured around specific modules for MLOps, like the Uncertainty Quantification Engine and Drift Detection Engine, and specialized modules for RLOps, like the Trustworthy RL engine. From a functional perspective, these components operate within the 6G-DALI MLOps experimentation framework to provide trustworthy

outputs and metrics throughout the AI lifecycle, from training to runtime monitoring. Each tool will be exposed through APIs or as microservices for real-world integration. This subsection provides a high-level functional description of each solution, while the technical, algorithmic, and implementation details will be provided in future deliverables.

The functional description of each tool is as follows:

- **Uncertainty Quantification Engine:** This tool provides metrics of aleatoric and epistemic uncertainty to ensure statistically reliable predictions at inference time. Utilizing algorithms like Conformal Prediction [17] and Probabilistic Inference, it enables the MLOps pipeline to identify when a model's output should be trusted.
- **Drift Detection Engine:** This module monitors data streams in real-time, leveraging Autoencoders and Dual Self-attention techniques to detect performance degradation caused by data or label drifts in dynamic 6G environments.
- **Explainable AI (XAI) Algorithms:** This component employs feature importance explainability algorithms, like SHAP [18], to provide interpretability by identifying which input features or KPIs most influenced a model's decision, particularly during high-uncertainty or high-drift events.
- **LLM-enabled Explainability:** This tool [19] leverages Reward Decomposition and Large Language Models (LLMs) to transform Deep Reinforcement Learning (DRL) agent traces into natural language explanations for the agent's actions. Explanations are provided through a chatbot interface, fostering experimenter trust in autonomous network management decisions.
- **Trustworthy RL Engine:** This tool ensures secure and resilient autonomous operations by keeping RL agents within safe operational bounds through mechanisms like safety shields and fallback policies. It outputs a unified Trust Score that combines various uncertainty metrics to provide a single, interpretable reliability measure for RLOps models.

Finally, the unified Decision Engine is not an algorithmic tool but serves as the functional coordination block of the pipeline that can trigger automated actions based on feedback from the tools. Thus, it can request actions like data augmentation, model retraining, or policy fallbacks to ensure model robustness is maintained through continuous, closed-loop monitoring and mitigation strategies.

4.2.6 Interfaces

The 6G-DALI MLOps interacts with other components via two primary categories of interfaces, described in Table 6.

Table 6: MLOps Interfaces

Interface	Role	High-level Description
AlaaS Interface	Exposed	Enables external experimenters and vertical applications to interact with the MLOps system. It exposes 3GPP-compliant endpoints for requesting MLProcesses, managing MLModels, and retrieving MLReports without requiring knowledge of the underlying infrastructure and tools
Tools Stack Interface	Consumed	Enables the MLOps Meta-Orchestrator to attach to various tool stacks (centralized or testbed-based). Through this interface, the system receives information about tool availability and status and subsequently injects the translated MLOps tasks for execution.

4.3 6G Data Space

The 6G Data Space serves as the foundational data sharing and governance infrastructure within the 6G-DALI framework, enabling secure, sovereign, and interoperable access to datasets and AI/ML models

generated across the project's testbed ecosystem. Drawing upon principles established by Gaia-X and the International Data Spaces Association (IDSA), the 6G Data Space implements a federated architecture that maintains data sovereignty while facilitating collaborative research and development activities.

The primary role of the 6G Data Space within 6G-DALI is to bridge the gap between distributed data generation (occurring across heterogeneous testbeds) and centralized data consumption (by MLOps pipelines, DataOps workflows, and external researchers). By implementing standardized metadata schemas based on **DCAT-AP**²⁴ (Data Catalogue Vocabulary - Application Profile), a European metadata standard for describing public sector datasets, the Data Space ensures that datasets are discoverable, accessible, and reusable according to FAIR (Findable, Accessible, Interoperable, Reusable) principles. This capability is particularly critical in the 6G context, where diverse data sources must be catalogued and made available for AI/ML experimentation.

The 6G Data Space architecture addresses several key challenges inherent to distributed 6G research environments: heterogeneous metadata formats across testbeds, the need for fine-grained access control and usage policies, semantic interoperability between different data domains (network, application, infrastructure), and compliance with European data protection and sovereignty requirements.

4.3.1 Functional Architecture

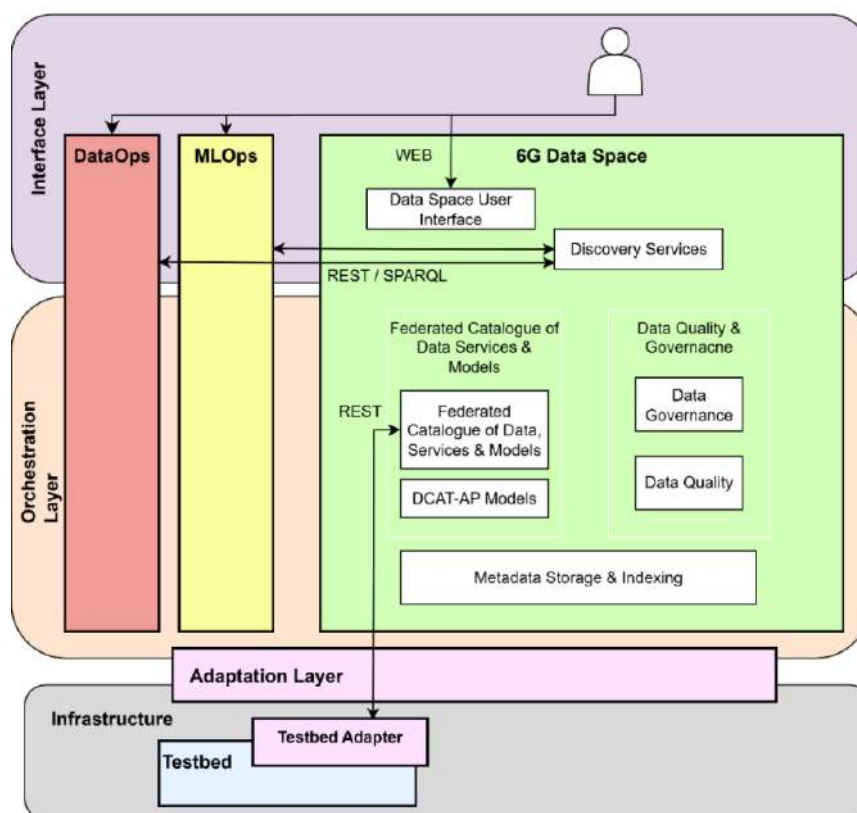


Figure 13: 6G-DALI Data Space architecture

The functional architecture of the 6G Data Space implements a layered design organized around six primary functional domains that work together to enable secure, standards-compliant dataset discovery and access across the distributed 6G-DALI testbed infrastructure. Figure 13 illustrates these functional building blocks and their interactions within the Data Space boundary.

²⁴ <https://semiceu.github.io/DCAT-AP/releases/3.0.0/>

- **Data Space User Interface** serves as a web-based portal for human interaction, providing dataset discovery, metadata viewing, and catalogue management capabilities through an intuitive browser-based interface.
- **Data Quality** performs comprehensive quality assessment, evaluating both metadata completeness and data validity, integrating quality scores from external data quality pipelines, and coordinating with governance mechanisms to enforce quality-based access policies.
- **Data Governance** enforces access policies, usage restrictions, and contractual obligations for all programmatic access from external 6G-DALI components. This component mediates dataset discovery and retrieval operations, ensuring data sovereignty principles and testbed-specific governance rules are respected.
- **Federated Catalogue of Data, Services & Models** acts as the central metadata management component, maintaining the DCAT-AP-compliant catalogue structure encompassing datasets, data services, ML models, distributions, and catalogues. This component orchestrates metadata flows from distributed testbed sources and coordinates metadata registration, quality enrichment, and versioning operations.
- **Metadata Storage & Indexing** provides dual-storage persistence infrastructure, maintaining metadata in both RDF triple stores (optimized for semantic queries) and search indices (optimized for performance), with synchronization mechanisms ensuring consistency between representations.
- **Discovery Services** implements intelligent search and recommendation capabilities, enabling users to efficiently locate relevant datasets, services, and models through faceted search, full-text search, semantic similarity recommendations, and SPARQL query interfaces.

4.3.2 Data Space User Interface

The Data Space User Interface provides a web-based portal for interacting with the 6G Data Space, enabling users to discover, access, and manage datasets generated by the 6G-DALI framework. It serves as the primary human interaction point with the data space ecosystem, supporting researchers and data consumers in locating and utilizing datasets produced through testbed experiments and DataOps processing workflows.

The interface offers advanced data discovery and search capabilities through a comprehensive faceted search system. Users can filter datasets by multiple criteria including publisher, format, catalogue, categories, keywords, data scope, country, data services, quality scoring, and license. The faceted search enables users to progressively narrow search results through combinations of metadata attributes, with real-time updates showing the count of available datasets matching selected criteria. This multi-dimensional filtering approach enables users to efficiently locate relevant data assets within potentially large and diverse data space catalogues.

Dataset management is handled through comprehensive detail views that present metadata, distributions, quality metrics, categories, and similar dataset recommendations. Users can download distributions directly, export metadata into multiple linked data formats such as RDF, Turtle, N-Triples, and JSON-LD, and access quality assessment scores that indicate the completeness and reliability of the dataset metadata. Similar datasets feature leverages semantic analysis to recommend related data assets, facilitating broader data exploration.

For authenticated users with appropriate permissions, the interface enables dataset lifecycle management for assets that have been generated by the 6G-DALI framework following experiment execution. These management capabilities include editing dataset metadata (updating descriptions, keywords, thematic classifications), managing dataset visibility (transitioning datasets from draft to published states, controlling access permissions), organizing datasets into catalogues and sub-catalogues, updating distribution information (adding new format variations, updating access URLs), and deprecating or archiving obsolete datasets. The interface implements form-based metadata editing

with validation to ensure metadata quality and compliance with the underlying DCAT-AP data model, preventing invalid metadata from entering the catalogue.

It is important to note that dataset creation is not performed directly through this interface. Rather, datasets are generated automatically when experiments are executed within the 6G-DALI framework—through testbed experiments, DataOps ELT pipeline executions, MLOps training runs, or Digital Twin simulations. The experiment execution infrastructure automatically generates initial metadata for resulting datasets and registers them in the Data space catalogue. The User Interface then enables users to discover these experiment-generated datasets and refine their metadata to improve discoverability and documentation quality.

The interface also provides catalogue browsing functionality with hierarchical navigation supporting sub-catalogue relationships and organizational structures. This is particularly relevant in the 6G-DALI federated scenario where multiple catalogues from different testbeds need to be browsed coherently. Additionally, an integrated SPARQL query interface using YASGUI components enables advanced users to perform semantic queries directly against the data space's RDF knowledge graph, supporting complex data discovery scenarios that go beyond simple faceted search.

Multi-language support is built into the interface, with internationalization covering European languages, ensuring accessibility for users across different EU member states and supporting the European data sovereignty principles that underpin the Gaia-X alignment.

Baseline: The Data Space User Interface builds upon the Piveau Hub UI, an open-source implementation of DCAT-AP-compliant data portal functionality developed within the European data.europa.eu ecosystem. Piveau Hub UI provides foundational capabilities for DCAT-AP metadata browsing, faceted search, RDF/SPARQL access, and multi-format metadata export. The baseline interface implements responsive web design principles, accessibility standards (WCAG 2.1), and multi-language support covering all EU official languages.

Innovation and Extension: The 6G-DALI implementation extends the baseline Piveau Hub UI in several dimensions specifically tailored to 6G research workflows and the project's federated testbed architecture:

- **Experiment-Driven Metadata Management:** Adaptation of the interface workflow to accommodate the 6G-DALI data generation model where datasets originate from framework experiments rather than manual upload. The interface provides specialized views showing experiment provenance (which testbed, DataOps pipeline, or MLOps training run generated each dataset), links to experiment execution logs and configurations, and workflows for metadata enrichment of automatically generated datasets. This distinguishes the 6G-DALI implementation from traditional data portals where users manually create and upload datasets.
- **6G-Specific Metadata Extensions:** The interface incorporates visualization and filtering capabilities for telecommunications-domain metadata specific to 6G research environments. This includes network topology descriptors (cell configurations, frequency bands, MIMO configurations), RAN parameter sets (beamforming schemes, handover policies, scheduling algorithms), edge computing resource characteristics (CPU/GPU availability, latency profiles, geographical distribution), and AI/ML model performance metrics associated with training datasets. These extensions appear as additional facets in the search interface and dedicated metadata sections in dataset detail views.
- **Quality-Aware Discovery:** Dataset quality scores appear as **visual badges** in search results, influence default result ordering (higher quality datasets ranked higher), and are explained through detailed quality reports accessible from dataset pages. These dataset quality indicators are computed by the Data Quality & Governance module and are integrated to the user interface to enhance the user experience by surfacing reliability information at multiple interaction points. Users can filter explicitly by quality thresholds, enabling workflows such as "show only datasets with quality scores above 0.8 suitable for AI/ML training."

- **AI/ML Model-Dataset Linkage Visualization:** Bidirectional relationship **visualization** shows connections between datasets and AI/ML models catalogued in the MLOps component. Dataset pages display lists of models trained using that data, including model performance metrics and versioning information. Conversely, when viewing AI/ML model metadata (accessed through cross-component links), users can explore the training datasets used, understand data preprocessing steps applied, and assess whether models were trained on suitable data for their intended use case.

4.3.3 Data Quality & Governance

Federation and trust provide the foundational principles for a unified data governance model in distributed data spaces and 6G environments. Inspired by Gaia-X, the approach relies on federated identity, transparent governance, and standardized metadata to ensure verifiable participation and trust across organizational boundaries. Complementary mechanisms, aligned with IDSA principles, ensure that data sovereignty, usage control, and compliance are always maintained. Together, these concepts establish a governance fabric in which data can be shared securely, transparently, and in full alignment with contractual, regulatory, and operational requirements.

At the heart of this model lies a set of integrated capabilities that ensure the quality, integrity, and trustworthiness of data throughout its lifecycle. This includes identity and access control, policy evaluation, evidence-based verification, dynamic authorization and formalized data usage agreements. All components collectively enable a trustworthy, sovereign, and interoperable data spaces ecosystem.

A key element of the governance model is **data contracts**²⁵, formal, machine-interpretable agreements that define the rights, obligations, and constraints associated with data access and usage. Data contracts specify who may access data, which parts or attributes are available, under what context or time restrictions, what purposes are allowed, how usage must be tracked, and what compliance conditions apply (e.g., confidentiality levels, regulatory norms, retention periods, provenance constraints). Beyond simply granting or denying access, data contracts articulate the *expected behavior* of all participants and the *permitted lifecycle* of the data. They provide the data provider with full sovereignty, ensuring that data cannot be used beyond what is explicitly permitted. Because data contracts are standardized and machine-readable, they enable interoperability between independent systems, automated enforcement of obligations, and transparent auditing²⁶.

To operationalize data contracts, the governance architecture integrates identity, authorization, and evidence-based decision-making into a unified enforcement flow. **Identity Providers** (IdPs) supply verified identities for users, services, and workloads across domains, enabling federated trust. All access requests pass through **Policy Enforcement Points** (PEPs), which consult a **Policy Decision Point** (PDP) responsible for evaluating requests against the rules defined by the data contract as well as contextual attributes such as time, security posture, compliance requirements, or provenance constraints.

In this architecture, **Policy-as-Code** (PaC)²⁷ is embedded directly into the authorization logic rather than functioning as an isolated element. The **Policy Administration Point** (PAP) expresses the clauses of the data contract as executable, version-controlled machine logic. The PDP applies these policies dynamically in real time, incorporating signals from evidence repositories, identity assertions, and environmental context. This approach enables authorization that is adaptive, explainable, and aligned

²⁵ <https://datacontract.com/>

²⁶ <https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol/contract-negotiation/contract.negotiation.protocol>

²⁷ <https://www.paloaltonetworks.com/cyberpedia/what-is-policy-as-code>

with contractual obligations. It ensures that every access decision is consistent across domains, traceable, and anchored in the explicitly declared intentions of the data provider.

Evidence plays a central role in reinforcing trust and ensuring secure decision-making. A **Transparent Notary Service (TNS)**²⁸ maintains cryptographically verifiable records related to data integrity, provenance, and the security posture of participating entities. This includes SBOMs, attestation results, expected measurements, security configuration states, records of applied policies, and metadata capturing data lineage and transformations. The TER enables the PDP to perform confidence-aware authorization, where data access is contingent not only on contract terms but also on verifiable technical evidence that the requesting entity is in a trustworthy state.

Quality-related metadata associated with each dataset could be digitally signed to ensure integrity, provenance, and traceability. These cryptographically²⁹ protected metadata elements document aspects such as origin, validation processes, transformations, quality assessments, and usage constraints, making them tamper-evident and independently verifiable across organizational boundaries. By combining federated identities, machine-interpretable contracts, and evidence-based authorization, the framework enables trustworthy data exchanges while preserving transparency and accountability. The signed quality metadata can then be used as a reliable input for governance processes, including policy enforcement, auditability, and compliance verification.

The governance architecture can be visualized as an integrated system of modules, each contributing to secure, traceable, and policy-driven data exchange. At the center, the **Policy Decision Point (PDP)** evaluates access requests based on **data contracts**, contextual attributes, and evidence from the **Transparent Notary Service (TNS)**. The **Policy Enforcement Point (PEP)** applies these decisions to data access operations, while the **Policy Administration Point (PAP)** translates contractual rules into executable, version-controlled policies (**Policy-as-Code**). Federated identities are managed by the **Identity Provider (IdP)**, ensuring trust across multiple domains. All interactions and decisions are recorded in the **Transparent Accounting Ledger (TAL)**, providing an auditable trail for compliance and accountability.

The proposed governance architecture builds upon established state-of-the-art principles, including federated identity, policy-based access control (PDP/PEP/PAP), Policy-as-Code, data contracts, and evidence mechanisms aligned with Gaia-X and International Data Spaces Association frameworks. Its novelty lies not in the individual components, but in their deep integration into a unified, executable governance fabric tailored for distributed data spaces and 6G environments. In this model, data contracts become machine-interpretable and directly enforceable through dynamic authorization, while real-time technical evidence feeds into policy evaluation to enable confidence-aware decision-making.

Below is a conceptual diagram (Figure 14) illustrating the interactions between these modules and the flow of data, contracts, policies, and evidence within the governance framework:

²⁸ <https://datatracker.ietf.org/group/scitt/about/>

²⁹ <https://datatracker.ietf.org/group/cose/about/>

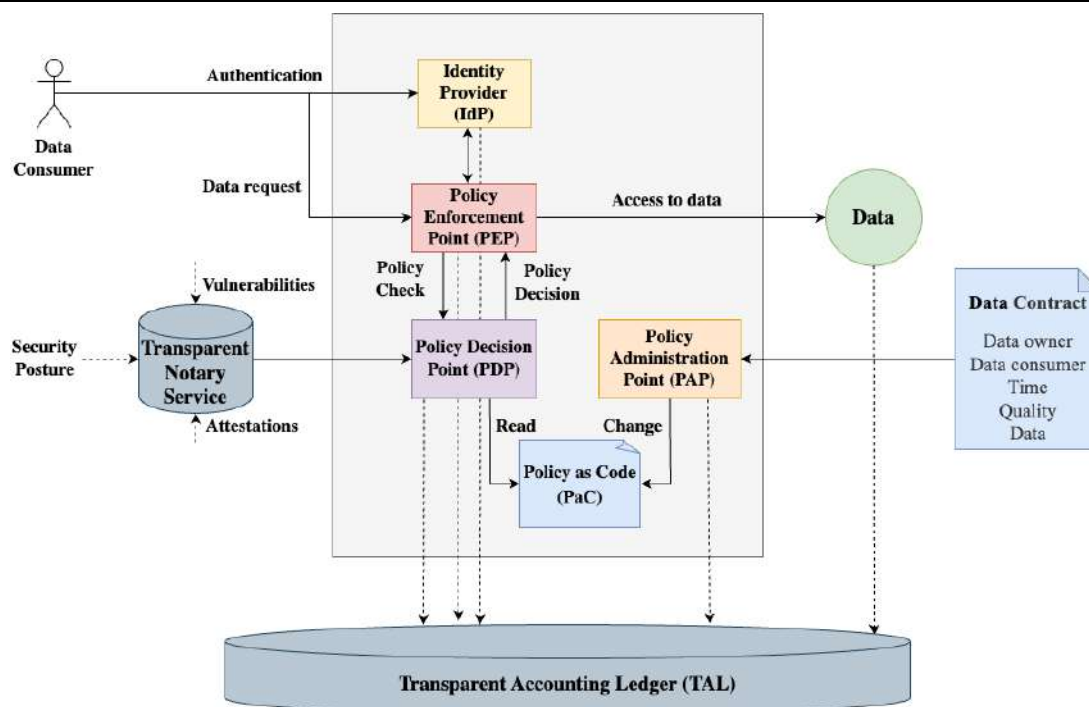


Figure 14: Unified Data Governance Architecture

4.3.4 Federated Catalogue of Data Services & Models

The Federated Catalogue serves as the central metadata aggregation and harmonization component within the 6G Data Space, implementing the semantic backbone for dataset, data service, and AI/ML model discovery across the project's distributed testbed infrastructure. It maintains a hierarchical catalogue structure conformant to DCAT-AP specifications, where each 6G-DALI testbed operates an autonomous sub-catalogue while the central Data space instance federates these distributed sources into a unified, semantically interoperable metadata collection.

The catalogue manages the following primary entity types aligned with DCAT-AP specifications:

- **Catalogues:** Top-level organizational containers representing testbed-specific or thematic groupings of datasets, data services, and models. Catalogue metadata includes descriptive attributes (title, multilingual descriptions, publisher information, thematic classifications), administrative attributes (creation and modification timestamps, license information, access rights statements), contact information (dataset curator emails, helpdesk URLs), and structural attributes (references to sub-catalogues, contained datasets and services, parent catalogues in hierarchical arrangements). In the 6G-DALI context, each testbed maintains its own catalogue instance: the KUL catalogue describing datasets and services from their SDR-based testbed, the ISI/ATH catalogue containing network slicing experiment data and processing services, the EUR catalogue with RAN AI/ML datasets and models, and the VIAVI catalogue providing digital twin simulation outputs and associated analytics services. This catalogue-level organization preserves institutional identity and governance autonomy while enabling federated discovery.
- **Datasets:** Logical data entities describing collections of related data representing the primary unit of discovery for users. Dataset metadata captures multiple facets essential for research reproducibility and appropriate reuse. Descriptive metadata includes titles, multilingual descriptions, keywords, thematic classifications (following controlled vocabularies like EuroVoc), and related resources. Provenance metadata identifies creators, publishers, contributors, and creation/modification dates, establishing data lineage and accountability. Temporal metadata specifies coverage periods (start/end dates for time-bounded datasets), update frequency (continuously updated, daily snapshots, one-time collections), and temporal

resolution (measurement intervals). Spatial metadata describes geographical coverage through bounding boxes, specific location identifiers, or administrative region references. Quality metadata incorporates automatically computed quality scores, validation status indicators, and completeness assessments derived from the Data Quality & Governance module. Distribution references link datasets to their concrete access mechanisms. Additionally, 6G-specific metadata extensions capture network configuration parameters, experimental conditions, measurement methodologies, and equipment specifications essential for interpreting dataset semantics in telecommunications research contexts.

- **Distributions:** Concrete access endpoints for datasets, representing specific file formats, serialization schemes, or access protocols through which dataset content can be retrieved. A single logical dataset typically has multiple distributions accommodating diverse user preferences and technical requirements—for example, a network performance dataset might offer CSV distribution for spreadsheet analysis, JSON for programmatic access, Parquet for big data processing, and an SQL endpoint for query-based access. Distribution metadata specifies format (using IANA media types), file size, access URL, integrity checksums (SHA-256 hashes for download verification), license terms (which may vary by distribution), and access service descriptions (for API-based access). Distributions may also reference data services (persistent query endpoints, streaming interfaces) rather than static files, supporting real-time data access scenarios.
- **Data Services:** Computational services that process, transform, or provide access to data within the 6G-DALI ecosystem. Data Services represent the operational components of the DataOps stack, including data cleaning services, augmentation algorithms, format conversion utilities, aggregation engines, and streaming data interfaces. Data Service metadata captures service endpoints (API URLs, protocol specifications), input/output data formats, processing capabilities (supported transformations, quality improvement operations), performance characteristics (processing latency, throughput limits), and usage policies (authentication requirements, rate limits, access restrictions). Services are catalogued alongside datasets to enable users to discover not only data but also the tools available for data processing and transformation. This integration supports workflows where users identify relevant datasets and appropriate services to prepare those datasets for AI/ML experimentation. Data Services are linked to the datasets they can process through DCAT-AP *dcat:servesDataset* relationships, enabling discovery queries such as "find services capable of processing RAN performance datasets in CSV format."
- **ML Models:** Trained artificial intelligence and machine learning models developed within the 6G-DALI MLOps framework. ML Model metadata extends DCAT-AP with AI/ML-specific properties capturing model architecture descriptions, training configuration (hyperparameters, optimization algorithms), performance metrics (accuracy, inference latency, computational requirements), deployment information (target platforms, API endpoints), and versioning information. Critically, ML Models maintain explicit relationships to their training datasets through *trainedOn* properties, establishing bidirectional linkages that support reproducibility research, transfer learning workflows, and dataset quality impact analysis. This comprehensive model cataloguing treats AI/ML models as first-class research artifacts alongside datasets, enabling unified discovery of both training data and trained intelligence within a single catalogue interface.

Catalogue metadata is maintained in RDF (Resource Description Framework) format strictly conformant to **DCAT-AP version 3.0.1**, the European application profile for data portals. This standardization ensures semantic interoperability with European data infrastructure (data.europa.eu), enables automated metadata validation against DCAT-AP SHACL shapes, and supports rich entity linking through URI-based identification. The RDF representation facilitates complex metadata queries through SPARQL (datasets created by specific publishers within date ranges, distributions available under open licenses with specific formats, catalogues containing datasets about particular themes),

relationship traversal (finding all datasets derived from a given source dataset through DataOps transformations), and inference (automatically classifying datasets into categories based on keyword analysis).

A critical extension to standard DCAT-AP implemented by the 6G-DALI Federated Catalogue is the integration of AI/ML model metadata alongside dataset metadata. While DCAT-AP primarily targets data cataloguing, 6G-DALI treats trained AI/ML models as first-class catalogue entities, capturing model architecture descriptions, training hyperparameters, performance metrics, and crucially, bidirectional linkages to training datasets. This integration enables several key workflows: researchers can discover which datasets were used to train models exhibiting certain performance characteristics, assess dataset suitability for new model training by examining models previously trained on similar data, and verify model reproducibility by locating exact training dataset versions. The model-dataset linkage metadata follows emerging community standards for ML model cataloguing (Croissant, ML Schema) while maintaining DCAT-AP compatibility through extension properties.

The catalogue also maintains rich provenance metadata using the PROV-O (Provenance Ontology) vocabulary, capturing the derivation chains through which datasets are created. When DataOps ELT pipelines transform raw testbed measurements into cleaned and augmented datasets, the resulting derived datasets include `prov:wasDerivedFrom` links to source datasets, `prov:wasGeneratedBy` links to pipeline execution records, and `prov:wasAttributedTo` links to the agents (human operators or automated processes) responsible for data generation. This provenance graph enables users to trace dataset lineage, understand data quality impacts of transformation steps, and assess whether derived datasets are suitable for specific analytical purposes.

Baseline: The Federated Catalogue builds upon **Piveau Hub Repository**, an open-source data catalogue platform designed for managing and federating DCAT-AP-compliant metadata across distributed environments. Piveau Hub implements **DCAT-AP**, a standardized European metadata model for dataset description and catalogue interoperability. Metadata is stored using an **RDF triple store** (Virtuoso³⁰), a semantic database technology that enables structured, machine-readable representation of dataset information and supports advanced querying through **SPARQL**³¹, a semantic query language. To enable distributed catalogue federation, the platform supports **OAI-PMH harvesting**, allowing metadata to be exchanged between multiple catalogue instances. Metadata quality and standards compliance are ensured through automated validation against **SHACL constraint shapes**, which verify structural and semantic correctness. The repository further supports multilingual metadata, controlled vocabularies (e.g., EuroVoc thematic classifications, standardized media types, and SPDX license identifiers), and versioned metadata management, ensuring interoperability with European data portal infrastructures such as data.europa.eu.

Innovation and Extension: The 6G-DALI implementation extends the baseline Piveau Hub Repository through several innovative enhancements tailored to AI/ML-centric 6G research workflows:

- **Integrated Data Service Cataloguing:** Extension of the catalogue to include Data Services as first-class entities alongside datasets, implementing DCAT-AP patterns for service description with 6G-specific enhancements. Data Service metadata captures not only service endpoints and interfaces but also processing capabilities, performance characteristics, and dataset compatibility information. The catalogue establishes explicit `dcat:servesDataset` relationships linking services to compatible datasets, enabling integrated discovery workflows where users locate both data and the tools to process it. This innovation bridges the traditional separation between data catalogues and service registries, providing a unified view of the DataOps ecosystem.

³⁰ <https://vos.openlinksw.com/owiki/wiki/VOS/VOSRDFFAQ>

³¹ <https://www.w3.org/TR/sparql11-query/>

- **Comprehensive AI/ML Model Cataloguing:** Extension of the DCAT-AP schema to include AI/ML model metadata as first-class catalogue entities. Model records capture architecture descriptions (neural network topologies, algorithm types), training configuration (hyperparameters, optimization algorithms, regularization techniques), performance metrics (accuracy, precision, recall, F1 scores, inference latency), computational requirements (memory footprint, GPU utilization), and dataset dependencies through explicit trainedOn relationships linking models to their training datasets. This extension positions the catalogue as a unified discovery interface for data, services, and AI artifacts, eliminating the traditional separation between data catalogues and model registries and supporting end-to-end MLOps workflows.
- **6G Testbed Metadata Vocabularies:** Integration of telco-domain vocabularies describing 6G network configurations, experimental conditions, and measurement methodologies. Testbed-specific metadata includes RAN configuration parameters (frequency bands, bandwidth, MIMO configurations, cell layouts), core network deployment models (standalone vs. non-standalone 5G, network slicing configurations), edge computing infrastructure (available computing resources, network latency profiles, geographic distribution), and experimental procedures (traffic patterns, mobility models, interference scenarios). These domain extensions preserve essential context for interpreting dataset semantics, enabling researchers to assess dataset applicability to their specific research questions.
- **Quality-Enriched Metadata:** Automatic incorporation of dataset quality scores and validation reports computed by the Data Quality & Governance module into catalogue metadata. Quality enrichment includes completeness scores (percentage of mandatory DCAT-AP properties populated), consistency indicators (format validation, range checks on numerical metadata), accuracy assessments (cross-validation with authoritative sources), and trustworthiness signals (cryptographic signatures, integrity checksums). Quality metadata enables quality-aware discovery workflows, supports automated data curation processes, and helps researchers prioritize high-quality datasets for AI/ML training.

4.3.5 Metadata Storage & Indexing

The Metadata Storage & Indexing layer provides the persistent storage infrastructure and high-performance query mechanisms that underpin the Federated Catalogue's operation. This component implements a dual-storage architecture balancing semantic expressiveness with query performance, ensuring that the catalogue can support both complex semantic queries over metadata relationships and sub-second response times for user-facing discovery interfaces.

RDF Triple Store: The primary authoritative storage backend maintains all catalogue metadata as an RDF graph persisted in an OpenLink Virtuoso triple store. Key capabilities include:

- **SPARQL 1.1 Query Protocol:** Support for complex graph pattern matching, optional property clauses, filter expressions, aggregation operations, and subqueries
- **Named Graphs:** Metadata organization where each graph represents a logical partition (testbed-specific metadata, quality assessment results, provenance records), enabling graph-level access control and selective query targeting
- **Inference and Reasoning:** Support for RDFS and OWL inference rules, enabling automatic relationship derivation
- **RDF Serialization Formats:** Export and import capabilities for Turtle, RDF/XML, N-Triples, JSON-LD, and N-Quads

The triple store serves as the definitive source of truth for all catalogue metadata, with all metadata updates flowing through validation pipelines ensuring DCAT-AP conformance before persistence.

Search Index: Complementing the triple store, an Elasticsearch-based search index provides optimized full-text and faceted search capabilities. The search index structure de-normalizes RDF graph data into document-oriented records optimized for common query patterns:

- **Full-Text Search:** Inverted indices with language analyzers supporting stemming, stop word removal, and synonym expansion
- **Faceted Navigation:** Pre-computed aggregations enabling efficient faceted search across categorical, numerical, temporal, and spatial attributes
- **Quality-Based Filtering:** Dedicated index structures for dataset quality scores and validation status

Synchronization Mechanisms: Automated pipelines maintain consistency between the authoritative RDF triple store and the performance-optimized search index:

- **Real-Time Incremental Sync:** Webhook-triggered synchronization when metadata changes occur, with delta detection ensuring only modified records are re-indexed
- **Periodic Full Synchronization:** Scheduled batch processes that completely rebuild search indices from the triple store, ensuring eventual consistency
- **Change Detection:** Timestamp-based change tracking identifies modified metadata entities since the last synchronization

Metadata Versioning: The storage layer implements temporal versioning of metadata records, maintaining historical snapshots of metadata evolution. Versioning metadata is captured using the PROV-O vocabulary, recording who made changes, when changes occurred, and why. Version management enables temporal queries, complete audit trails, and rollback capabilities for error recovery.

Caching Layer: Strategic caching reduces load on both storage backends while improving response latency:

- **Query Result Caching:** Frequently executed SPARQL queries and Elasticsearch searches have their results cached with time-based expiration
- **Entity-Level Caching:** Individual metadata records are cached after retrieval for fast repeated access
- **Computed Aggregation Caching:** Expensive aggregation operations are pre-computed and cached with invalidation triggered by relevant metadata updates

Baseline: The Metadata Storage & Indexing infrastructure builds upon the established Piveau Hub technology stack, combining OpenLink Virtuoso RDF store with Elasticsearch for search optimization. This combination has been proven in large-scale European data portal deployments and provides DCAT-AP compliance and SPARQL endpoint capabilities out of the box.

Innovation and Extension: The 6G-DALI implementation enhances the baseline storage architecture through several targeted innovations addressing the unique requirements of AI/ML-centric 6G research workflows:

- **Quality Metadata Indexing:** Specialized index structures optimized for dataset quality assessment results, capturing quality dimensions (completeness, consistency, accuracy, trustworthiness) and enabling efficient multi-dimensional quality filtering.

4.3.6 Discovery Services

Discovery Services provide intelligent search and recommendation capabilities that enable users to efficiently locate relevant datasets, data services, and AI/ML models within the 6G Data space. These services operate on the semantic metadata graph maintained by the **Federated Catalogue** and persisted through the **Metadata Storage & Indexing layer**, offering multiple discovery modalities tailored to different user expertise levels and search intentions.

Faceted Search Engine: Multi-dimensional filtering system enabling progressive query refinement through combinations of metadata attributes. Available facets include:

- **Structural Facets:** Format, file size ranges, number of distributions

- **Provenance Facets:** Publisher organization, testbed source, creation date ranges
- **Thematic Facets:** Dataset categories, keywords, theme classifications
- **Quality Facets:** Completeness score ranges, validation status, overall quality thresholds
- **Licensing Facets:** License type, access restrictions
- **Temporal Facets:** Coverage period, temporal resolution, collection frequency

The faceted interface displays result counts for each facet value before selection, enabling users to assess result set sizes. Facet combinations operate conjunctively with intuitive visual feedback showing active filters.

Full-Text Search: Keyword-based search across textual metadata fields with language-aware text processing, relevance ranking combining TF-IDF scoring with metadata field importance weights and quality score boosts, query enhancement through synonym expansion and controlled vocabulary mapping, and search result highlighting showing matched terms in context.

SPARQL Query Interface: Advanced semantic query capability exposing the RDF knowledge graph through a SPARQL endpoint and interactive query builder based on YASGUI components. Power users can construct complex queries combining multiple metadata dimensions, traverse relationships across multiple hops, apply inference leveraging RDFS/OWL reasoning, and execute federated queries spanning multiple endpoints.

Baseline: Discovery Services build upon Piveau Hub Search, which provides Elasticsearch-based indexing and search capabilities for DCAT-AP catalogues, including faceted search, full-text search with basic relevance ranking, and SPARQL query endpoints with YASGUI integration.

Innovation and Extension: 6G-DALI enhances baseline discovery capabilities through several targeted innovations:

- **Quality-Informed Ranking:** Integration of dataset quality scores into search ranking algorithms, combining completeness scores, validation status, consistency indicators, and trustworthiness signals to surface high-quality datasets prominently.
- **Cross-Entity Discovery:** Integrated discovery across datasets, data services, and ML models enabling users to find not only data but also processing services and models trained on relevant data through bidirectional relationship traversal.

4.3.7 Interfaces

The 6G Data Space exposes and consumes several categories of interfaces enabling integration with external components and users.

- **Web User Interface (WEB)** provides browser-based human access to Data space functionality, enabling interactive dataset discovery through faceted and full-text search, metadata viewing and export in multiple RDF formats, catalogue browsing with hierarchical navigation, metadata management capabilities for authorized users including editing and publishing workflows, and interactive SPARQL query execution through integrated YASGUI components. Access is secured through OAuth 2.0/OIDC authentication integrated with the project's Keycloak identity provider, supporting single sign-on with institutional identity providers. The interface is implemented as a responsive HTML5/JavaScript web application based on piveau-hub-ui technology.
- **Metadata Registration API** enables programmatic metadata ingestion supporting automated workflows where metadata is registered, when experiments are executed, and enriched, when quality assessments complete. The API accepts DCAT-AP-compliant metadata in multiple RDF serializations including Turtle, JSON-LD, and RDF/XML formats. All incoming metadata undergoes validation against DCAT-AP 3.0.1 SHACL shapes before persistence, with validation errors returned to clients for correction. Authentication is handled through OAuth 2.0 API tokens using client credentials flow, with authorization scopes controlling metadata creation and update permissions.

- **Dataset Discovery API** provides programmatic dataset discovery capabilities for external 6G-DALI components including DataOps pipelines, MLOps training workflows, and testbed services. Access is mediated through the Data Governance component which evaluates access policies before returning results. The API supports faceted search across multiple metadata dimensions including quality scores, data formats, publishers, temporal coverage periods, and thematic categories. Results can be filtered and paginated, with response formats including JSON metadata summaries and full DCAT-AP RDF representations. Authorization is policy-based, with results filtered according to the requesting component's access privileges and the governance rules associated with each dataset.
- **SPARQL Endpoint** exposes direct semantic query access to the RDF metadata graph, implementing the W3C SPARQL 1.1 protocol. The endpoint supports all SPARQL query forms including SELECT for tabular results, CONSTRUCT for graph generation, ASK for boolean queries, and DESCRIBE for resource descriptions. Federated query capabilities enable queries spanning multiple SPARQL endpoints using the SERVICE keyword. Authentication requirements are policy-dependent, with some metadata accessible publicly while sensitive metadata requires OAuth 2.0 token authentication. The endpoint is implemented through the Virtuoso triple store's native SPARQL interface.
- **Identity Provider Interface** integrates with Keycloak for authentication and authorization operations including user authentication, OAuth 2.0 token issuance and validation, and user attribute retrieval for access control decisions. The interface implements OpenID Connect and OAuth 2.0 protocols enabling federated authentication with institutional identity providers and API token management for programmatic access by external 6G-DALI components.

4.4 Adaptation Layer

This section presents the detailed logical architecture of the 6G-DALI Adaptation Layer. The Adaptation Layer facilitates seamless and reliable connectivity between the 6G-DALI framework and its associated testbeds. It ensures the efficient execution of core functional capabilities, including **dynamic experiment management**, real-time **control between the 6G-DALI framework and distributed testbed** infrastructures, and **bidirectional data exchange** with the data lake. By providing common interfaces and robust communication mechanisms, the framework enables seamless orchestration of experiments, consistent data collection, and synchronized interaction across the 6G-DALI ecosystem.

4.4.1 Functional Architecture

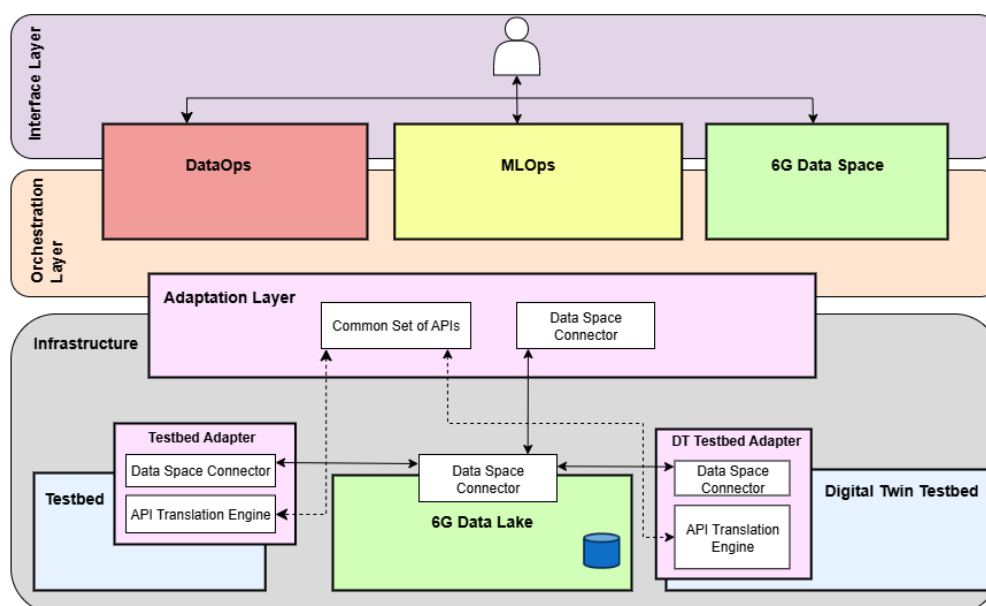


Figure 15: Adaptation layer architecture

As shown in the Figure 15, the Adaptation Layer is responsible for enabling seamless communication between the 6G-DALI framework and the connected partner testbeds. This layer acts as an interoperability enabler, abstracting the heterogeneity of testbed interfaces and providing a unified interaction model for experiment management, data exchange, and orchestration. On the 6G-DALI framework side, the Adaptation Layer is implemented as a core functional component. Since different testbeds support diverse types of experiments and expose heterogeneous interfaces for experiment management, the presence of a common interface is essential to ensure smooth, scalable, and technology-agnostic communication between the 6G-DALI framework and the testbeds. To achieve this, the Adaptation Layer adopts a common set of APIs that simplify and harmonize experiment management operations across all integrated testbeds.

On the testbed side, adapters are deployed to interface with the Adaptation Layer. These adapters serve as translation mechanisms between the common APIs exposed by the Adaptation Layer and the testbed-specific APIs used for local experiment control and resource management.

In addition to experiment orchestration, the Adaptation Layer includes data space connectors on both the 6G-DALI framework side and the testbed side, as well as integration with the data lake. This design simplifies and streamlines data transactions among all involved parties.

Once a new dataset is generated as a result of an experiment executed on a testbed, the dataset must first be registered in the 6G-DALI data space. After successful registration, the dataset can be securely transferred to the data lake for persistent storage. These data transfers are governed by predefined data-sharing policies agreed upon between data producers and data consumers. All data exchange processes, including dataset registration, policy enforcement, and secure data transfer, are facilitated through the data space connectors, ensuring compliance, traceability, and controlled access.

4.4.2 Common Set of APIs

For the common set of APIs in the Adaptation Layer, DataOps communication can be realized by extending the Adaptation Framework [1] developed in the SUNRISE-6G project. This framework already provides APIs for deploying, executing, and managing experiments in heterogeneous testbeds and can be reused and extended to support 6G-DALI requirements as described in Table 7. These operations include, but are not limited to:

- Starting and stopping experiments,
- Providing and updating experiment configurations,
- Monitoring experiment execution status,
- Collecting experiment metadata and runtime information.

By exposing these operations through common APIs, the Adaptation Layer enables uniform experiment control regardless of the underlying testbed implementation.

Table 7: Adaptation Framework APIs developed in the SUNRISE-6G project

Endpoints	Method	Description
../availability/testbed	GET	Retrieves the operational status of the testbed. This endpoint allows the Adaptation Framework to verify if the testbed is online and ready to receive requests.
../experiments	POST	Triggers the start of an experiment on the testbed, based on the configurations provided in the request body. This endpoint is crucial for executing experiments on the testbed.
../experiments/{exp_id}	DELETE	Instructs the testbed to stop a running experiment identified by the provided exp_id. This endpoint allows

		the Adaptation Framework to gracefully terminate experiments, ensuring proper cleanup and resource release on the testbed.
../experiment/status	POST	Allows the testbed to report immediate updates on the status of a running experiment to the Adaptation Framework. This endpoint is crucial for real-time monitoring and should be used each time the experiment status updates.

For MLOps communication between the 6G-DALI framework and the testbeds, the meta-orchestrator can directly interface with a wide range of MLOps stacks deployed within the testbeds. This enables centralized control and coordination of machine learning workflows across distributed environments. Through this integration, these operations include but are not limited to:

- Model training
- Hyperparameter optimization
- Model deployment
- Model inference
- End-to-end ML pipeline deployment and execution

This approach allows 6G-DALI to support advanced AI-driven experimentation while leveraging the native MLOps capabilities of each testbed.

4.4.3 Data Space Connector

The Data Space Connector serves as the integration bridge between individual testbeds and the central 6G-DALI infrastructure, automating the dual processes of **dataset file transfer to the Data Lake** and **metadata registration in the Data Space** following experiment completion. This component resides at each testbed location, operating as part of the testbed's local infrastructure to handle the immediate post-experiment workflows that make newly generated data discoverable and accessible across the federated 6G-DALI ecosystem.

When experiments complete and generate output files, the Data Space Connector detects these outputs and orchestrates a coordinated ingestion process. The connector reads experiment metadata including network configurations, measurement parameters, temporal coverage, and equipment specifications from experiment execution logs or configuration files. It packages the raw dataset files, typically in CSV format but supporting multiple formats including JSON, Parquet, and domain-specific telemetry formats, and transfers them to the central Data Lake's S3-compatible object storage. The transfer process includes integrity verification through checksum computation, ensuring that files arrive intact and can be reliably accessed for subsequent processing.

Simultaneously with data transfer, the connector constructs DCAT-AP-compliant metadata records describing the newly generated datasets. This metadata construction process translates testbed-specific experiment information into standardized Data Space metadata vocabularies, mapping testbed experiment parameters to DCAT-AP properties and 6G-specific metadata extensions. The connector populates descriptive metadata including dataset titles derived from experiment names, descriptions extracted from experiment documentation, keywords reflecting measurement types and network configurations, and thematic classifications appropriate to the data domain. Provenance metadata captures the testbed source, experiment identifiers, creation timestamps, and responsible parties. Temporal and spatial metadata describes the measurement coverage period and geographical location of the testbed. Distribution metadata includes references to the Data Lake storage URLs where files can be accessed, format specifications, and file size information.

The connector invokes the Data Space's metadata registration API to persist these constructed metadata records, making datasets immediately discoverable through Data Space search interfaces. Registration

includes initial quality indicators based on experiment execution status, data completeness checks, and format validation, providing users with preliminary quality signals before comprehensive quality assessment pipelines execute. The connector handles registration failures gracefully, implementing retry mechanisms for transient network issues and logging errors for manual investigation when metadata validation fails due to incomplete or malformed information.

Beyond basic registration, the Data Space Connector supports the configuration and enforcement of testbed- and asset-specific governance policies, defined at the time of dataset registration, that control which datasets are registered publicly versus restricted to consortium-internal access. Some testbeds may generate sensitive measurement data subject to access restrictions, preliminary experimental data not yet validated for general use, or proprietary configurations requiring confidentiality. The connector applies testbed-configured policies during metadata construction, setting appropriate access control attributes and license terms that the Data Space's governance mechanisms enforce when users attempt discovery or retrieval.

Baseline: The Data Space Connector builds upon the Eclipse Data space Components (EDC) framework, an open-source implementation of data space connector functionality aligned with International Data Spaces Association (IDSA) specifications and Gaia-X federated services principles. EDC provides foundational capabilities for secure data exchange including policy-based access control, contract negotiation protocols, data transfer coordination, and identity management integration. The baseline connector implements standard data space patterns for federated data sharing with sovereignty-preserving access policies and audit logging of data transactions.

Innovation: The 6G-DALI implementation extends the EDC baseline through several innovations specifically addressing the requirements of automated, experiment-driven data ingestion in federated 6G testbed environments:

- **Experiment-Triggered Automation:** Unlike traditional data space connectors designed for interactive data sharing workflows where humans initiate data transfers and negotiate contracts, the 6G-DALI connector implements fully automated operation triggered by experiment completion events. The connector monitors testbed experiment execution systems, detecting output generation through file system watches, event notifications, or periodic polling. When experiments are completed, the connector automatically initiates ingestion without manual intervention, dramatically reducing the time between data generation and data availability for consortium researchers. This automation is critical in high-throughput experimental scenarios where testbeds may execute dozens of experiments daily, making manual data publishing workflows impractical.
- **DCAT-AP Metadata Generation:** The connector implements automatic metadata generation translating testbed-specific experiment parameters into DCAT-AP-compliant metadata records. Rather than requiring experimenters to manually author metadata forms describing datasets, the connector extracts information from experiment configurations, execution logs, and testbed infrastructure metadata, automatically populating the extensive metadata properties required for effective Data Space cataloging. This automation ensures metadata completeness and consistency while eliminating tedious manual documentation that experimenters often omit or abbreviate when under time pressure to analyse results.
- **Dual-Path Orchestration:** The connector orchestrates parallel data and metadata flows, transferring files to the Data Lake while simultaneously registering metadata in the Data Space. This dual-path architecture maintains the critical separation between data storage (optimized for large-scale file management) and metadata storage (optimized for discovery queries) while presenting a unified abstraction to testbed infrastructure. The connector handles the complexity of coordinating these parallel operations, ensuring that metadata references point to valid Data Lake locations and that registration only completes after successful file transfer verification.
- **Testbed-Specific Adaptation:** Recognizing that each testbed in the 6G-DALI consortium operates different experimental infrastructure with unique configuration formats, output

conventions, and metadata schemas, the connector implements an adapter pattern enabling testbed-specific customization. Adapters translate between testbed-native formats and standardized Data Space representations, handling variations in how testbeds describe network configurations, measurement methodologies, and equipment specifications. This pluggable architecture allows the core connector logic to remain consistent across testbeds while accommodating the diversity of experimental platforms including software-defined radio testbeds, network slicing environments, and digital twin simulators.

4.4.4 Testbed Adapter

The Testbed Adapter is responsible for enabling experiments managed through the 6G-DALI framework to be correctly interpreted and executed by the underlying testbed. It provides the necessary abstraction and translation mechanisms to align 6G-DALI experiment control interfaces with testbed-specific capabilities. In addition, the Testbed Adapter ensures seamless and secure communication between the 6G-DALI Data space and the 6G-DALI Data Lake, supporting the orchestration of data transfer workflows in compliance with applicable data space policies.

To realize these functions, the Testbed Adapter is composed of two main components as shown in Figure 15: an **API Translation Engine** and a **Data Space Connector**.

- **API Translation Engine:** The API Translation Engine maps and translates the common set of APIs defined in the adaptation layer, used to manage experiments through the 6G-DALI framework, into the local APIs exposed by each testbed. This approach enables uniform control of experiment execution across heterogeneous testbed environments.
- **Data Space Connector:** The Data Space Connector is an architectural component of the 6G-DALI Data Space. It is deployed at each Testbed Adapter to enable seamless communication between the testbeds, the 6G-DALI Data Space, and the 6G-DALI Data Lake. The connector is responsible for registering datasets generated during experiments, managing dataset metadata, and orchestrating the secure transfer of datasets from the testbed to the Data Lake in accordance with applicable data space policies.

4.5 Testbeds

The 6G-DALI testbeds constitute the experimentation environments where data is generated. Testbeds may also provide the required infrastructure to train and validate ML models.

4.5.1 Functional Architecture

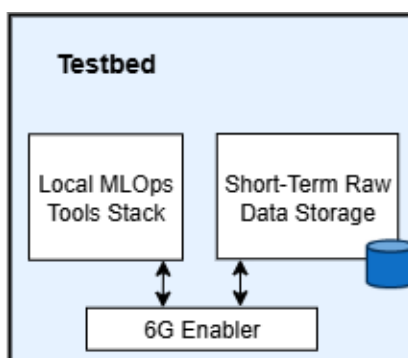


Figure 16: Testbed functional architecture

This section presents the generic functional architecture of the 6G-DALI testbeds, defining a harmonized structure that ensures consistent integration with the 6G-DALI framework while preserving the flexibility of individual deployments. Each testbed is organized around three core components: a Local MLOps Tools Stack, a Short-Term Raw Data Storage facility, and a 6G Enabler. As illustrated in Figure 16, these components collectively enable data generation through experiments, local AI lifecycle management and local AI experimentation.

4.5.2 Local MLOps Tools Stack

The Local MLOps Tools Stack provides the capabilities to deploy, validate, and optionally train AI/ML models within the testbed. It may support core ML operations such as experiment configuration, hyperparameter optimization, performance evaluation, model versioning, and artifact storage. Depending on the testbed’s capabilities, it may also include experiment tracking services and pipeline orchestration tools. These tools are integrated with the central 6G-DALI MLOps orchestrator through the meta-orchestrator. In particular, local MLOps stacks are registered to the central meta-orchestrator, enabling them to be discovered, configured, and triggered as part of end-to-end ML workflows. This registration allows the central MLOps meta-orchestrator to maintain visibility over available local resources and manage distributed experiment execution across multiple testbeds. As a result, distributed experimentation can be performed in a controlled and unified manner, while preserving the autonomy and specialization of each testbed’s local AI environment.

4.5.3 Short-Term Raw Data Storage

The Short-Term Raw Data Storage component is responsible for **temporarily storing raw data** generated during experiments or operational measurements within the testbed. It acts as a local buffering and staging area before data is loaded to be curated, filtered, and transferred to the central 6G Data Lake through the data space connectors. This storage facility supports high-throughput data ingestion from the 6G Enabler and ensures that raw, potentially high-volume data can be preserved for validation. By separating short-term storage from long-term storage, the architecture supports efficient data handling and reduces unnecessary data transfers.

4.5.4 6G Enabler

The 6G Enabler represents the communication and network experimentation backbone of the testbed. It provides the programmable 6G connectivity infrastructure and network capabilities required to execute advanced networking experiments and generate data. Each testbed supports different 6G-enabler capabilities like Cognitive Compute Continuum, Open-RAN experimentation, and PHY layer sensing. This complementarity allows the orchestration of various 6G data generation, network telemetry collection, and ML experiments execution through the 6G-DALI facility.

4.6 Digital Twin Testbed

This section presents the detailed logical architecture of the Digital Twin Testbed, which provides the foundational resources required to support data generation experiments.

4.6.1 Functional Architecture

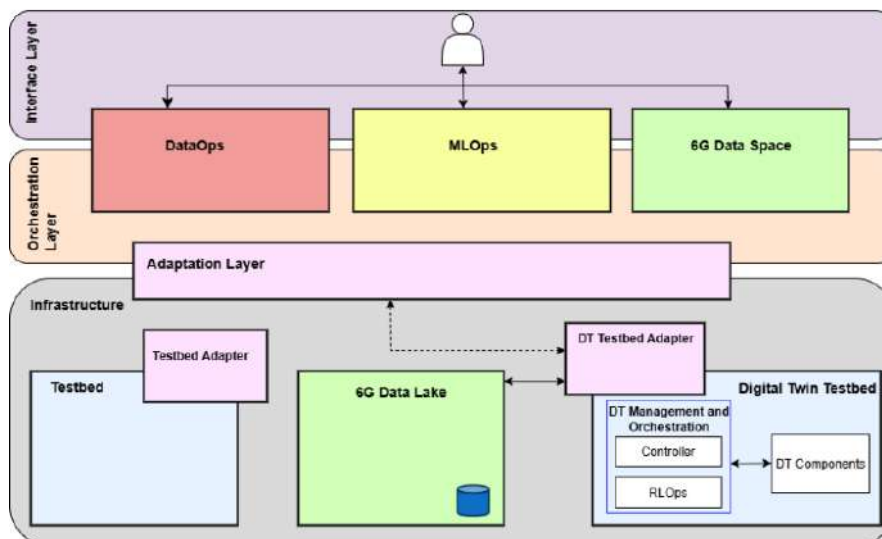


Figure 17: The Digital Twin Architecture

The Digital Twin (DT) architecture is presented in Figure 17. Requests originate from the 6G-DALI Core Infrastructure and flow through the Adaptation Layer, which expresses the desired experiment in a standardized compatible form. The DT Testbed Adapter receives these requests via the API Translation Engine, which converts them into internal commands that the **Controller** can execute. The Controller then configures and launches the Digital Twin based on the experiment's goal. When the objective is **RL validation** through RLOps, the Controller would ensure the DT runs in an interactive mode where the actions suggested by the RL agent can be applied. When the goal is **data collection**, the DT could run in non-interactive mode where it generates data for the duration of the simulation, and the data will finally be forwarded to the Data Space via the Data Space connector.

4.6.2 DT Management and Orchestration

The DT Management and Orchestration component (Controller) controls the simulation lifecycle including scenario creation, configuration, starting/pausing/stopping simulations, resource management, and KPI collection orchestration. A key enabler is the DT Application Development Kit (ADK) Pacing mode that allows the controller to control simulation time progression, enabling step-by-step reinforcement learning training where AI agents can pause, observe, act, and resume. This pacing capability is essential for RLOps integration, allowing RL algorithms to be validated in a controlled, repeatable environment. In addition, the ADK also provides means to collect data directly when the goal is dataset generation.

4.6.3 DT components

Depending on the desired scope of the DT, it would include different network components. For a RAN Digital Twin, it would include sophisticated RAN simulation, including base stations, users and interfaces among them. All of these components would be configurable according to the needs of the experiments and the desired scenarios. On the other hand, a Core Network DT could support emulating the Core Network (CN) for a given scenario, supporting various features including but not limited to NWDAF. A more sophisticated would include the RAN and CN components so that it covers a wide range of the real network.

4.6.4 DT Testbed Adapter

The DT Testbed Adapter serves as the integration bridge between the DTT and the 6G-DALI ecosystem. It contains two key components: the API Translation Engine that converts upper layer calls into calls compatible with the DT components, and the Data Space Connector that handles data export to the 6G Data Lake.

4.6.5 Interfaces

The Digital Twin includes various interfaces:

- **DT Adapter ↔ Controller:** Upper API Translation to Simulation parameters/configuration
- **Controller ↔ Digital Twin:** Control of the DT using the ADK (REST API)
- **Digital Twin ↔ InfluxDB:** Simulations local storage and query
- **Data Space Connector ↔ Data Lake:** Interface for Datasets export
- **RLOps ↔ ADK:** Interface to run RL algorithms within the DT

4.7 6G Data Lake

This section presents the detailed logical architecture of the 6G Data Lake. The 6G Data Lake is a core component of the infrastructure architectural block of the 6G-DALI framework and serves as the central repository for all data assets generated, processed, and consumed throughout the system. Its primary role is to provide scalable, reliable, and high-performance storage for both raw experimental data and curated datasets produced during the execution of DataOps and MLOps workflows. By acting as a persistent data backbone, the 6G Data Lake enables the end-to-end data lifecycle envisioned in 6G-DALI,

from experiment-driven data generation to AI/ML model training, validation, and benchmarking. Within the overall 6G-DALI architecture, the Data Lake is intentionally designed as a storage-centric infrastructure component, decoupled from data governance, metadata management, and access policy enforcement. These responsibilities are delegated to the 6G Data Space, in line with Gaia-X and IDSA principles. The Data Lake therefore focuses on ensuring data availability, durability, performance, technical accessibility, higher-level concerns related to sovereignty, discovery, and usage control. This is a key architectural design decision that enhances scalability, interoperability, and architectural clarity. This operational model allows the project to rely on a robust, cloud-native storage environment while retaining full architectural control over data lifecycle management, integration patterns, and compliance with the 6G-DALI reference architecture.

4.7.1 Functional Architecture

The functional architecture of the 6G Data Lake follows a cloud-native, modular design and is implemented using object storage technologies exposed through standardized, **S3-compatible interfaces**. The Data Lake must be deployed on infrastructure and is accessed by the rest of the 6G-DALI framework exclusively through well-defined programmatic interfaces, ensuring loose coupling and technology independence. From a logical perspective, the Data Lake is structured into two distinct storage domains that reflect different stages of the data lifecycle: **short-term raw data storage** and **long-term final data storage**. This logical separation enables differentiated retention policies, access patterns, and performance optimizations while relying on a common underlying storage technology. The separation is enforced at the level of storage namespaces or buckets and is transparent to higher-layer components such as DataOps and MLOps. Integration with testbeds, the DTT, and the 6G Data Space is achieved through Data Space Connectors deployed as part of the Adaptation Layer. These connectors orchestrate the automated ingestion of datasets into the Data Lake and ensure consistency between stored data and metadata registered in the 6G Data Space. This integration implements a dual-path orchestration model, in which data files are transferred to the Data Lake while metadata is simultaneously registered and indexed in the Data Space. The baseline architecture builds on established object storage and Eclipse Data Space Components technologies, while the main innovation introduced in 6G-DALI lies in the tight coupling between experiment execution, data ingestion, and metadata publication, enabling fully automated, high-throughput data workflows.

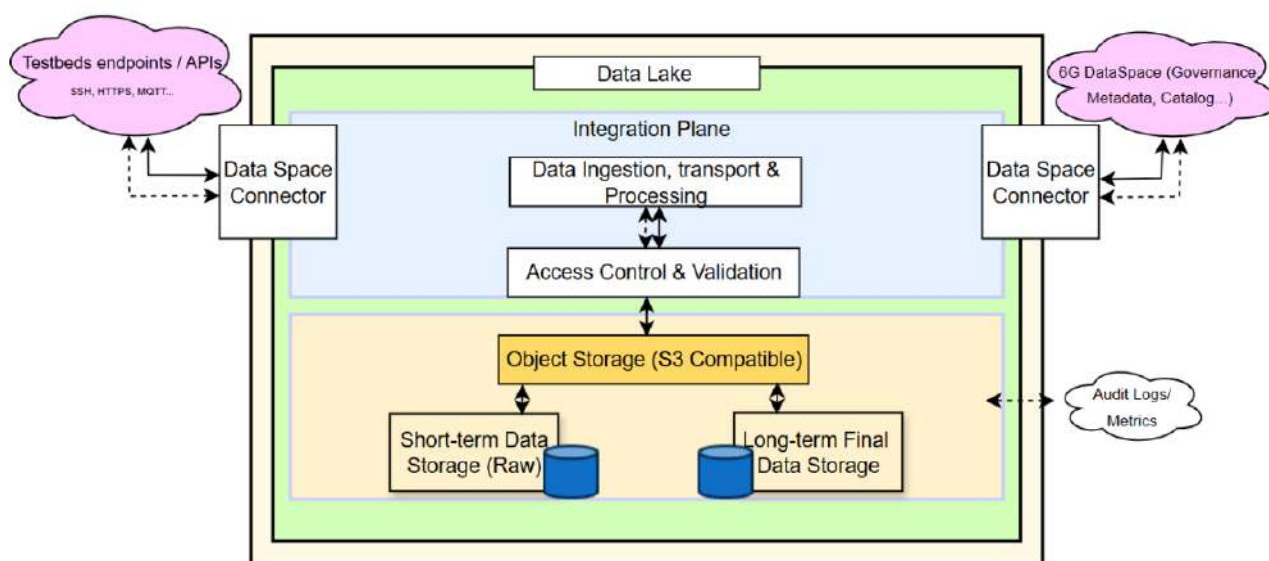


Figure 18: 6G Data Lake Storage High Level Architecture

Figure 18 presents the high-level architecture of the **6G Data Lake Storage Architecture** operated by Telefónica ID (TID). It represents the technical infrastructure layer responsible for secure data

transport, validation, and storage within the overall 6G architecture. This block acts as the entry and exit point for data coming from testbeds, the Digital Twin environment, and AI pipelines through standard interfaces (e.g., HTTPS, MQTT, S3, streaming protocols). Incoming data first passes through the Integration Plane, where ingestion and transport are handled, and where access requests are validated via the Control Plane before any storage operation is permitted. Once authorized, data is persisted in an S3-compatible object storage structured into short-term (raw) and long-term (final) repositories. Importantly, only the data payload is stored in this layer, while metadata and governance-related information are forwarded to the 6G Dataspace layer for cataloguing and policy management. Outbound data follows the reverse path: authorized consumers retrieve objects through the Integration Plane after access validation. Additionally, the system generates audit logs and operational metrics that are exported to monitoring and security platforms, ensuring traceability, compliance, and observability. This design maintains a strict separation between data transport/storage (handled by TID) and governance/catalog functions (handled by the 6G Dataspace), avoiding functional overlap while ensuring scalability, security, and architectural consistency.

4.7.2 Short-term Raw Data Storage

The Short-term Raw Data Storage component is responsible for the ingestion and temporary storage of data generated directly by experiments executed on physical testbeds and the DTT. This storage domain is optimized for high-throughput, write-intensive workloads and supports the automated ingestion of large volumes of heterogeneous data produced during experimental campaigns. Typical data stored in this component includes raw measurements, network traces, logs, intermediate results, and simulation outputs, represented in formats such as CSV, JSON, Parquet, binary dumps, and time-series exports. Data ingestion into this storage domain is fully automated and triggered by experiment lifecycle events, with datasets being transferred to the operated Data Lake through Data Space Connectors without manual intervention. Data stored in the short-term raw storage is retained for a limited and configurable period, sufficient to support downstream DataOps activities such as validation, cleaning, transformation, and enrichment. After processing, selected datasets may be promoted to the long-term final storage, while obsolete or redundant raw data may be discarded according to predefined retention policies. This approach aligns with the ELT paradigm adopted by 6G-DALI and supports efficient handling of high-frequency experimental data.

4.7.3 Long-term Final Data Storage

The Long-term Final Data Storage component provides persistent, durable storage for curated and validated datasets that are intended for reuse across the 6G-DALI framework. These datasets are typically produced by DataOps pipelines and represent **stable data assets** suitable for AI/ML training, validation, benchmarking, and reproducibility of experiments. This storage domain is optimized for read-heavy access patterns and supports logical dataset versioning, enabling experimenters to track dataset evolution and reproduce results over time. Datasets stored in this component constitute the primary data assets published in the 6G Data Space, where they are enriched with DCAT-AP-compliant metadata and made discoverable through the federated catalogue. The added value introduced by 6G-DALI lies in the seamless integration of this storage domain with DataOps and MLOps workflows, allowing finalized datasets to be accessed programmatically by automated pipelines without data duplication or manual coordination.

4.7.4 Data Space Connector

The Data Space Connector is the integration component that enables secure and automated interaction between the 6G Data Lake, the 6G Data Space, and the distributed testbeds. Deployed as part of the Adaptation Layer, the connector orchestrates the transfer of datasets generated by experiments into the operated Data Lake while ensuring that corresponding metadata is registered in the 6G Data Space. The

connector builds on the Eclipse Data Space Components framework and adheres to Gaia-X and IDSA specifications for sovereign and interoperable data exchange. In 6G-DALI, this baseline functionality is extended through experiment-driven automation: connectors are triggered directly by experiment completion events, automatically detect newly generated datasets, verify data integrity, and initiate coordinated data and metadata publication workflows. By embedding the Data Space Connector into the experimental lifecycle, 6G-DALI enables near real-time availability of experimental data for DataOps and MLOps, supporting high-throughput experimentation scenarios and eliminating manual data publishing bottlenecks.

5 Architectural Process Perspective

This section presents the architectural process perspectives of the 6G-DALI framework. Building on the detailed architectures described in section 4, it illustrates how the architectural components interact through a set of end-to-end representative workflows. These process perspectives provide a system-level view of the execution of key functionalities and highlight how the 6G-DALI architecture supports the project objectives in operational scenarios.

5.1 End-to-end DataOps Workflows

This section presents the end-to-end DataOps workflows supported by the 6G-DALI architecture, illustrating how the DataOps components interact with the 6G Data Space, 6G Data Lake and testbeds to manage the complete data lifecycle. The workflows demonstrate how the experimenters can discover, generate and process datasets through intent-driven requests, leveraging both existing datasets and on-demand experimentation across distributed 6G testbeds. Each workflow is described through a sequence diagram that details the involved components and the step-by-step interactions enabling automated, interoperable data operations.

5.1.1 Cold Data Search and Retrieval

This workflow describes the process through which an experimenter requests access to an existing dataset that is already available within the **6G Data Space**. Figure 19 illustrates the sequence of interactions. As a first step, the **experimenter** interacts with the framework through the **chatbot user interface** and submits a data request expressed in natural language. This request is translated by the **data search component** into a search query compatible with the 6G data space and forwarded to the **discovery services** component. Following authorization and policy enforcement, the relevant dataset is identified in the federated catalogue based on metadata matching. The corresponding dataset information, including metadata and its storage location in the **6G data lake**, is then forwarded to the data search component and ultimately presented to the experimenter through the chatbot interface.

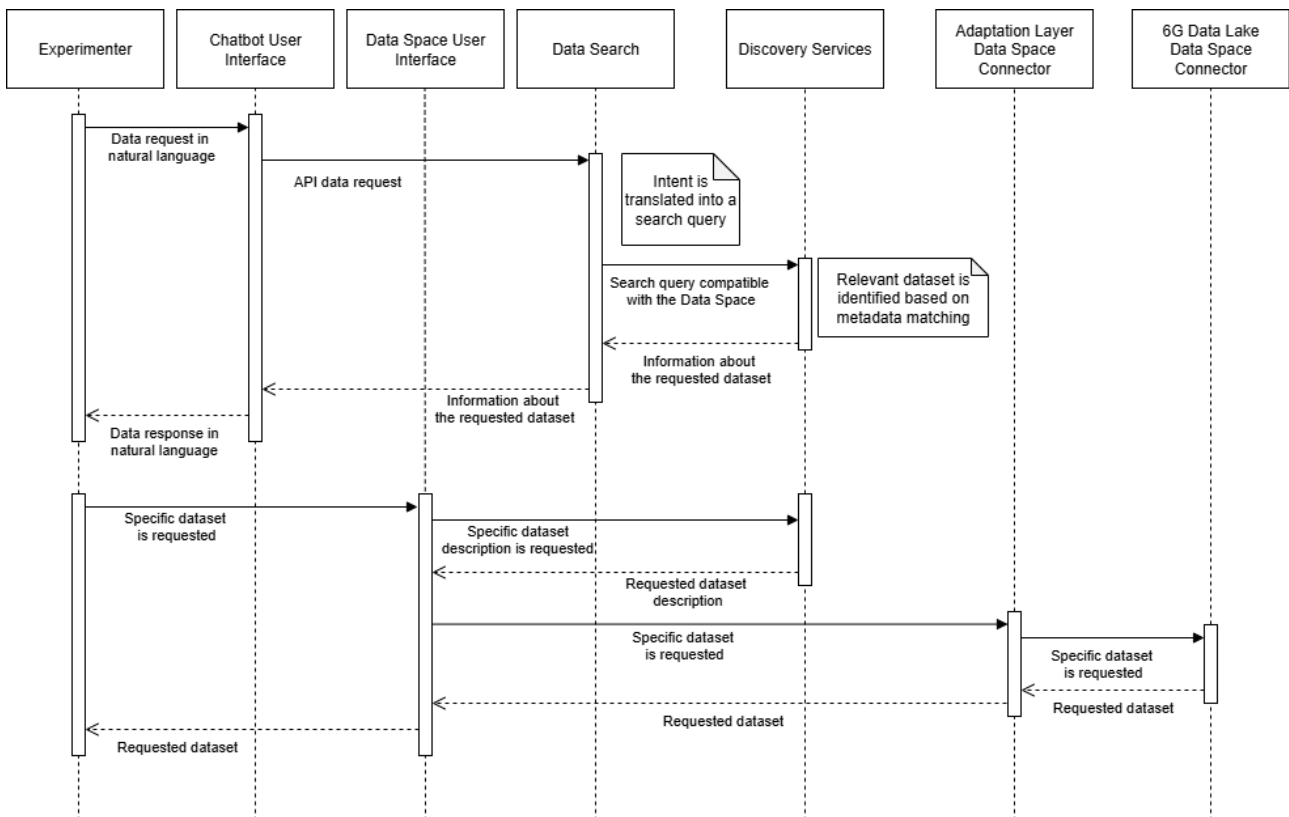


Figure 19: DataOps workflow for cold data search and retrieval

Subsequently, the experimenter requests access to the selected dataset via the **data space user interface**. Internally, the request is handled by the discovery services component, which retrieves the dataset metadata, and the requested dataset is extracted from the 6G data lake through the **data space connectors** implemented in the adaptation layer and the 6G data lake. The dataset is then exposed to the experimenter for further use. This workflow highlights the role of the 6G data space in enabling secure, sovereign, and interoperable access to historical (“cold”) data.

5.1.2 Hot Data Generation via Chatbot

This workflow describes the process through which an experimenter requests a dataset that is not currently available in the **6G Data Space** and therefore must be generated on demand. Figure 20 illustrates the sequence of interactions. Similar to the cold data search workflow, the experimenter interacts with the 6G-DALI framework through the **chatbot user interface** and expresses the data request in natural language. The **data search component** translates the user intent into a search query compatible with the 6G data space and forwards it to the **discovery services** component.

In this scenario, no relevant dataset is identified through metadata matching, and the requested dataset must be generated dynamically. To this end, the **experimenter** requests a data generation experiment in natural language, and the **data generation component** translates the user intent into an experiment request and applies a selection algorithm to identify the most suitable testbed for executing it. The experiment request is then forwarded to the selected testbed through the **adaptation layer**.

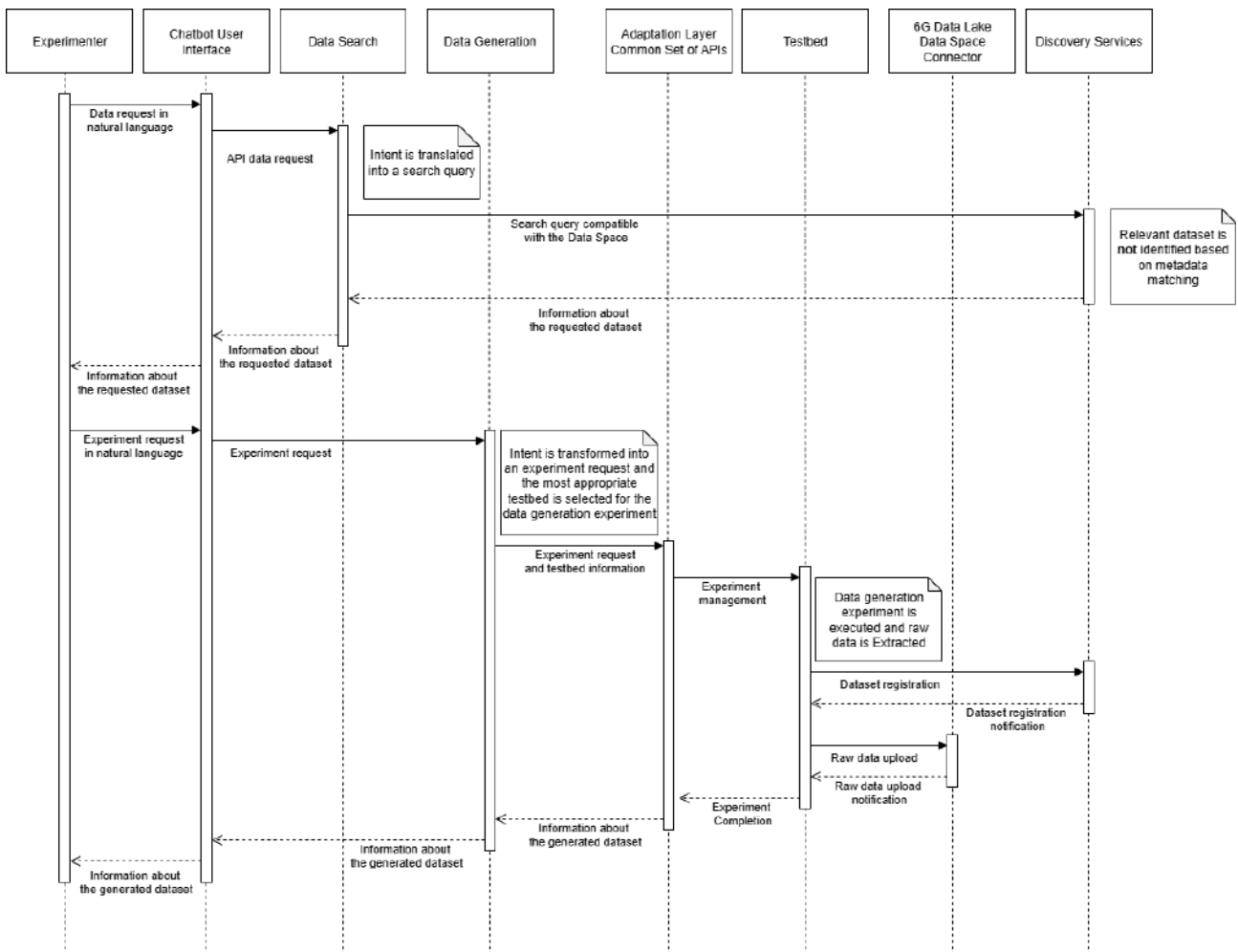


Figure 20: DataOps workflow for hot data generation via chatbot

The **testbed** executes the data generation experiment, and the resulting raw data is extracted and loaded into the **6G data lake** via the **data space connectors**, while metadata describing the newly

generated dataset is registered in the data space and exposed to the experimenter through the chatbot interface. This workflow demonstrates the dynamic generation of “hot” data across distributed 6G testbeds and their seamless integration into the 6G-Dali data space.

5.1.3 Hot Data Generation via Experimentation on Demand

This workflow describes the process through which an experimenter requests a data generation experiment using the DataOps APIs interface. Figure 21 illustrates the sequence of interactions. Similar to the previous workflow, the experimenter interacts with the 6G-DALI framework through the **dataops APIs** and expresses the data request in terms of characteristics of the dataset and testbeds to be used. Alternatively, users can decide to have the dataset generated via a surrogate model, instead of experiments on an actual testbed, with a user defined minimum confidence (i.e., maximum error with respect to the real data). By using a surrogate model, users can generate data quicker and with lower testbed resources utilization.

The **experimentation on demand (EoD) component** translates the user request into an experiment request and applies a selection algorithm to identify the most suitable testbed for executing it. The EoD component will then identify the experiments to be executed and forward the requests to the appropriate testbed via the selected testbed through the **adaptation layer**. The **testbed** executes the data generation experiment, and the resulting raw data is extracted and loaded into the **6G data lake** via the **data space connectors**, while metadata describing the newly generated dataset is registered in the data space and exposed to the experimenter through the dataops interface.

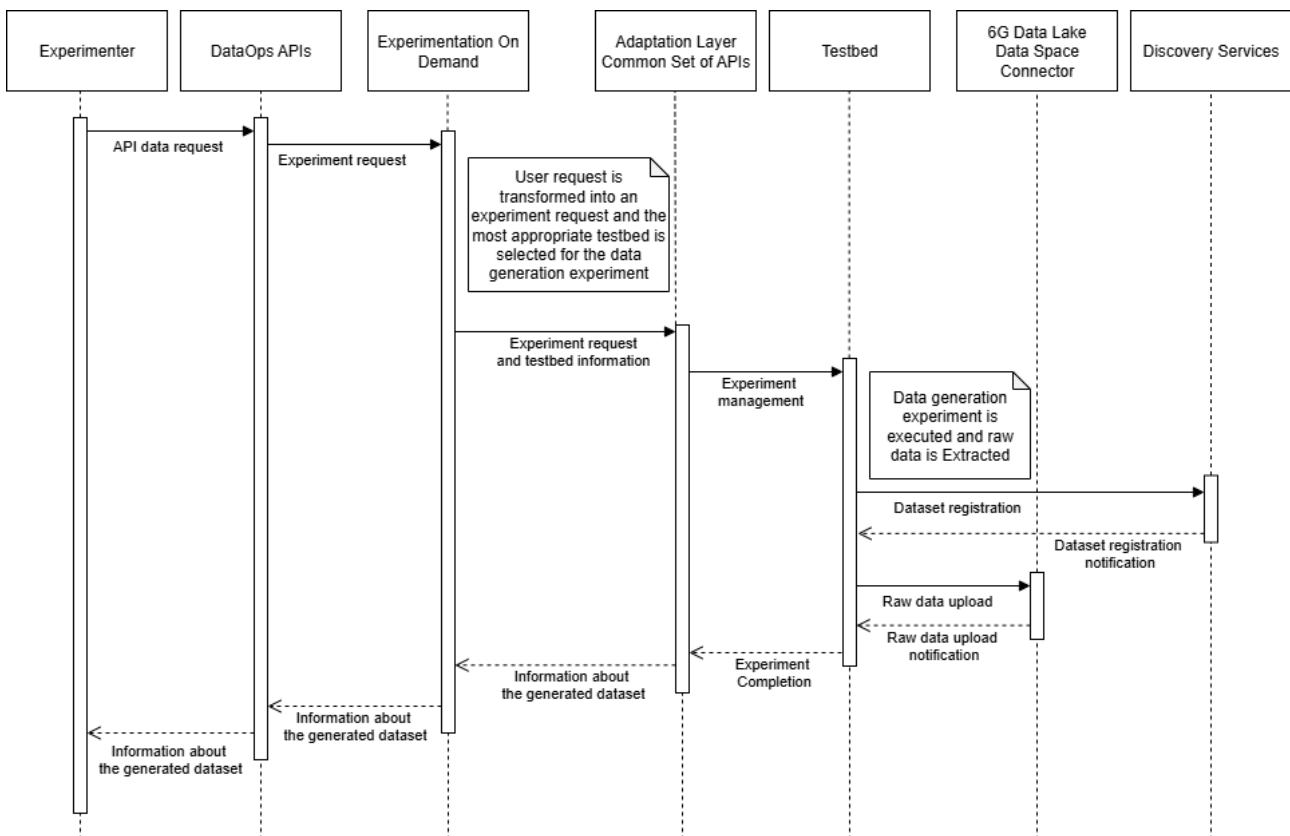


Figure 21: DataOps workflow for hot data generation via experimentation on demand

5.1.4 Data Transformation and Augmentation Services

This workflow describes the process through which an experimenter requests the application of data transformation and/or augmentation algorithms to one or more existing datasets. Figure 22 illustrates the sequence of interactions. The **experimenter** interacts with the 6G-DALI framework through the **dataops user interface** to express their request by selecting the desired data transformation or augmentation algorithm and the dataset to which it will be applied.

The **dataops orchestrator** processes this request and assigns it to the **dataops tools stack**. The dataops tools, as a first step, retrieve the selected dataset by querying the discovery services component and fetching the data from the **6G data lake** via the **data space connectors**. Subsequently, the processing pipeline is executed, and the selected algorithms are applied to the dataset. Once the transformation is completed, the curated dataset is stored in the **6G data lake** through the data space connectors, and its corresponding metadata are registered in the **6G data space catalogue** via the discovery services component. Finally, metadata describing the transformed dataset, together with its storage location, are exposed to the experimenter through the dataops user interface. This workflow highlights the flexibility of the 6G Dali DataOps framework in refining and adopting datasets to meet specific AI/ML experimentation requirements.

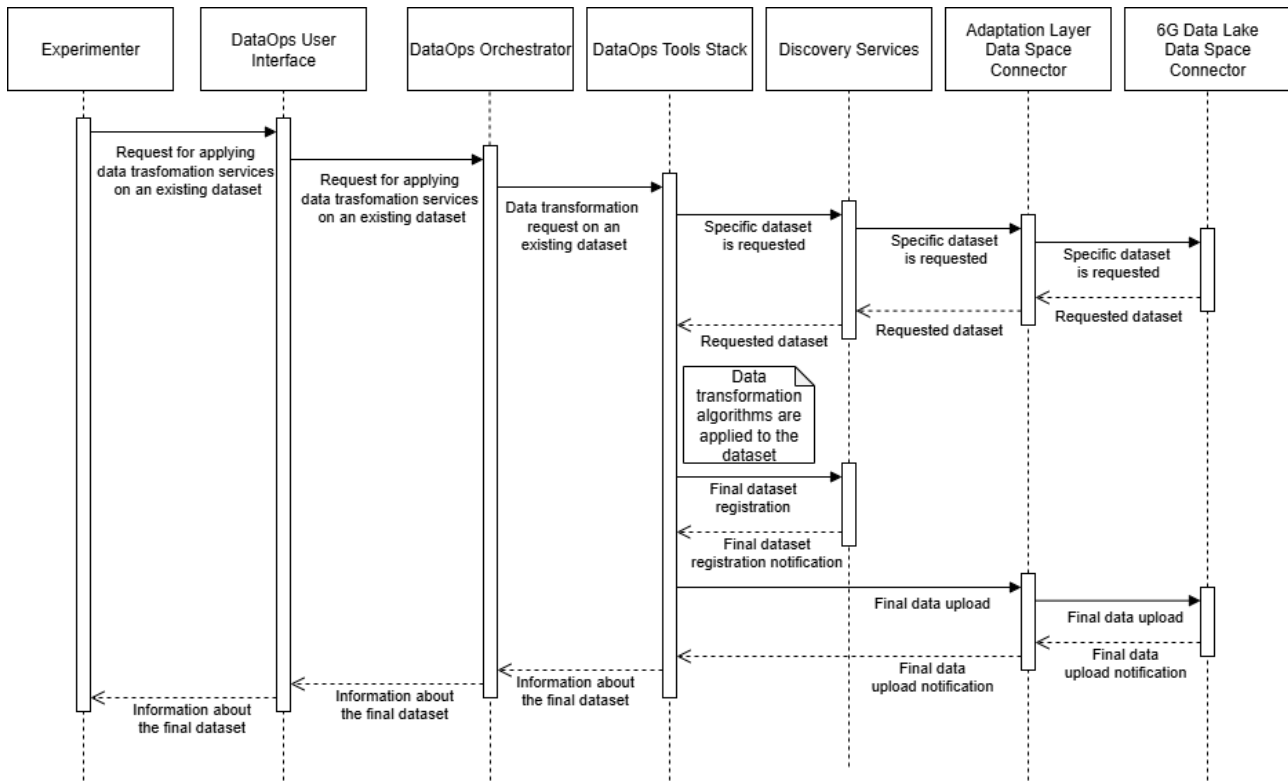


Figure 22: DataOps workflow for data transformation services

5.2 End-to-end MLOps Workflow

This workflow, detailed in Figure 23, describes the process through which an experimenter requests the execution of an MLOps task within the 6G-DALI ecosystem. As a first step, the experimenter interacts with the framework through the MLOps User Interface to submit task request. This request is forwarded to the MLOps AaaS API, which serves as the entry point to the management layer. Internally, the Services Manager processes this request by generating a MLProcess entity and assigning a unique Process ID. This ID is immediately returned to the experimenter, enabling an asynchronous workflow where the user can track progress without blocking the interface. Subsequently, the Services Manager passes the tool-agnostic process description to the Meta-Orchestrator. The Meta-Orchestrator analyses the task requirements (e.g., compute resources, location, or specific tool needs) to identify and select the most suitable Tools Stack, whether hosted in the 6G-DALI centralized facility or a distributed testbed. Once selected, the task is injected into the chosen stack for implementation. If the task requires data, the Tool Stack utilizes an internal Data Space Adapter to securely retrieve the necessary datasets from the 6G-DALI Data Space. Following data ingestion, the specific MLOps tools perform the requested task. Upon completion, the results are sent back to the Meta-Orchestrator, which triggers a notification to the Services Manager. This final step ensures that the MLReport and the related MLEntities are updated, allowing the experimenter to ultimately retrieve the results and performance metrics via the AaaS API.

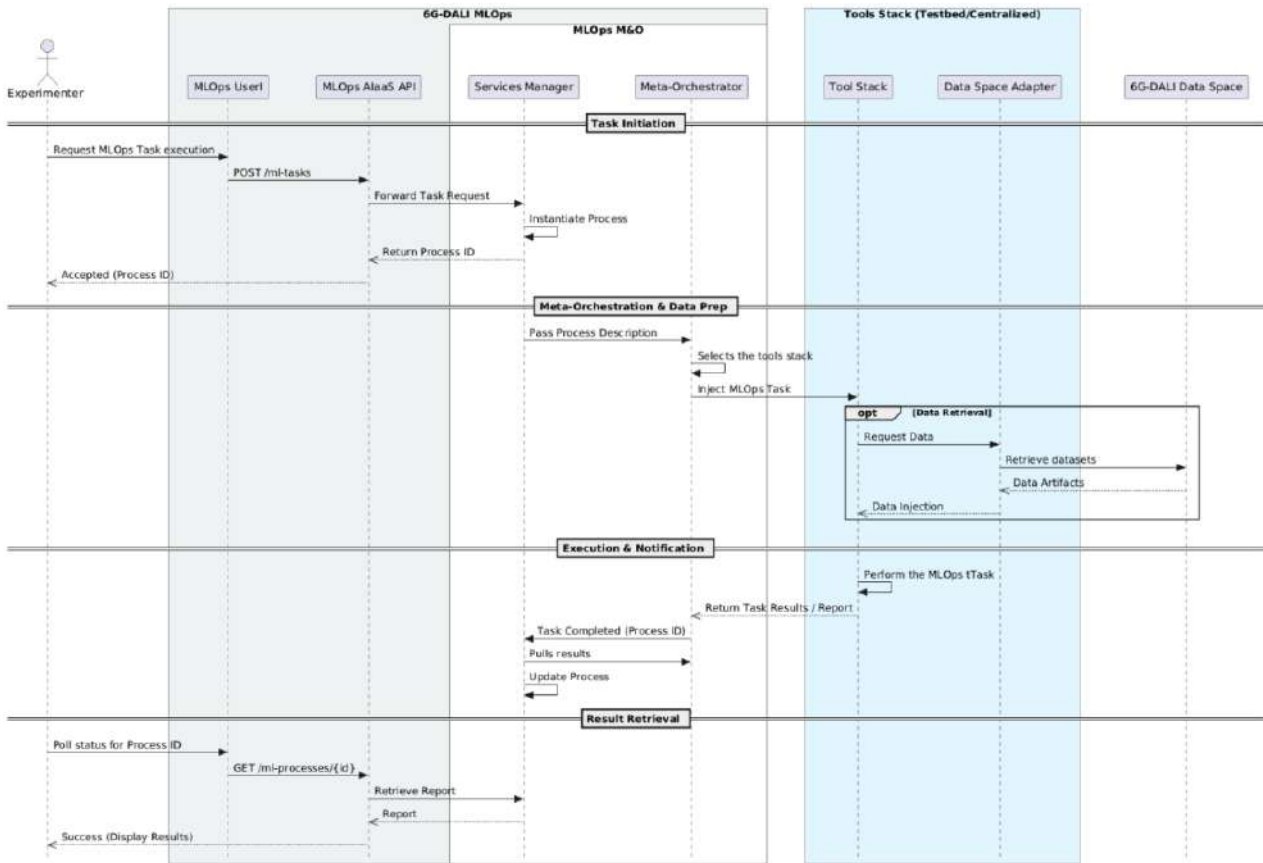


Figure 23: MLOps E2E Workflow sequence diagram

5.3 End-to-end Digital Twin Workflow

5.3.1 Data Generation Workflow

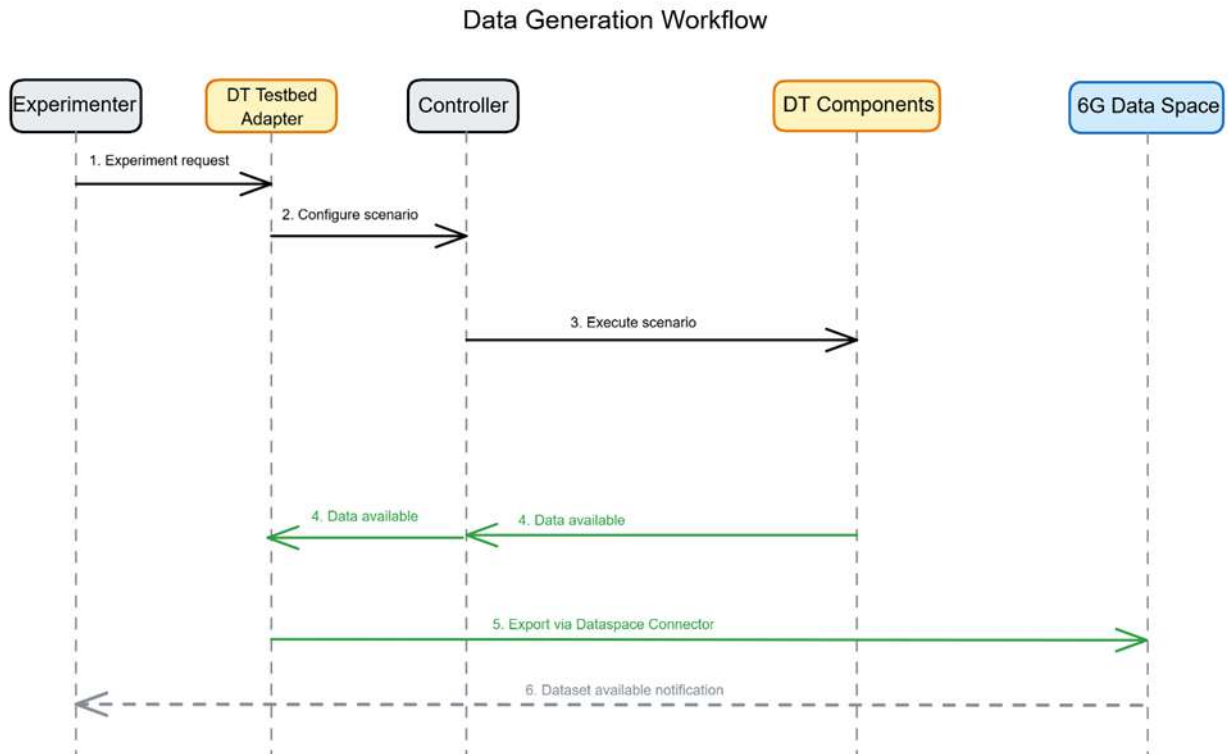


Figure 24: Data generation workflow on DTT.

This workflow describes the process through which an experimenter requests the generation of synthetic RAN datasets for AI/ML training purposes. Figure 24 illustrates the sequence of interactions. The experimenter interacts with the 6G-DALI framework, expressing the request, and specifying scenario parameters, simulation duration, and required KPIs. The DT Testbed Adapter processes this request and translates it into configuration commands for the Controller. The Controller launches the simulation on the RAN Digital Twin via the ADK. Once data becomes available, it is forwarded to the DT Testbed Adapter. The collected dataset is then exported to the 6G-DALI Data Space via the Data Space Connector, and the user will be notified once the data is available at the Data Space.

5.3.2 RLOps Validation Workflow

This workflow describes the process through which an experimenter validate reinforcement learning algorithms within the DTT before deployment to production networks. Figure 25 illustrates the sequence of interactions. The experimenter expresses intent to validate an RL algorithm. The RLOps component interacts with the Controller to express the validation request to test an RL model. The Controller processes this request and runs the validation on the DT. This process involves continuous pacing of the DT; running the simulation, pausing, extracting metrics, populating the state, calculating the reward, and executing actions suggested by the RL agent. This will be done on repeat until the validation concludes. Once the validation episodes complete, the results are returned to RLOps for analysis and for potential model refinement if needed.

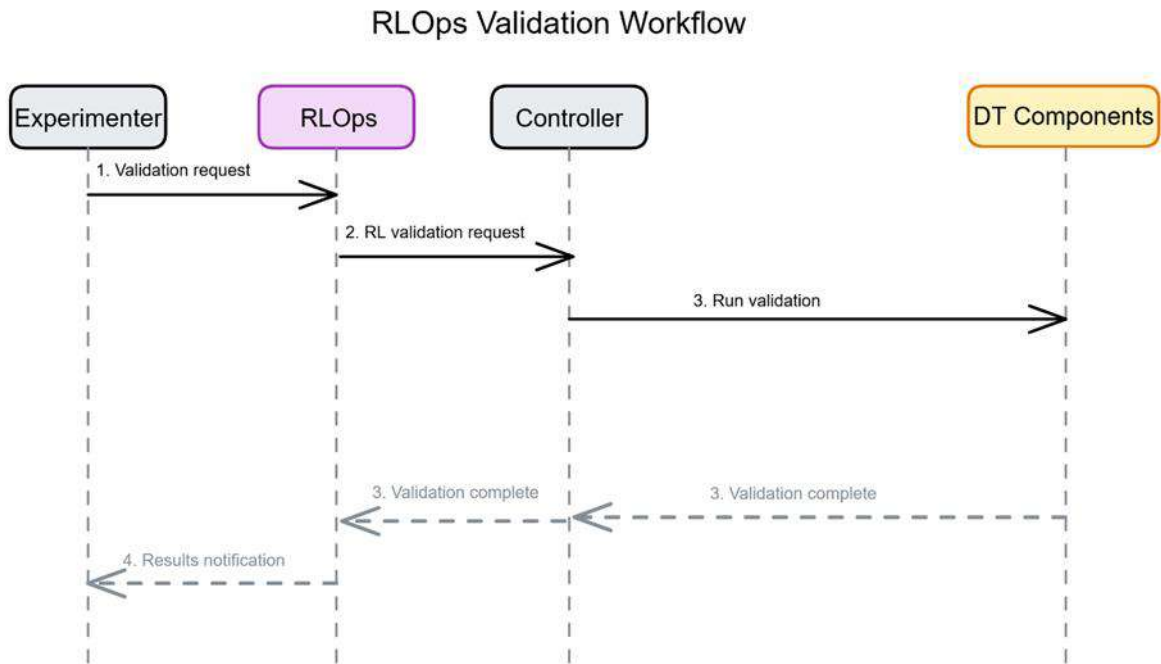


Figure 25: RLOps Validation Workflow on DTT

6 Conclusions

This deliverable contributed the initial architectural design of the 6G-DALI framework, providing a comprehensive and structured view of how DataOps, MLOps, Data Space, and Infrastructure components collectively enable end-to-end AI experimentation for 6G systems. Building on the project vision, objectives, requirements, and state-of-the-art, this deliverable introduced the high-level architecture and elaborated its detailed logical realization, offering a clear blueprint for implementing an interoperable and trustworthy AI framework tailored to 6G testbeds.

The work conducted in this deliverable focused on three main outcomes. First, it defined the high-level architecture and main building blocks of 6G-DALI, ensuring alignment with user requirements, relevant research initiatives, and standards. Second, it provided detailed logical architectures for each major architectural block (MLOps, DataOps, 6G Data Space, and Infrastructure) clarifying their internal components, responsibilities, and interfaces. Third, it complemented the static architectural view with architectural process perspectives, illustrating how the proposed architecture supports end-to-end workflows for data discovery, data generation, data transformation, and AI model lifecycle management in realistic experimentation scenarios.

In conclusion, this architectural design establishes a solid foundation for subsequent implementation, integration, and validation phases of the 6G-DALI project. As the framework is progressively instantiated, the architecture will be further refined to incorporate feedback from experimentation and performance evaluation. Future work will focus on realizing the defined components and interfaces, as well as on enhancing automation and orchestration intelligence, to support scalable and reliable AI-driven experimentation for 6G systems.

7 References

- [1] SUNRISE-6G_D2.2, "Initial SUNRISE-6G overall architecture" : Available at: https://sunrise6g.eu/wp-content/uploads/2025/07/SUNRISE-6G_D2.2-Initial-SUNRISE-6G-overall-architecture_IQU.pdf
- [2] 6G-BRICKS_D2.4, "Final 6G-BRICKS overall architecture design": Available at: [D2.4 Final-6G-BRICKS-overall-architecture-design-v1.0-to-submit FINAL.pdf](#)
- [3] 6G-SANDBOX_D2.1, "Ecosystem analysis and 6G-SANDBOX facility design": Available at: [6G-SANDBOX](#)
- [4] ADROIT6G_D4.2, "AI-driven Management & Orchestration and Control frameworks". Available at: [ADROIT6G-D4.2](#).
- [5] 3GPP TS 23.482 v19.3.0 (2025-09), "Functional architecture and information flows for AIML Enablement Service", available at: [TS23.482-v19.3.0](#)
- [6] 3GPP TS 28.105 V19.3.0 (2025-09), "Artificial Intelligence / Machine Learning (AI/ML) management", available at: [TS28.105-v19.3.0](#)
- [7] Gaia-X European Association for Data and Cloud AISBL, "Gaia-X Architecture Document," Release 22.10, October 2022. Available: <https://docs.gaia-x.eu/>
- [8] Publications Office of the European Union, "DCAT Application Profile for data portals in Europe," Version 3.0.1, 2024. Available: <https://joinup.ec.europa.eu/collection/semic-support-centre/solution/dcat-application-profile-data-portals-europe>
- [9] International Data Spaces Association (IDSA), "Reference Architecture Model," Version 4.0, 2022. Available: <https://docs.internationaldataspaces.org/>
- [10] Eclipse Foundation, "Eclipse Dataspace Components Documentation," 2024. Available: <https://eclipse-edc.github.io/docs/>
- [11] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, ... & T. Schneider, (2022). {FLAME}: Taming backdoors in federated learning. In 31st USENIX Security Symposium (USENIX Security 22), pp. 1415-1432.
- [12] J. Hamer, M. Mohri, & A. T. Suresh, A. T. "Fedboost: A communication-efficient algorithm for federated learning", in *International Conference on Machine Learning*, pp. 3973-3983, 2020, November. PMLR.
- [13] H. Daga, P. K. Nicholson, A. Gavrilovska, and D. Lugones, "Cartel: A System for Collaborative Transfer Learning at the Edge", in *Proceedings of the ACM Symposium on Cloud Computing (SoCC '19)*, 2019, Association for Computing Machinery, New York, NY, USA, 25–37. <https://doi.org/10.1145/3357223.3362708>
- [14] M. Zaharia et al., "Accelerating the Machine Learning Lifecycle with MLflow," in *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018.
- [15] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," 2018, arXiv:1807.05118. [Online]. Available: <https://arxiv.org/abs/1807.05118>
- [16] E. Liang et al., "RLlib: Abstractions for distributed reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn.*, J. Dy and A. Krause, Eds., vol. 80, Jul. 10–15, 2018, pp. 3053–3062. [Online]. Available: <https://proceedings.mlr.press/v80/liang18b.html>
- [17] E. Keramidi, P. Zervou, J. Lakoumentas, K. Ramantas and C. Verikoukis, "Towards the Uncertainty Quantification of AI-Models in 6G Networks," 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 2024, pp. 1-6.
- [18] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.
- [19] M. Ameer, B. Brik, and A. Ksentini, "Leveraging LLMs to Explain DRL Decisions for Transparent 6G Network Slicing," in 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), 2024, pp. 204-212.

- [20] F. Minucci, R. M. Noguera Oishi, H. Xiong, D. Verbruggen, C. Thys, R. Hersyandika, R. Beerten, A. Colpaert, V. Ranjbar and S. Pollin, "Building a Real-Time Physical Layer Labeled Data Logging Facility for 6G Research," 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Athens, Greece, 2024, pp. 1-7, doi: 10.1109/CAMAD62243.2024.10942742.
- [21] "Charmed Kubeflow documentation," Canonical Ltd. <https://documentation.ubuntu.com/charmed-kubeflow> (accessed Jan. 20, 2026).
- [22] "Kubernetes Documentation," Kubernetes. kubernetes.io (accessed Jan. 20, 2026).
- [23] "NI RF Data Recording API Documentation," 2026 NATIONAL INSTRUMENTS CORP. <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q0000019mApCAI&l=nl-BE> (accessed Jan. 20, 2026).
- [24] "Kubeflow Documentation," Kubeflow. www.kubeflow.org (accessed Jan. 20, 2026).
- [25] D. Merkel, "Docker: Lightweight Linux Containers for Consistent Development and Deployment," Linux J., Mar. 2014.
- [26] OpenStack 2026.1 Documentation," OpenStack. <https://docs.openstack.org/2026.1/> (accessed Jan. 20, 2026).
- [27] P. Moritz et al., "Ray: A distributed framework for emerging AI applications," in 13th USENIX symposium on operating systems design and implementation, 2018, pp. 561–577.
- [28] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
- [29] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," Advances in neural information processing systems 32, 2019.
- [30] F. Chollet et al., "Keras," Keras. <https://keras.io> (accessed Jan. 20, 2026).
- [31] "Prometheus Documentation," Prometheus. [Online]. Available: prometheus.io (accessed Jan. 20, 2026).
- [32] "Angular Documentation," Angular. <https://angular.dev> (accessed Jan. 20, 2026).
- [33] "Flask Documentation (v3.1.2)," Pallets Projects. flask.palletsprojects.com (accessed Jan. 20, 2026).
- [34] "FastAPI Documentation," FastAPI. <https://fastapi.tiangolo.com> (accessed Jan. 20, 2026).

Annex I: 6G-DALI Testbeds

This section describes the logical architecture of the participating 6G-DALI testbeds, which constitute the execution environment for data generation experiments. It provides an overview of the functional architecture of each testbed, i.e., KUL, ISI/ATH, and EUR, along with their local MLOps tool stacks, short-term raw data storage capabilities, and exposed interfaces.

I.1 KUL Testbed

I.1.1 Functional Architecture

The schematic architecture of the KU Leuven 6G-Bricks Testbed [20] is represented in Figure 26. It is a modular testbed containing multiple functionalities which can be set up according to the intended use. The main component everything is connected to is the KU Leuven Cluster. It contains the testbed controller and the gateway. The testbed controller runs all the signal processing, physical layer algorithms (including AI models), and controls the radios. The gateway allows for remote login and deployment of experiments on the server.

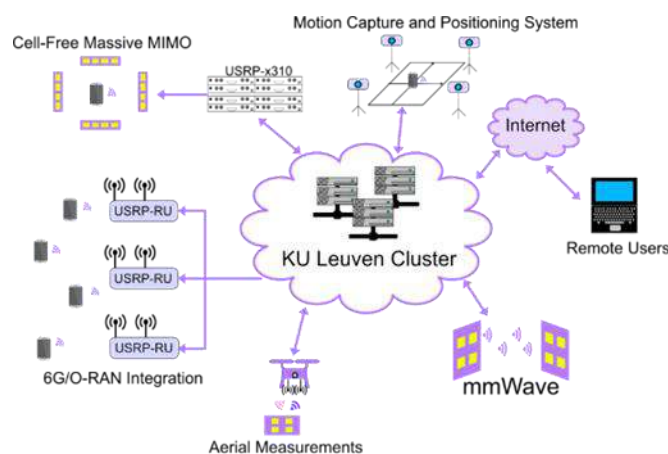


Figure 26: KU Leuven 6G-Bricks Testbed Schematic Representation

The modular parts of the testbed consist of a cell-free massive MIMO setup, a motion capture and positioning system, 6G/O-RAN integration, an mmWave setup, and an aerial measurements setup. The cell-free massive MIMO setup allows for a modular assembly of up to 64 antennas driven by 32 USRPs and up to 12 UE, each driven by a USRP. Synchronization of the antennas and the UE is performed via over-the-air coax cables. The setup supports a frequency range of 400 – 4400 MHz and a bandwidth of 40 – 120 MHz. An I/Q samples rate of 200 Msps can be achieved. The antennas are portable by means of a wheeled rack, and the antenna elements can be flexibly distributed. A custom FPGA image can be leveraged for smaller USRP to support a portable, even drone mounted, UE.

The motion capture and positioning system consists of a motion capture system and four positioners. The positioners cover a 1250 x 1250 mm area each, resulting in a total of 2.5 x 2.5 m. The motion capture system contains four cameras with 2 Mpixel, resulting in a resolution of 1824 x 1088. It captures IR video at 340 fps and resolves positions with 0.11 mm precision in 3D, with a maximum range of 15 m. Both positioners and motion capture system are fully integrated and synchronized with the cell-free massive MIMO setup.

The 6G/O-RAN integration consists of software support for O-RAN and three USRPs for 6G prototyping. The USRPs support a frequency range from 1 MHz to 7.2 GHz and can be tuned to extend to up to 8 GHz. The RUs have an instantaneous bandwidth capacity of up to 400 MHz. The setup supports an I/Q sample rate of 500 Msps and is synchronized via a clock distributor and a GPS time reference. Rubidium clocks can be deployed for higher accuracy.

The mmWave setup contains four 60 GHz RF transceivers and supports a MIMO application. Of the four transceivers, two consist of two 4x16 element-sized arrays, and two consist of an 8x8 element-sized

array with fast beam switching capability. The aerial measurements setup entails a drone equipped with a USRP and modular antennas and allows for aerial measurements outdoors.

I.1.2 Local MLOps Tools Stack

ML model training and inference on the testbed are enabled by Charmed Kubeflow [21], an open-source platform for MLOps on K8s [22].

I.1.3 Short-Term Raw Data Storage

No extra hardware is dedicated to short-term raw data storage. Data collected from experiments is stored internally until being moved by the experimenter to an external long-term storage.

I.1.4 Interfaces

User applications can access the hardware to trigger experiments as well as the RF Data Recording API [23] to access collected data using a custom-built REST API. Each entry point of the REST API is linked to a specific functionality of the testbed. APIs are described in Table 8. The RF Data Recording API is a Python library to perform automatic measurements with the USRPs and further gives access to the locally stored data. Finally, Kubeflow [24] provides a web-based user interface that serves as the central hub for managing ML workflows in the Local MLOps Tools Stack. Through this interface, users can create and monitor pipelines, manage notebooks, track experiments, access artifacts, and integrate with tools like MLflow.

Table 8: KU Leuven 6G-Bricks Testbed Interfaces, as depicted in Figure 7.

Interface ID	Interface Description	Involved Components
I_1	REST API decoupling user applications from underlying low-level libraries for triggering experiments and accessing collected data	User, RF Data Recording API
I_2	RF Data Recording API performing automatic measurements with the USRPs and accessing collected data	REST API, Short-Term Raw Data Storage
I_3	Kubeflow User Interface for managing ML workflows.	User, Local MLOps Tools Stack

I.2 ISI/ATH Testbed

I.2.1 Functional Architecture

The ATH testbed facility, located at the ISI premises in Greece, implements a comprehensive 5G/6G ecosystem comprising the Core Network (CN), Radio Access Network (RAN), and Cloud-Edge domains, with a focus on the Cognitive Compute Continuum. Figure 27 presents an overview of the testbed, with the Testbed-as-a-Service (TaaS) framework and computational components depicted on the left, and the Multi-access Edge Computing (MEC) and Radio components depicted on the right.

Regarding computational hardware specifications, the testbed comprises four physical servers with Intel Xeon and AMD Threadripper processors, totalling 112 physical cores and 704GB of RAM. The GPU infrastructure includes 2 NVIDIA H100 PCIe, 3 NVIDIA RTX 6000 Ada, and 1 RTX 4500 Ada, providing a combined 328GB of VRAM. The storage devices include 10.6TB of high-speed NVMe storage and 8TB of HDD capacity. Edge computing nodes utilize a combination of x86-based mini PCs and ARM-based Raspberry Pi 4 devices equipped with SIM8200EA-M2 5G HATs, and the utilized switch is an Nvidia Spectrum-2 SN3420. User Equipment (UE) devices include 2 OnePlus 8T 5G phones, 2 Samsung Galaxy S23+ phones, 1 Teltonika RUTX50 CPE, as well as the Raspberry Pi devices. The 5G RAN setup comprises 1 Amarisoft Callbox Mini and 1 Amarisoft Callbox Advanced. The CN runs both Open5GS and the Amarisoft 5G Core within a private cloud environment.

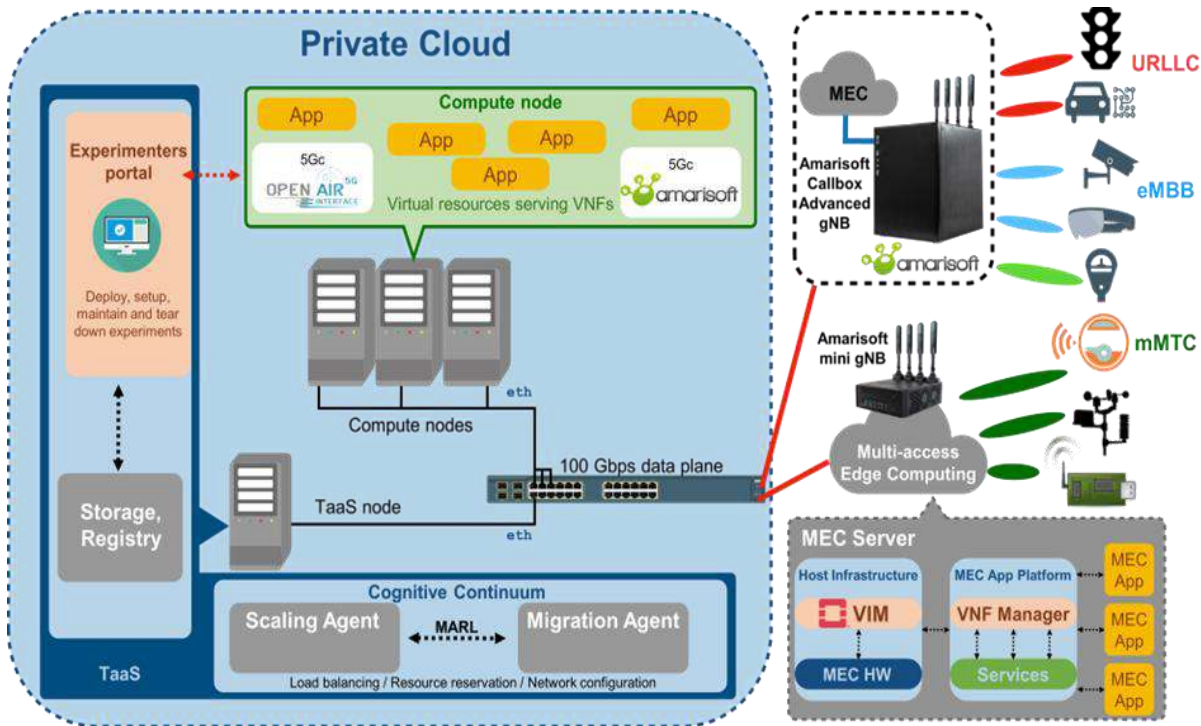


Figure 27: ATH 5G/6G testbed facility overview

Regarding the Cloud-Edge domain, the testbed leverages Kubernetes (k8s), Docker [25], and OpenStack [26] for managing containerized applications. A specialized abstraction framework acts as an enabler to facilitate the Cognitive Compute Continuum, allowing for AI-optimized management of computing, networking, and storage resources. This includes scaling and migration agents that use Multi-Agent Reinforcement Learning (MARL) for resource reservation and load balancing.

A workload orchestrator is included, which acts as a 6G enabler by dynamically managing the placement of next-generation network functions alongside intelligent edge applications. By utilizing a unified pool of hardware resources, the placement logic ensures that demanding 6G communication tasks and AI-driven services coexist efficiently on the same infrastructure, allowing smarter resource allocation and seamless scalability at the edge.

1.2.2 Local MLOps Tools Stack

The testbed supports an MLOps environment designed for automated AI model lifecycle management across the cloud-edge continuum. Currently, Mlflow is the primary tool used for experiment tracking, versioning, and managing the AI model lifecycle. Regarding MLOps experiment orchestration and workflows via k8s, Kubeflow is utilized as the experimentation scheduler and for setting up end-to-end MLOps pipelines. Further, RAY [27] can be employed for dynamic and efficient workload placement across nodes.

Support for frameworks like Scikit-Learn [28], PyTorch [29], and Keras [30] is integrated for ML model development, currently applied to tasks like real-time anomaly detection and intelligent resource orchestration. Prometheus and Grafana can be used to monitor models in production, tracking metrics such as inference latency and throughput.

Finally, Split Learning is also being actively developed as part of the MLOps stack of the testbed. A cluster of server and client nodes is managed by k8s, while Split Learning algorithms are being applied to efficiently split ML model training and inference across these server/client nodes.

I.2.3 Short-Term Raw Data Storage

The ISI/ATH testbed utilizes a multi-tiered approach to storing raw generated data depending on the data type and experiment requirements. As most data collected from ML or data generation experiments at the cloud-edge k8s cluster are in the form of time-series, Prometheus [31] serves as the primary storage engine for these metrics. This includes network performance (throughput, latency), system resource utilization (CPU/RAM per pod), and k8s control plane metrics. Regarding experiment definition artifacts and metadata information, a dedicated model and data registry node is used within the TaaS framework to store container images and experimental data dumps. Finally, after being collected, raw logs, configuration files (YAML), and exported datasets (CSV, JSON) are stored locally on the high-performance NVMe/SSD storage of the compute nodes, accessible via direct SSH for post-experiment analysis.

I.2.4 Interfaces

The ISI/ATH testbed provides several interfaces for both experimenters and system-to-system communication. The Experimenter Portal, initially developed within the 6G-BRICKS and SUNRISE-6G projects, serves as a web-based entry point (built with an Angular [32] frontend/Python Flask [32] backend) for user authentication, intent submission, and experiment triggering. Regarding Northbound APIs, the PromQL HTTP API is used for querying time-series data from the monitoring stack, while Kubernetes Metrics APIs provide data related to pod placement and resource consumption. Also, Istio is integrated to export network telemetry, such as latencies, traffic flows, and service-level observability. Additionally, Flask and FastAPI [34] [33] interfaces are used for ML data ingestion, model training, and custom script onboarding or exposure. Besides the currently available APIs, for more complex integration or custom experimentation, Secure Shell (SSH) remains the primary method for direct infrastructure access, onboarding experiments, and retrieving data dumps.

I.3 EUR Testbed

I.3.1 Functional Architecture

The 5G facility at EUR, illustrated in Figure 28, offers experimental 5G services encompassing enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC). Built entirely on open-source technologies and an open-architecture design, the facility enables the onboarding of new network functions into an operational 5G infrastructure and supports their evaluation in both controlled laboratory environments and deployed live networks.

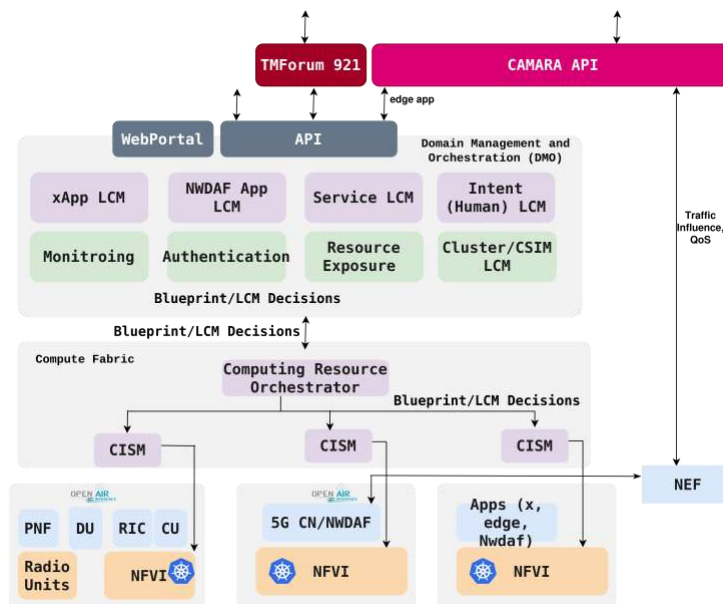


Figure 28: EUR Testbed architecture

From a technical perspective, the facility provides 5G connectivity based on OpenAirInterface (OAI). It supports the deployment of cloud-native applications and services in the form of Virtual Network Functions (VNFs) and Container Network Functions (CNFs). The management and orchestration framework comply with ETSI NFV and MEC specifications, enabling the management of virtualized resources across both cloud and edge environments. The NFV Orchestrator (NFVO) and MEC Orchestrators are in-house developments that manage the lifecycle of network services using Network Service Descriptors (NSDs), which are composed of cloud-native functions described through Application Descriptors (AppDs). These functions are deployed on container management platforms such as OpenShift and Kubernetes.

The 5G New Radio (NR) connectivity is fully supported by OAI-based Radio Access Network (RAN) and Core Network (CN) components. The RAN supports both FR1 and FR2 frequency ranges. In FR1 deployments, the RAN is disaggregated into Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU), following two functional split options. Split 7.2 uses OAI CU and DU combined with commercial RUs from vendors such as VVDN, LiteON, and Benetel for indoor and outdoor deployments. Split 8 employs OAI CU and DU with USRP platforms (B210, N300, X310) for indoor use, as well as commercial RUs from AW2S for both indoor and outdoor scenarios.

The FR1 RAN operates in bands 38 (2.6 GHz) and 78 (3.4 GHz), supporting subcarrier spacings of 15 and 30 kHz, bandwidths ranging from 10 to 100 MHz, and multiple Bandwidth Parts (BWPs), including initial and dedicated BWPs. It supports various Time Division Duplex (TDD) configurations, allowing asymmetric uplink and downlink resource allocation with short TDD periods down to 2.5 ms. Additionally, the system implements 4-layer downlink and 2-layer uplink MIMO, with support for 256-QAM modulation.

For FR2 deployments, the facility currently supports a commercial Radio Unit from LiteON operating at 27 GHz. To complete the O-RAN architecture, the facility integrates a Real-Time Intelligent Controller (RIC), namely OAI FlexRIC, enabling the execution of xApps for RAN control, management, and optimization.

The 5G Core Network is based on the OAI Core and implements a partial 3GPP 5GC service-based architecture. It includes key network functions such as NRF, AMF, SMF, UPF, UDM, UDR, AUSF, NSSF, PCF, NEF, NWDAF, and CAPIF. The NEF and CAPIF components are used to expose core network capabilities to vertical applications.

The EUR testbed supports the collection of the following KPIs during experiments, as shown in Table 9.

Table 9: KPIs provided by EUR testbed

Domain	KPIs
RAN	RSRP, RSRQ, CQI, PUSCH SNR, PUCCH SNR, UL/DL data rate, UL BLER, DL BLER, and RLC queuing latency
CORE	The number of attach requests over a given period, PDU session data rate, number of failed PDU sessions, and number of connectivity loss events
EDGE	VNF CPU and RAM usage, VNF data rate, and estimated VNF energy consumption

1.3.2 Local MLOps Tools Stack

The EUR MLOps testbed is designed to provide robust support for small- to medium-scale local MLOps experimentation, enabling end-to-end development, evaluation, and operationalisation of machine learning models in realistic network environments. A key distinguishing capability of the testbed is its tight integration with the Network Data Analytics Function (NWDAF) and a fully operational 5G Core Network based on the OpenAirInterface (OAI), RAN and edge software stack.

This integration enables direct access to real-time, high-fidelity telemetry data originating from heterogeneous network components. Such access is critical for training, validating, and evaluating ML models under real operational conditions. By ingesting live network data, the testbed ensures that both training and inference workflows accurately reflect real-world network dynamics, significantly enhancing model robustness and relevance.

The availability of real-time network data is particularly valuable for use cases such as anomaly detection, traffic forecasting, and network optimisation, where rapidly changing network behaviour is a defining characteristic.

EUR's MLOps platform follows a cloud-native, microservice-based architecture, with Kubernetes serving as the orchestration layer. This architectural choice provides modularity, scalability, and resilience, which are essential for managing complex ML pipelines and supporting iterative experimentation.

For experiment tracking and model lifecycle management, the testbed leverages MLflow v2.9.3, with a strong focus on its Model Registry and experiment tracking capabilities. MLflow is deployed as a service within the Kubernetes cluster, offering a centralised interface for managing experiments, model versions, and deployments. The platform is configured to automatically capture and log key metadata during experiments, including:

- Model hyperparameters
- Evaluation metrics (e.g., accuracy, loss, latency)
- System resource utilisation
- Environment and dependency versions

These capabilities enable real-time observability, systematic comparison of experimental runs, and full reproducibility across the ML lifecycle.

MinIO is used as the local object storage backend for persisting model artefacts and datasets generated during training and evaluation. This ensures consistent, scalable, and accessible storage of all experiment-related assets. In parallel, PostgreSQL serves as the metadata store for MLflow, maintaining structured records of experiments, metrics, and model versions, which are essential for traceability and long-term experiment management.

To support automation and external integration, the EUR MLOps testbed exposes a comprehensive set of RESTful APIs, enabling seamless interaction with all major stages of the AI pipeline, from model training and registration to deployment and inference consumption. These APIs facilitate integration with external platforms and support continuous experimentation and operational workflows.

I.3.3 Short-Term Raw Data Storage

For local short-term raw data storage within the EUR testbed, a MinIO-based object storage service is deployed. The storage is provisioned with a default capacity supporting raw datasets of up to 10 GB, with the ability to scale capacity on demand based on experimental requirements.

During experiment execution, data collected by the EUR testbed DataOps framework, materialized into one or more CSV data files, accompanied by a JSON formatted metadata file describing the dataset. These artifacts are exposed via the data space connector, which facilitates dataset registration and orchestrates the secure transfer of data to the 6G-DALI Data Lake.

As the EUR testbed does not provide persistent dataset hosting, the MinIO object storage is used strictly for transient data staging. All stored datasets are subject to a predefined retention policy and are automatically purged after the expiration period, after which they are no longer accessible.

I.3.4 Interfaces

The facility can be accessed through two primary interfaces: **a Web Portal and a set of exposed APIs**. The Web Portal is tailored to support the deployment, configuration, and management of vertical-

specific use cases, offering users an intuitive environment to design and operate end-to-end experiments. In parallel, the exposed APIs enable seamless integration with external orchestration and management components, allowing advanced users to automate workflows and incorporate the facility into larger system architectures.

In addition to its custom-developed APIs, the EUR testbed also supports standardized interfaces, including TM Forum Open API 921 and CAMARA-compliant APIs, ensuring interoperability and alignment with industry standards. Furthermore, the facility is interconnected with the broader European federation of testbeds through the SLICE-PP, 6G-BRICKS, and SUNRISE-6G projects, enabling cross-testbed experimentation and collaborative research at a European scale.