6GARROW

6G AI-Native Integrated RAN-Core Networks

Deliverable D5.1
Initial System Architecture of Validation and Experimental Testing

Co-funded by the European Union

6GSNS

IITP

This work (or 'part of this work') has been supported by the 6GARROW project which has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101192194 and from the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00435652).

Date of delivery:	31/07/2026	Version:	1
Project reference:	101192194	Call:	HORIZON-JU-SNS-2024-STREAM-B-01-06
Start date of project:	01/01/2025	Duration:	36 months

Document properties:

Document Number:	D5.1
Document Title:	6GARROW Initial Architectures of Validation and Experimental Testing
Editor(s):	Yuzhen Ke (FhG)
Authors:	Yuzhen Ke (FhG), Mehdi Heshmati (FhG), Zoran Utkovski (FhG), Hoon Lee (ETRI), Hewon Cho (ETRI), Mingun Kim (ETRI), Nicolas Cassiau (CEA)
Contractual Date of Delivery:	31/07/2026
Dissemination level:	PU (public)
Status:	Final
Version:	1.0
File Name:	6GARROW D5.1_v1.0

Revision History

Revision	Date	Issued by	Description
0.0	01.05.2026	6GARROW WP5	Template for Deliverable 5.1
0.1	23.07.2026	6GARROW WP5	Version for cross review
1.0	29.07.2026	6GARROW WP5	Final version for review

Abstract

The present document introduces the initial system architecture of the laboratory demonstrations developed within WP5 of the 6GARROW project. It defines four representative demonstrations for validating AI-native technologies across the device, RAN, and CN domains. The deliverable further establishes the relationship between the demonstrations and the 6GARROW functional architecture and exemplifies, based on one demonstrator, a unifying demonstration environment for collaborative edge inference. The architectural descriptions provide the foundation for the subsequent implementation, integration, and validation of the proposed AI/ML technologies.

Keywords

6G, laboratory demonstrations and testbeds, architecture, experimental validation

Disclaimer

Funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect the views of 6GARROW Consortium nor those of the European Union or Horizon Europe SNS JU. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

Deliverable D5.1 contributes to Objective 4 of the 6GARROW project: Develop and Validate Experimental Testbeds and Proof of Concept Platforms to showcase the key 6GARROW innovations. It presents the initial architectural design of the four laboratory demonstrations of Task 5.1. These demonstrations assess the proposed AI/ML concepts for 6G networks in a controlled environment. They comprise semantic-aware device–edge co-inference, cross-domain AI coordination across the radio access and core domains, AI/ML-enabled physical-layer processing, and CSI compression. For each demonstration, the objectives, logical architecture, physical/software architecture, interfaces, capabilities and validation targets are specified, establishing a consistent framework for implementation and experimental validation. Each design is grounded in the AI-native system architecture defined in D2.2 and revised in D2.3, and in the AI/ML methods under development in WP3 and WP4. The deliverable also introduces a unifying demonstration environment for edge intelligence applications based on Demonstration 1, mapped to the layers of the 6GARROW functional architecture. Experimental results obtained on these platforms will be reported in D5.3.

List of Tables

List of Figures

Acronyms and abbreviations

Term	Description
3GPP	3rd Generation Partnership Project
AI	Artificial Intelligence
AIaaS	Artificial Intelligence as a Service
BLER	BLock Error Rate
CN	Core Network
CSI	Channel State Information
C/U/S-Plane	C-Plane / U-Plane / Synchronization-Plane
CQI	Channel Quality Indicator
DU	Distributed Unit
FR	Frequency Range
GPU	Graphic Processing Unit
KPI	Key Performance Indicator
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLOps	Machine Learning Operations
NF	Network Function
NEF	Network Exposure Function
NTN	Non-Terrestrial Network
Non-RT	Non-Real-Time

NR	Near-Real-Time
O-DU	O-RAN Distributed Unit
O-RU	O-RAN Radio Unit
PHY	Physical layer
PTP	Precision Time Protocol
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RIC	Radio Intelligent Controller
RT	Real Time
RU	Radio Unit
UE	User Equipment
URLLC	Ultra-Reliable Low Latency Communications

1 Introduction

6GARROW aims to demonstrate how the AI/ML technologies developed across the project can be integrated into practical network systems and validated under realistic operating conditions. Within this vision, WP5 focuses on bridging algorithmic research and practical implementation by developing laboratory testbeds that enable the integration and experimental assessment of the proposed AI/ML solutions. Within WP5, this covers semantic communication, distributed inference, AI-enabled RAN and PHY functions, and cross-domain AI coordination. These testbeds provide a representative environment for evaluating the feasibility of some of the developed concepts and their compatibility with the overall 6GARROW system architecture.

In this context, Deliverable D5.1 presents the initial design of the planned laboratory demonstrations. Four demonstration platforms are developed to validate AI-native technologies across different network domains: (i) a device-edge co-inference architecture combining semantic-aware encoding and wireless communication for gesture recognition in robotic control; (ii) an inference-coordination mechanism for cross-domain network slicing in end-to-end AI-native 6G networks; (iii) a lab demonstration of AI/ML techniques applied to the physical layer; and (iv) an edge intelligence platform demonstrating CSI compression algorithms on an enhanced MIMO testbed.

The work presented in this deliverable is closely related to other activities within the project. It builds upon the high-level architectural framework defined in Deliverable D2.2 [6GARROW-D22], which establishes the overall AI-native system architecture of 6GARROW, and instantiates selected architectural concepts into concrete laboratory testbeds. These architectural descriptions provide the basis for the implementation, integration, and validation activities that will be reported in later WP5 deliverables, including the joint EU–ROK proof-of-concept demonstration and the experimental validation results.

This deliverable is organized as follows:

- Section 2 recalls the 6GARROW functional architecture and describes its relationship to D5.1.
- Section 3 presents the overall architecture of the four planned demonstrations, including their logical architecture, physical/software architecture, interfaces, capabilities, and validation targets.
- Section 4 discusses how these demonstrations relate to the overall 6GARROW functional architecture and introduces a unifying demonstration environment for collaborative edge inference, based on demonstration 1.
- Section 5 concludes the deliverable and outlines the next steps toward implementation, integration and experimental validation within WP5.

2 6GARROW Functional Architecture

This section recalls the 6GARROW functional architecture. It is the common reference for every testbed in this deliverable. D5.1 describes the testbeds to be implemented in WP5, along with their architecture, interfaces, capabilities and targets. Each of these needs a shared view of where an AI/ML function sits in the system. The functional architecture provides this view by organizing AI-native functionalities across the UE, radio access network (RAN), and core network (CN) domains and defining the interfaces and information flows between them. Recalling the functional architecture here makes this dependency explicit and provides the basis for mapping each demonstration onto the overall 6GARROW system architecture. A detailed example is presented later in this deliverable.

The initial use case driven, functional 6GARROW architecture was presented in D2.2 [6GARROW-D22]. The architecture emphasizes the AI-native capabilities supporting joint RAN–CN optimizations with the aim of efficiency and sustainability. The functional architecture was derived based on the use cases described in D2.1 [6GARROW-D21] (see Table 1) and it groups different AI-native functionalities in different subsystems and provided initial information flows between the functional groupings.

Table 1: 6GARROW Use Case Families and Use Cases (from D2.1).

Families	1. AI native 6G design empowering new network services for stakeholders	AI native 6G design for substantially more efficient & sustainable network operation:	
		2. Resource Usage (lower layers)	3. Automation & Failure Recovery (network functions)
Use Cases	AI-as-a-Service for stakeholders & creation of tailored AI models according to End Users needs	Highly efficient communications with light infrastructure	Anomaly Detection & Recovery Strategies
	URLLC for Machine Vision-based Robotic Control	Dynamic resource allocation for intercontinental RAN-Core	Cultivating Radio Measurements into Value-Added Services for Spectrum Sharing
	Semantic Communications Services for Agents	Overhead reduction through Semantic Closed Control Loop for Private 5G Management Systems	
		RAN/UE & RAN/CN Cross-Domain Coordination	

A revised version of the functional architecture was presented in D2.3 [6GARROW-D23]. The main changes in comparison to the earlier version in D2.2 are the addition of the AI-native Link and Flow control layer in the UE and in the RAN, the replacement of AI model and agent deployment on the top in the “AI stack” on the same level with MLOps, and the shift of the loss and parameter control into the AI-control layer, as shown in Figure 1 (from D2.3).

Not all elements of the functional architecture are addressed experimentally within WP5. The demonstrations described in Section 3 instantiate those layers and blocks that are reachable in a laboratory environment and relevant to the AI/ML techniques developed in WP3 and WP4; each subsection states which architectural elements the demonstration instantiates and which use case family it addresses. Section 4 aggregates these statements and identifies the resulting coverage of the architecture.

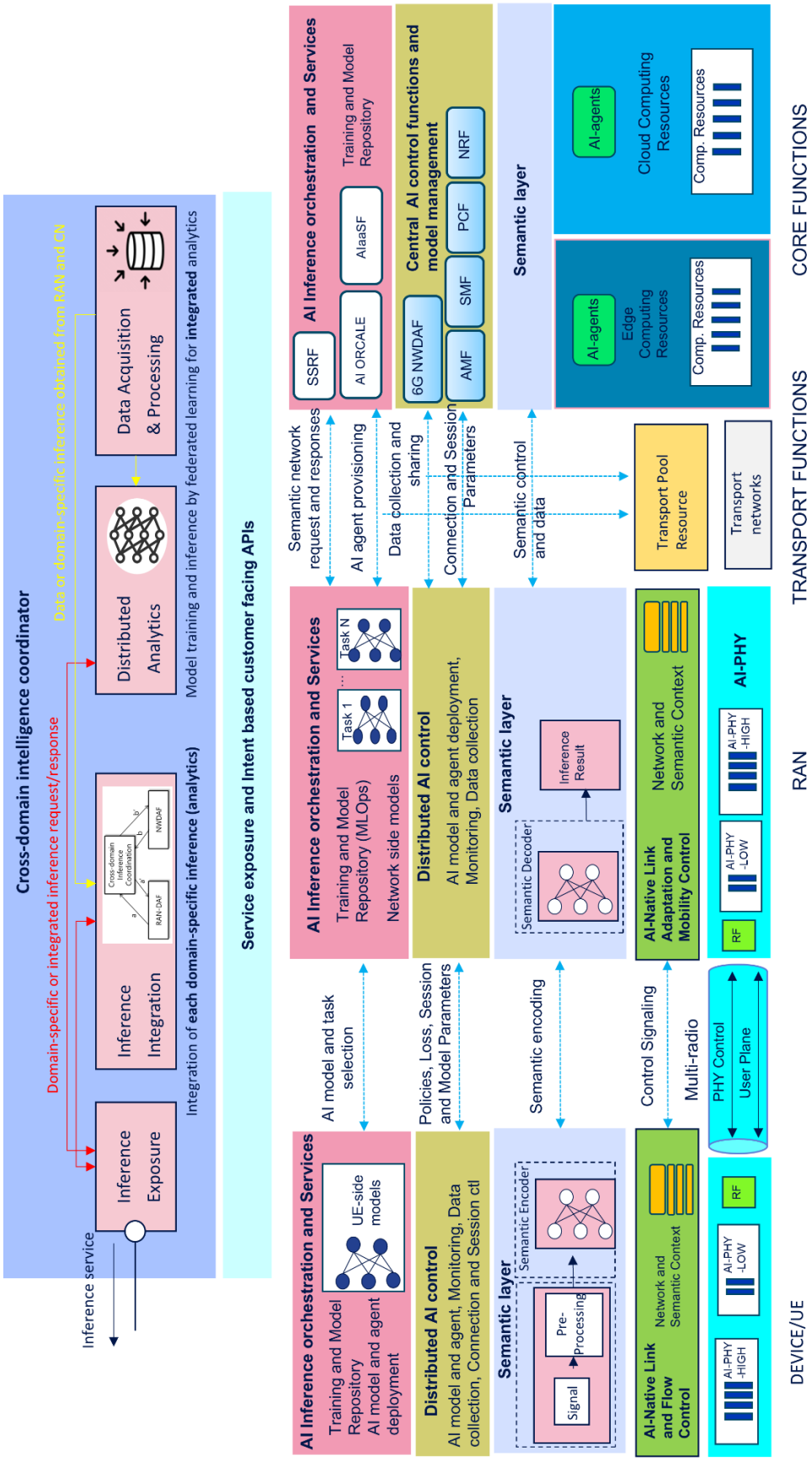

Figure 1: 6GARROW functional architecture (from D2.3).

3 Description of Demonstrations

This section presents the four demonstrations developed within WP5 to validate key AI/ML innovations of the 6GARROW project. Each demonstration is described from an architectural perspective, highlighting its design, functionality, and validation approach. Collectively, the demonstrations cover AI-native functionalities across the device, RAN, and CN domains, illustrating how the technologies developed in other WPs are instantiated within the 6GARROW system architecture.

3.1 Semantic-Aware Device-Edge Co-Inference Demonstration

This demonstration will deliver an implementation of an architecture for device-edge co-inference that integrates semantic-aware encoding and inference with wireless communication. The demonstration will be based on the testbed and experimentation platform for edge intelligence applications, currently under development at FhG. A concrete example is provided by the demo setup illustrated in Figure 2, which implements gesture recognition in a robotic control application. The depicted setup integrates sensing, on-device semantic encoding performing, NR-based transmission/reception over a wireless channel, and semantic decoder for edge inference.

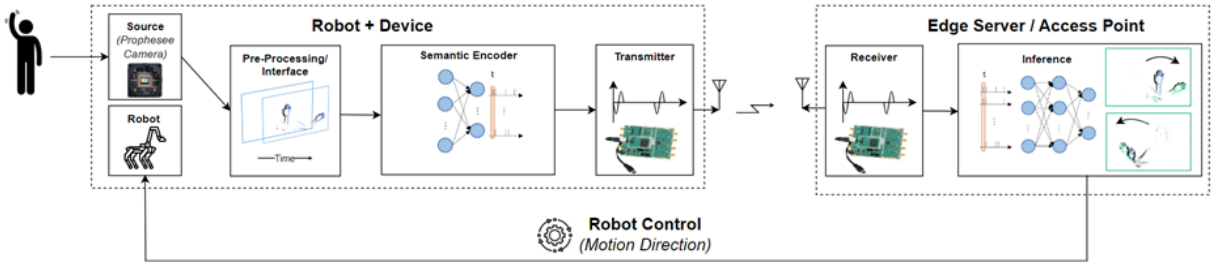

Figure 2: Semantic aware device-edge co-inference for robotic control.

3.1.1 Objectives and Architectural Relevance

The Semantic-Aware Device-Edge Co-Inference Testbed aims to validate the feasibility and performance of AI-native, semantic-aware distributed inference across device and edge infrastructure within the 6GARROW architecture.

The main objectives are:

- Demonstrate device-edge co-inference where AI processing is split between device and edge.
- Validate semantic communication principles, enabling transmission of task-relevant information instead of raw data.
- Evaluate trade-offs between task accuracy, communication overhead, and computational complexity.
- Support low-latency and energy-efficient AI-driven services, aligned with URLLC-type use cases such as robotic control and machine vision.
- Provide evidence of practical feasibility and system integration, in line with WP5 validation goals.

From an architectural perspective, this testbed instantiates key D2.2 elements, including:

- AI inference orchestrations & services
- Distributed AI control and model management
- Semantic layer
- Infrastructure layer

Furthermore, it directly maps to D2.3 use cases such as:

- semantic-aware device-edge co-inference
- AI-driven robotic control
- task-oriented communication services

3.1.2 Logical Architecture

Figure 3 illustrates the logical architecture of the proposed semantic-aware device-edge co-inference demonstration. The architecture consists of three logical domains: the edge server, the network, and the UE (robot equipped with a communication module), which jointly support adaptive semantic communication and distributed AI inference.

The edge server maintains a repository of candidate models and performs task-driven AI model management. Based on the application requirements, device capability, and network conditions, the edge selects the most appropriate encoder-decoder pair and coordinates model deployment to both the edge and the UE. During runtime, the deployed semantic decoder performs task inference on the received semantic representations and generates the corresponding robot control commands.

The network provides wireless connectivity between the UE and the edge server while continuously monitoring the communication environment and transports the semantic feature representations generated by the UE with low communication overhead.

The UE maintains a local repository of semantic encoder models and deploys the encoder selected by the edge. It acquires sensor observations and executes semantic encoding to extract compact task-oriented representations. These semantic features are transmitted to the edge for distributed inference.

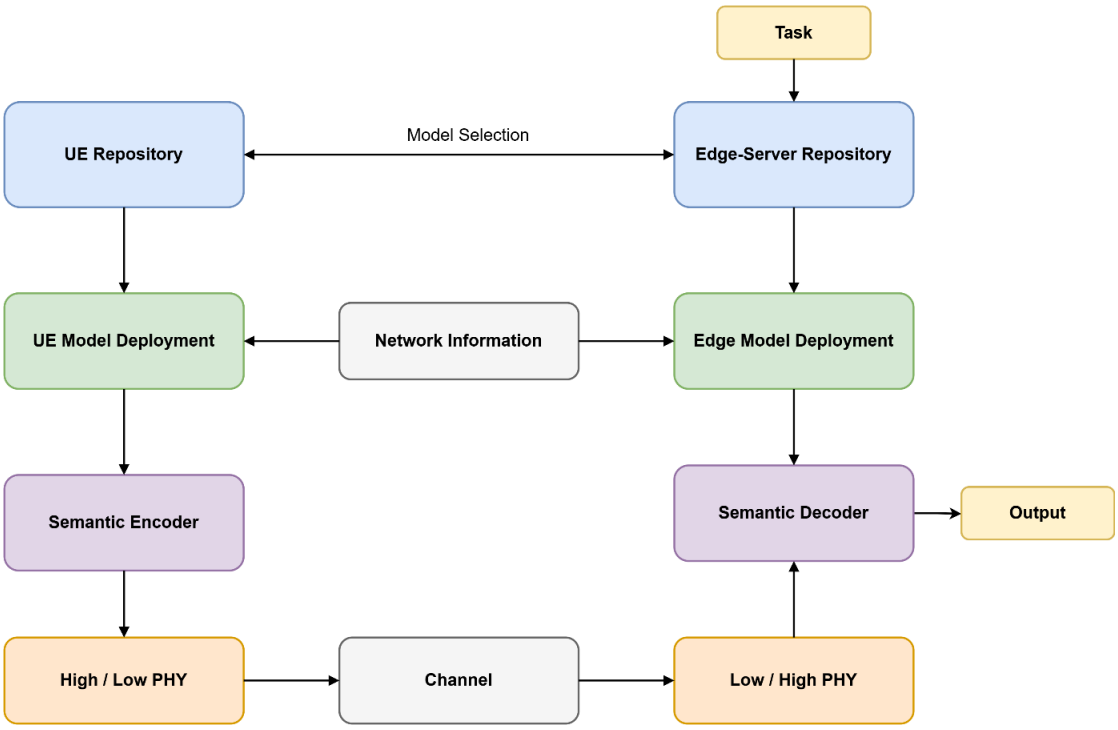

Figure 3: Logical architecture of the proposed demonstration.

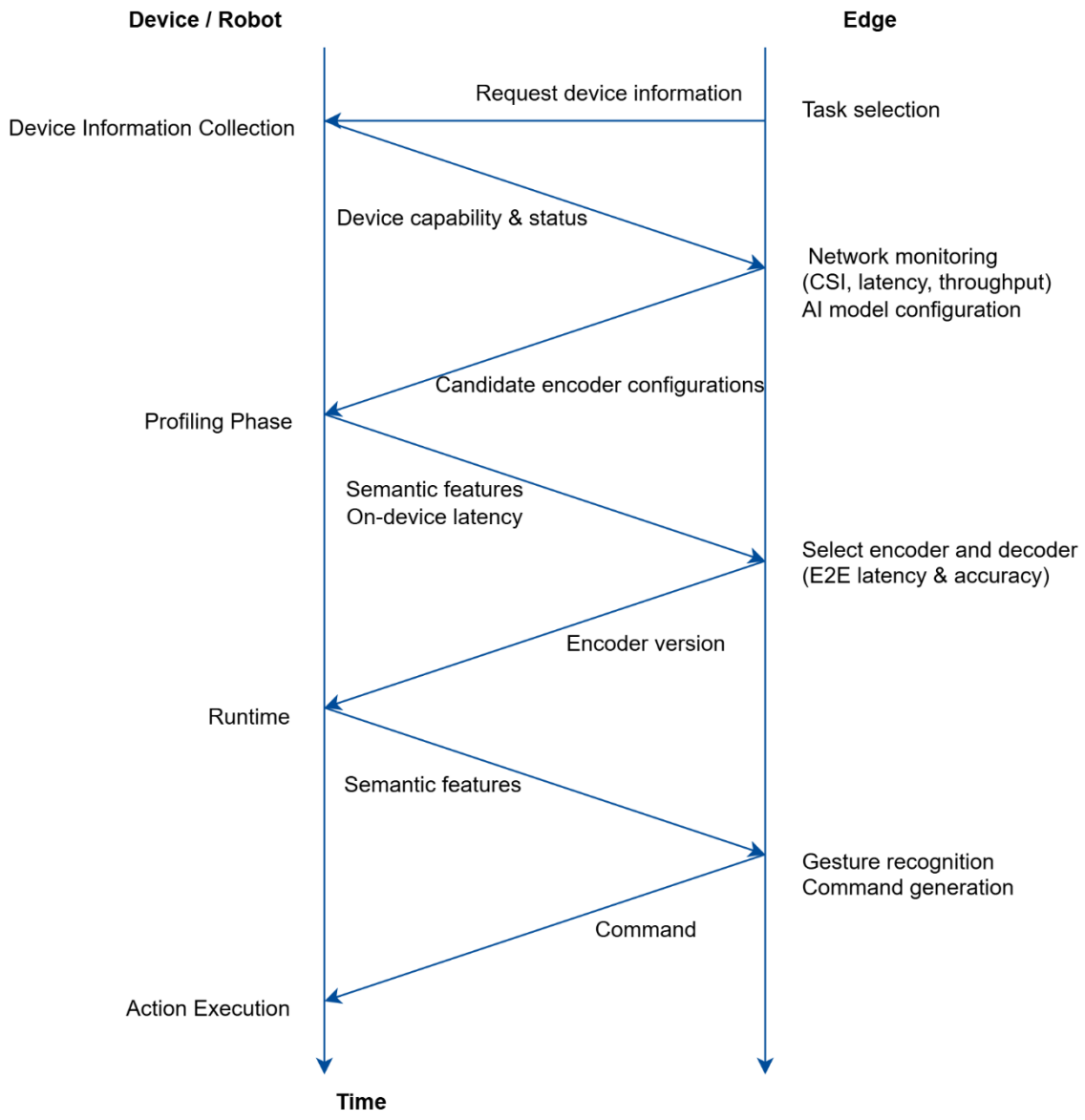

Figure 4: Operational workflow of the proposed demonstration.

Following the logical architecture, Figure 4 illustrates the operational workflow of the proposed framework by describing the interactions between the device and the edge during system initialization, model adaptation, and runtime execution.

The workflow begins at the edge side, where a target robotic task is selected. The edge then requests device information from the robot. In response, the robot performs device information collection and reports its capability and runtime status, such as available sensing resources, computing capability, memory availability, and current operating condition. Based on the device status and network conditions, the edge performs AI model configuration and determines candidate encoder configurations for the robot.

A test-runtime phase is then used to profile different encoder configurations. During this phase, the robot executes the configured encoder and sends the extracted semantic features together with the measured on-device latency to the edge. The edge evaluates the received feature representation jointly with the end-to-end latency and recognition accuracy, and then selects the most appropriate encoder-decoder pair for the current device and wireless communication conditions. The selected encoder version is then sent to the robot before the operational runtime begins.

During runtime, the robot processes sensory observations using the selected encoder and transmits compact semantic features to the edge instead of raw data. The edge performs gesture recognition based on the received semantic features and generates the corresponding control command. The command is sent back to the robot, which executes the required action.

3.1.3 Physical/Software Architecture

The demonstrator comprises three nodes: a device-side node, a network-side node and an edge server. Each node is a host computer with its attached accelerators and radio front end. The demonstration realised on this platform is the one defined in Section 3.1.2.

Device Side Node:

- Semantic encoding accelerator. Two options are available. An embedded GPU platform of the NVIDIA Jetson class [NV26] runs frame-based encoder architectures and provides on-board power measurement. The SpiNNaker2 neuromorphic platform [GMH+24] provides an event-driven processing path, in which computation follows scene activity rather than a fixed frame clock.
- Vision sensors. A conventional RGB camera and a Prophesee event-based vision sensor [PRO26]. The event sensor produces sparse asynchronous events rather than full frames, which changes the data offered to the encoder.
- Device radio front end. A USRP B210 software-defined radio, connected to the device host over USB 3.0, covering 70 MHz to 6 GHz with two transmit and two receive channels.
- Robot and control interface. The robotic platform and the interface through which it receives and executes the control commands returned by the edge.
- Device host computer. An embedded PC or the robot's onboard computer. It runs the OpenAirInterface (OAI) NR user-equipment stack, which performs the upper-layer, MAC and physical layer processing, and drives the attached radio front end. It also collects the device capability and status information reported to the edge. Where no separate accelerator is used, the semantic encoder runs on this host as well.

Edge Side Node:

- Edge server. A GPU-equipped server hosting the semantic decoder, the task inference function, the model repository and the model selection and deployment logic. It is realised on the edge intelligence experimentation platform under development at FhG.

Network Side Node:

- Network-side host computer. A general-purpose server running the OAI gNB stack. It performs the network-side protocol processing and the mapping of transmitted content onto radio resources, and reports the channel-state and link-adaptation quantities exposed by the stack.
- Network-side radio front end. A USRP N310 or X-series software-defined radio, connected over 10 Gigabit Ethernet, supporting an external frequency and timing reference.

Software Components:

- Software. The OAI 5G stack and the USRP hardware driver run on a real-time-patched Linux kernel with core isolation and CPU pinning, which NR timing on general-purpose hardware requires. On the device host, this partitioning also separates the cores serving the protocol stack from those available to semantic encoding.
- Semantic encoders and decoders are implemented in standard AI frameworks, with an export path to the embedded and neuromorphic targets. Logging is common to all nodes and timestamped.

- Channel. Measurements are made in two arrangements: a conducted arrangement, in which the radio front ends are connected through calibrated attenuators, and over the air.

Two link configurations are supported. In the first, the device-side node and the network-side node establish a 5G NR link over the Uu interface (Radio interface between a user equipment and the 5G network, typically a gNB), and semantic representations are carried as payload on the uplink shared channel. In the second, two device-side nodes communicate directly over the PC5 sidelink interface (Direct sidelink interface/reference point between UEs, used for services such as NR-V2X) in Mode 2, without a network-side node, one of them acting as synchronisation reference. It is based on an existing FR1 sidelink evaluation platform using the OAI NR UE with USRP B210 front ends, which implements the Rel-16 Mode 2 sidelink physical channels and has been operated over the air; the available feature set is narrower than that of the Uu interface.

Parameter	**Uu Configuration**	**PC5 Sidelink Configuration**
Mode	5G NR SA, FR1 TDD	NR sidelink, Rel-16 Mode 2
Centre frequency *(target)*	FR1 mid-band	n38 region, approx. 2.6 GHz
Channel bandwidth *(target)*	20–40 MHz	20 MHz
Subcarrier spacing *(target)*	30 kHz	30 kHz
Nodes	Device-side node, network-side node	Two device-side nodes
Front ends	B210 at device, N310 / X-series at network side	B210 at each node
Transport of semantic representations	Uplink shared channel	Sidelink shared channel

Mapping of functions (Section 3.1.2) to the architecture:

- Vision sensors: sensing and data acquisition.
- Semantic encoding accelerator, or the device host: semantic representation.
- Device host computer: transport of semantic representations and UE-side channel measurement and reporting; UE-side AI-native link and flow control; capability and context monitoring.
- Device radio front end: radio transmission and reception at the device.
- Network-side host computer: transport of semantic representations, resource mapping, channel measurement and reporting; RAN-side AI-native link and flow control.
- Network-side radio front end: radio transmission and reception at the network side.
- Edge server: semantic interpretation and task inference; model selection, deployment and versioning; inference session orchestration; model repository.
- Robot and control interface: execution of the returned control command.

In the sidelink configuration, the functions listed above for the network-side host computer are performed by the participating device-side nodes.

3.1.4 Interfaces and Data Flows

The testbed defines a set of interface classes describing the interaction and data exchange between functional components, rather than the components themselves. Each interface specifies the type of information exchanged, its structure, and its role in enabling distributed AI inference across the device–edge continuum.

- Sensing/Data Acquisition Interface: transfers sensor observations (e.g., camera or event-based vision data) from the sensing subsystem to the semantic encoder for preprocessing and feature extraction.
- Semantic Representation Interface: transports compact semantic feature representations from the UE to the edge server for distributed AI inference, replacing conventional raw-data transmission.
- Device–Edge AI Interface: exchanges intermediate inference information and AI control parameters between the UE and the edge, supporting distributed inference execution and runtime coordination.
- Communication Interface: provides wireless transport of semantic features, AI control messages, and monitoring information over the RAN while supporting low-latency and reliable communication.
- AI Model Management Interface: supports AI model selection, deployment, configuration updates, and lifecycle management between the edge server and the UE.

3.1.5 Capabilities

The testbed provides the following capabilities:

- Device-edge distributed AI inference
- Semantic communication for task-oriented data transmission
- Adaptive AI processing based on channel and system context
- Energy-efficient computation through split inference
- Support for low-latency applications (e.g., robotic control)
- Integration with real wireless communication environments
- AI model management and orchestration (AIaaS paradigm)

3.1.6 Validation Targets

The testbed is designed to support validation of the following KPI/KVI categories:

1. Performance KPIs

- End-to-end latency (device → edge inference)
- Task accuracy (e.g., classification / recognition accuracy)
- Robustness of inference under varying communication conditions

2. Communication KPIs

- Data reduction / compression ratio

3. Validation Goals (aligned with WP5 / Objective 4)

- Demonstrate practical feasibility of AI-native concepts
- Identify integration and interoperability issues
- Optimize AI/ML models for real deployment
- Provide experimental evidence supporting 6G architecture integration

3.2 Cross-Domain Inference Coordination Testbed

This testbed addresses the practical realisation and validation of the RAN-CN Coordination concept introduced in the 6GARROW system architecture. Its principal use case is UE mobility prediction across the integrated RAN-Core domain, which is treated here as a representative scenario through which the broader value of RAN-CN Coordination can be demonstrated. The underlying premise is that mobility-related inference performed jointly over RAN-side and Core-side observations can yield higher accuracy and operational efficiency than inference performed in either domain alone. In a later stage, the resulting prediction capability is intended to drive proactive and efficient network slicing resource allocation; in the current stage, the testbed concentrates on building the cross-domain data foundation on which future inference functions will rely. A high-level overview of the testbed is shown in Figure 5.

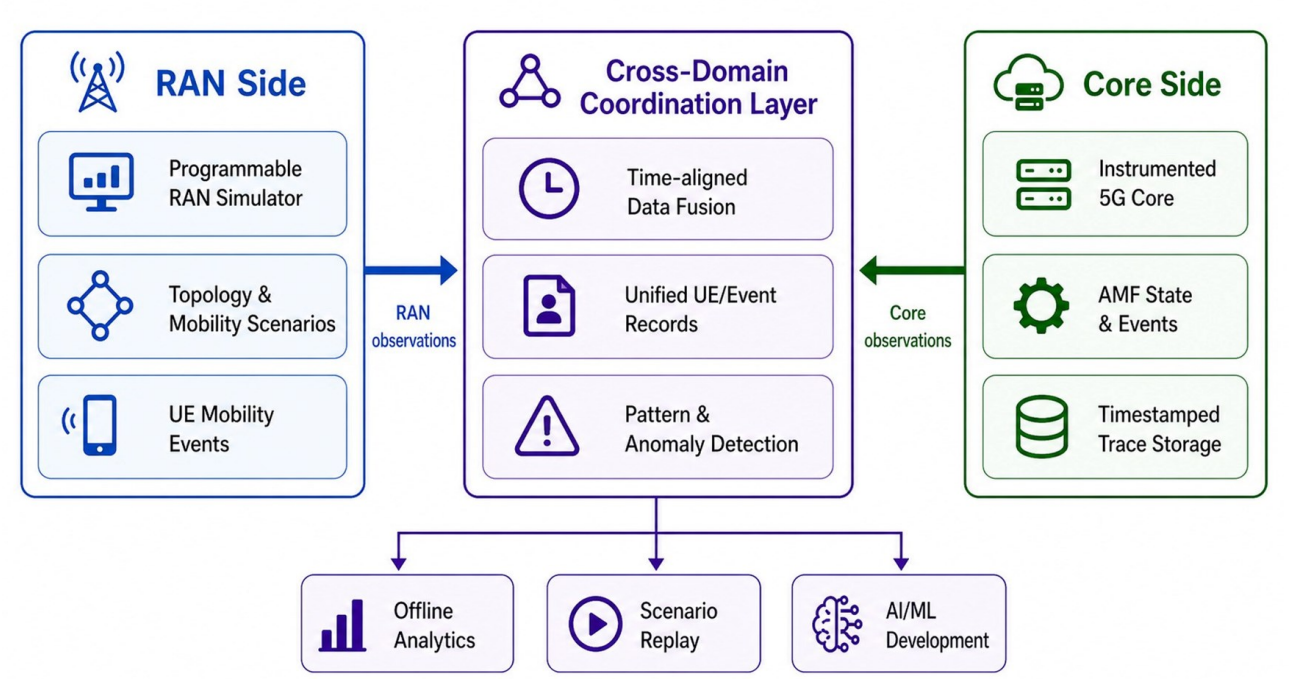

Figure 5: Cross-Domain Inference Coordination Testbed — high-level overview

The setup integrates a multi-cell RAN simulator capable of producing programmable mobility scenarios involving multiple concurrent UEs, together with an instrumented 5G Core that exposes mobility-relevant state. A coordination layer captures, time-aligns, and consolidates events observed at both domains into a unified per-event record suitable for downstream cross-domain analytics.

3.2.1 Objectives and Architectural Relevance

The Cross-Domain Inference Coordination Testbed for Network Slicing aims to validate the feasibility of jointly leveraging RAN-side and Core-side observations for AI-native, mobility-driven decision making, in line with the 6GARROW vision of integrated RAN-Core operation.

The main objectives are:

- Provide a controlled experimentation environment in which UE mobility events can be deterministically generated across a multi-cell RAN.

- Capture, in a temporally consistent manner, mobility-relevant signals from both the RAN domain (handover, mobility registration update, registration, de-registration) and the Core domain (UE context state, location associations, session-level effects).
- Build a cross-domain data foundation on which future AI/ML-based inference functions can be developed, trained, and benchmarked.
- Demonstrate, through the representative use case of UE mobility prediction, the architectural value of RAN-CN Coordination as defined in D2.2.
- Serve as a development substrate towards future proactive network slicing resource allocation guided by cross-domain mobility inference.

From an architectural perspective, this testbed instantiates the following D2.2 elements:

- RAN-CN Coordination as a first-class architectural concept.
- AI-enabled RAN and Core layers, by exposing the measurement and event surfaces that future AI/ML functions will consume.
- Distributed AI control and model management, anticipating the placement of inference functions across RAN and Core domains.

It maps to the use cases concerning:

- UE mobility prediction as an enabler of proactive resource management.
- Network slicing resource allocation guided by RAN-Core joint analytics.

Service continuity under inter-cell and inter-Tracking-Area mobility.

3.2.2 Logical Architecture

The logical architecture of the Cross-Domain Inference Coordination Testbed follows a layered view consistent with the 6GARROW functional architecture, combined with a deployment mapping that reflects the distributed nature of RAN-Core coordination. This dual perspective ensures consistency with the architectural principles introduced in WP2 while reflecting the practical realisation of the testbed in WP5.

A. Functional Architecture (Horizontal View)

The testbed exposes the following functional layers:

1. Infrastructure Layer

This layer provides the simulated multi-cell radio access and the associated transport between RAN and Core. The cellular topology, including Tracking-Area arrangement and inter-cell adjacency, is configurable to support diverse mobility patterns.

2. Mobility Event Surface (RAN side)

This layer generates and exposes UE mobility-related events arising from the radio access, including inter-cell handover, mobility registration update upon Tracking-Area change, initial registration, and de-registration. These events constitute the primary input on which cross-domain coordination is built.

3. Network Function State Surface (Core side)

This layer exposes the relevant Core-side state changes triggered by RAN-originated mobility, captured at the Access and Mobility Management Function and made available through a structured query and event interface. It enables the Core domain to participate as a first-class data source in cross-domain inference.

4. Cross-Domain Coordination Layer

This layer aligns the two surfaces in time, correlates RAN events with their Core-side consequences, and consolidates the result into a unified per-UE, per-event record suitable for downstream

consumption. The Cross-Domain Coordination Layer represents the architectural locus in which RAN-CN Coordination materialises as a measurable and reusable artefact.

5. Inference and Services Layer (forward-looking)

This layer is reserved for future AI/ML inference functions that will consume the unified cross-domain record. At the present stage, the layer is architecturally identified but not yet instantiated; its inclusion ensures that the testbed remains aligned with the AI-native principles of the 6GARROW system and supports a coherent evolution path towards joint RAN-Core inference.

B. Deployment Mapping (Vertical View)

The functional layers are instantiated across multiple physical domains:

Domain	Role within the Architecture
User Equipment (simulated, multiple)	Generates mobility behaviour and triggers RAN-side procedures across the configured cellular topology
RAN (multi-cell, simulated)	Realises mobility procedures and exposes RAN-side mobility events as a continuous, time-stamped stream
Core Network	Maintains UE context and exposes the Core-side mobility state induced by RAN-originated events
Cross-Domain Coordination	Time-aligns, correlates, and consolidates RAN and Core observations into a unified cross-domain record
Analytics / Inference (forward-looking)	Future AI/ML inference functions operating over the unified cross-domain record

This mapping reflects the distributed nature of cross-domain coordination, where mobility-relevant information is produced by both RAN and Core domains and joined at a dedicated coordination point.

C. Architectural Alignment

The logical architecture is consistent with the 6GARROW functional architecture and supports traceability to higher-level design elements. In particular, the testbed contributes to:

- Practical realisation of RAN-CN Coordination as a measurable architectural property.
- Provision of a data substrate for joint RAN-Core inference, in support of the AI-native premise of the 6GARROW system.
- Validation of mobility-driven cross-domain analytics as a precursor to proactive slicing-aware resource management.

3.2.3 Physical/Software Architecture

The testbed is implemented as a software-based platform that combines a programmable RAN simulator, an instrumented 5G Core, and a coordination component that operates between them. The software-defined realisation supports reproducibility, deterministic replay of mobility scenarios, and integration with later AI/ML development stages.

RAN side:

- A multi-gNB simulator supporting concurrent UE registrations, inter-cell handover, and Tracking-Area-based mobility behaviour.
- A configurable cellular topology, including per-cell Tracking-Area assignment and inter-cell neighbour relationships.
- Programmable mobility scenarios in which sequences of UE events (registration, handover, mobility update, de-registration) can be deterministically replayed in time.
- An interactive control surface for ad-hoc experimentation and scenario authoring.

Core side:

- A 5G Core deployment exposing the Access and Mobility Management Function and its associated mobility-management procedures.
- Instrumentation of the AMF to expose UE context state through both periodic queries and continuous event observation.
- Persistence of the resulting state and event traces with consistent timestamps suitable for cross-domain correlation.

Cross-Domain Coordination Layer:

- Continuous, time-aligned consolidation of RAN-side and Core-side observations into a unified record per UE and per event.
- Detection of multi-step mobility patterns (for example, inter-Tracking-Area movement and round-trip mobility) and of cross-domain anomalies such as procedure-level failures.
- A serialised record format suitable for offline analytics, scenario replay, and downstream inference development.

The implementation is currently realised in a fully software-defined manner.

3.2.4 Interfaces and Exposed Analytics

The testbed exposes a small number of well-defined interface classes that together implement the cross-domain coordination function. These interfaces are intentionally kept at the level of architectural abstractions rather than finalising protocol-level specifications at this stage.

- RAN event interface: a continuous stream of mobility-related events originating from the multi-cell RAN, including handover, mobility registration update, initial registration, and de-registration, each annotated with the participating UE identity, source and target cell information, and timing.
- Core state interface: a query and event surface over the AMF, exposing UE context including registration status, current serving cell, Tracking-Area association, and connection-management state.
- Cross-domain coordination interface: the consolidation point at which the RAN-side and Core-side surfaces are time-aligned and merged into a unified per-event record annotated with both RAN and Core attributes.
- Inference consumption interface (forward-looking): a downstream-facing interface through which the unified record will be exposed to future AI/ML inference functions, including those for mobility prediction.
- Validation and measurement interface: persistent logging of observed events and state transitions for the purpose of reproducibility, scenario replay, and downstream KPI computation.

Data flows in the testbed follow a clear pattern: scripted UE behaviour generates RAN events; those events propagate to the Core and induce corresponding state changes; the coordination layer captures both sides and produces the unified cross-domain record, which becomes the artefact on which all subsequent analytics depend.

3.2.5 Capabilities

The current capabilities of the testbed include:

- Generation of programmable, deterministic mobility scenarios involving multiple concurrent UEs across a multi-cell topology with configurable Tracking-Area structure.
- Real-time capture of RAN-side mobility events at the granularity required by cross-domain analytics, including handover, mobility registration update, and registration lifecycle events.
- Real-time capture of Core-side UE context state and mobility-related transitions, exposed by the AMF and consistent in time with the RAN-side observations.

- Time-aligned consolidation of the two domains into a unified per-UE, per-event record that is persisted for downstream consumption.
- Detection of cross-domain anomalies, including procedure-level failure conditions that become visible only when RAN and Core observations are jointly considered.
- Replayability of recorded mobility traces, enabling controlled experimentation with future inference and resource-management functions over identical input sequences.

Together, these capabilities establish the data foundation required to develop, train, and evaluate RAN-CN joint inference functions for the use cases targeted in WP5, with UE mobility prediction acting as the representative entry point and proactive slicing-aware resource allocation as the longer-term objective.

3.2.6 Validation Targets

The testbed defines validation targets at the architectural and data-collection level. The targets supported by the present testbed include:

- Cross-domain event capture coverage: the proportion of RAN-originated mobility events for which a temporally consistent Core-side counterpart is captured in the unified record.
- Cross-domain correlation latency: the elapsed time between the occurrence of a RAN event and the availability of the corresponding unified, cross-domain record.
- Mobility traceability across multi-cell paths: the ability to reconstruct, from the unified record, the full mobility history of a UE traversing multiple cells and Tracking-Areas without loss of continuity.
- Scenario reproducibility: the degree to which an identical mobility scenario can be replayed to produce comparable cross-domain records, supporting controlled evaluation of future inference functions.
- Data completeness for downstream inference: the extent to which the unified record contains the attributes required by candidate cross-domain inference functions, particularly those targeting mobility prediction.

These targets establish the conditions under which the testbed can serve as a substrate for the RAN-CN coordinated inference functions to be developed and evaluated in subsequent activities, including the proactive slicing-aware resource allocation that constitutes the longer-term objective of this contribution.

3.3 Open RAN PHY AI/ML Techniques Testbed

This demonstration replaces selected legacy RAN PHY-layer function with an AI/ML model to validate its integration, interoperability and performance. The demonstration will be based on the RAN testbed currently under development at ETRI. As shown in Figure 6, the demonstration setup consists of an O-DU implementing PHY-layer functions, a PTP Grandmaster for network synchronization, an O-RU responsible for digital-to-RF signal conversion, and a user equipment (UE).

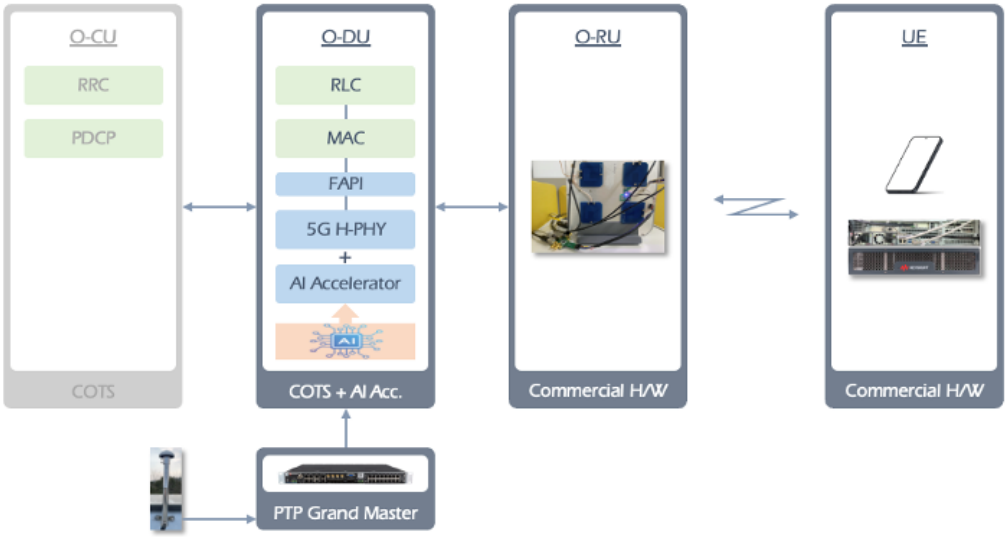

Figure 6: AI/ML techniques for RAN Physical layer.

3.3.1 Objectives and Architectural Relevance

The Physical-Layer AI/ML Technique Testbed demonstrates the feasibility of deploying AI/ML inference model alongside conventional PHY-layer functions in a wireless infrastructure.

The main objectives are:

- Demonstrate the operation of an AI/ML-based PHY-layer technique within a testbed
- Validate the feasibility and interoperability of an AI/ML-based PHY-layer function integrated with conventional PHY-layer processing chains
- Evaluate the performance of the AI/ML-based PHY-layer function under realistic condition
- Provide evidence of practical feasibility and system integration, in line with WP5 validation goals

From an architectural perspective, this testbed instantiates key D2.2 elements, including:

- Physical layer AI/ML techniques

Furthermore, it directly maps to D2.3 use cases such as:

- RAN/UE & RAN/CN Cross-Domain Coordination

3.3.2 Logical Architecture

The logical architecture of the Physical-Layer AI/ML Technique Testbed is derived from the AI-PHY infrastructure-layer defined in WP2 and implemented within the WP5 validation environment.

The architecture is represented through two complementary views: a Functional Architecture describing the logical functions and their interactions, and a Deployment Mapping describing how these functions are instantiated on the testbed.

A. Functional Architecture (Horizontal View)

The testbed instantiates two functional layers that reflect the AI-native architecture of 6GARROW

1 AI/ML PHY Layer

This layer hosts an AI/ML inference model that replaces a selected conventional PHY-layer function, enabling the evaluation of AI-native PHY processing within the testbed.

It provides:

- an AI/ML inference model replacing a selected PHY-layer processing function
- interoperability with the remaining conventional PHY processing chain

2 Infrastructure Layer

This layer provides the underlying wireless infrastructure required to run the AI/ML PHY function.

It provides:

- wireless connectivity for AI/ML PHY-layer validation
- Open-RAN O-DU and O-RU functions
- network synchronization using a PTP Grandmaster

B. Deployment Mapping (Vertical View)

The functional layers are instantiated across multiple physical domains:

Domain	Role within the Architecture
O-DU	• Conventional PHY, MAC, and RLC functions • AI/ML-based PHY-layer function runs on AI accelerator
O-RU	• Radio transmission and reception • Interfaces with the O-DU through the O-RAN fronthaul interface
Synchronization	• Provides network-wide timing synchronization using PTP Grandmaster

3.3.3 Physical/Software Architecture

The testbed is implemented on an Open-RAN that integrates conventional PHY-layer processing with AI/ML inference acceleration

O-DU Side:

- Hosts 5G NR baseband processing, including conventional PHY, MAC, and RLC functions
- Integrates an AI accelerator for executing an AI/ML inference model
- Hosts an AI/ML-based implementation of a selected PHY-layer function

O-RU Side

- Provides O-RAN fronthaul connectivity with the O-DU
- Performs Radio transmission and reception with UE

Synchronization Side

- Distributes timing references to the O-DU and O-RU
- Supports synchronized real-time PHY-layer processing and O-RAN fronthaul operation

3.3.4 Interfaces and Data Flows

The testbed defines a set of interfaces describing the interaction between the AI/ML-based PHY-layer function, conventional PHY processing functions, and the underlying Open-RAN infrastructure.

1. AI/ML PHY Function Interface

This interface defines the exchange of data between the AI/ML inference model and the conventional PHY processing chain.

It specifies:

- Input: raw signals, measurements, or outputs from conventional PHY processing functions
- Output: AI/ML model inference results (e.g. signals, data or measurements)
- Timing modes: continuous processing or event-driven execution

This interface enables seamless integration of the AI/ML inference model with the conventional PHY processing chain.

2. O-DU/O-RU Fronthaul Interface

This interface defines the exchange of radio and control information between the O-DU and O-RU.

It specifies:

- O-RAN C/U/S-Plane compliant fronthaul connectivity
- Real-time radio transmission and reception support.

3.3.5 Capabilities

The testbed provides the following capabilities:

- Deployment of an AI/ML-based implementation of a selected PHY-layer function
- Integration of an AI/ML inference model with conventional PHY-layer processing chains
- Interoperability validation between AI/ML-based and conventional PHY-layer functions
- Evaluation of a PHY-layer AI/ML technique under realistic wireless operating conditions

Additionally, it enables:

- Experimentation with interface design and information exchange between AI/ML-based and conventional PHY-layer functions

3.3.6 Validation Targets

The testbed is designed to support validation of the following KPI/KVI categories:

1. AI-Specific KPIs

- PHY-layer AI/ML task accuracy (e.g., estimation accuracy, recognition accuracy)
- PHY-layer AI/ML Inference latency

2. Communication KPIs

- Wireless link performance (e.g., BLER and throughput)
- Robustness of AI/ML inference under channel conditions

3. Efficiency KPIs

- Energy consumption

5. Validation Goals (aligned with WP5 / Objective 4)

- Demonstrate practical feasibility of the PHY-layer AI/ML techniques
- Integration and interoperability with the conventional PHY processing chain
- Provide experimental evidence supporting AI-native RAN envisioned by 6GARROW

3.4 AI/ML-Based CSI/CQI Compression Testbed

The CSI Compression Testbed aims to demonstrate and validate the efficiency of AI/ML techniques for decreasing the traffic on the uplink of future AI-Native systems.

The main objectives are:

- Validate the efficiency of CSI compression in controlled environment
- Demonstrate that inference can be run with limited power consumption and low latency on transmitter and receiver side.
- Validate the efficiency of CSI compression with over the air transmission.

From an architectural perspective, this testbed instantiates key D2.2 elements, including:

- AI-PHY
- PHY control Plane (RAN – device link)

Furthermore, it maps to D2.3 use case family “AI native 6G design for substantially more efficient & sustainable network operation.”

3.4.1 Logical Architecture

The functional architecture of the demonstrator is shown in Figure 7. It defines the actions the system performs in support of the objective described above, and how those actions relate to each other, irrespective of how these actions are implemented.

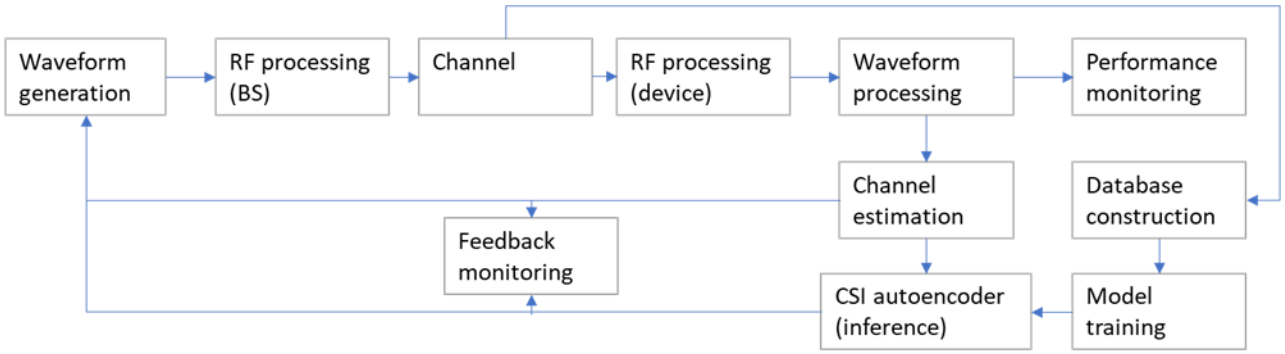

Figure 7: Functional architecture of the CSI compression demonstrator.

The actions are detailed below:

- Waveform generation: encoding of the information bits to be transmitted, framing, modulation.
- Waveform processing: synchronization, equalization, demodulation, deframing and decoding.
- RF processing: digital to analog (or analog to digital) conversion, filtering, amplification, frequency upconversion (or downconversion).
- Channel: over the air transmission of RF signal or emulation of the channel.
- Channel estimation: channel estimation could have been included in waveform processing, but we have isolated it here as it is central to the demonstrator's goals.
- Database construction: recording of channel realizations (when the channel is emulated) in order to build the database for CSI compression model training.
- Model training: construct the model for CSI compression.
- CSI autoencoder: compression and decompression of CSI. The output of the compression feeds the 'feedback monitoring' block.
- Feedback monitoring: compares the feedback of CSI with and without compression.
- Performance monitoring: packet error rate measurement, qualitative assessment of the received constellation.

An example of temporal sequencing is provided below:

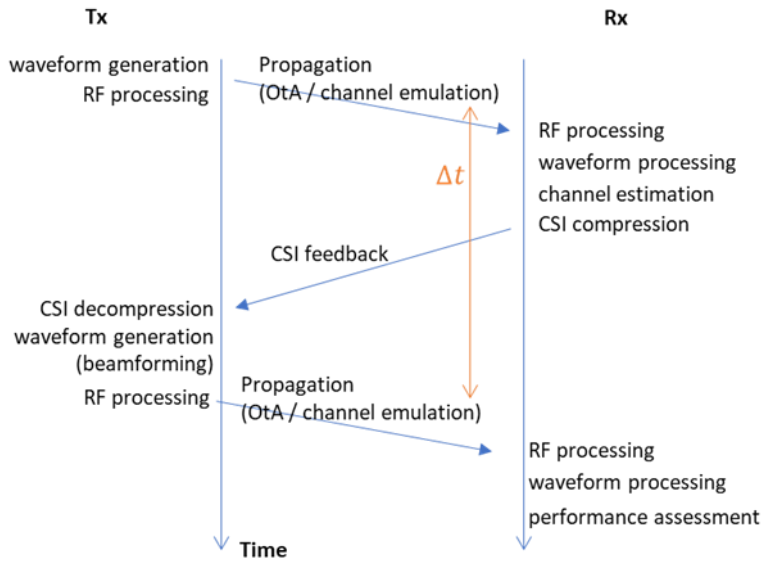

Figure 8: Temporal sequencing for CSI feedback compression assessment

The channel of the first propagation is estimated and sent back to the transmitter. The transmitter then uses this channel estimate for beamforming the transmitted signal toward the receiver.

Two problematics arise from this description: first the quality of the feedback information for the beamforming must be as high as possible (in order to maximize the power transmitted to the receiver) and then the delay between the estimation and the application (, channel aging) must be taken into account (the channel may have changed in between).

3.4.2 Physical / Software architecture

The high-level structure of the transmitter of the demonstrator is described below (Figure 9).

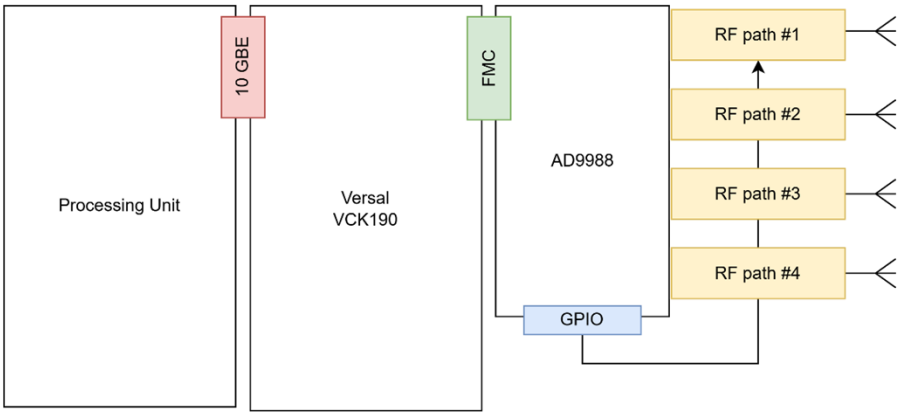

Figure 9: High Level architecture of the transmitter.

Description of architectural components:

- The processing unit is a computing unit dedicated to signal processing and deployment of network functions. It is designed as a high-performance computing server, equipped with a latest generation x86 processor and reinforced by various hardware accelerators to meet the demands of intensive real-time processing for the physical layer.

  It will consist of several modules:
  - The processor, designed for high workloads, handles the general processing of the system and manages the control and management operations. It acts as the main brain, coordinating the various tasks and data exchanges between components.
  - Intensive computing accelerators: For massively parallel operations, the unit will integrate GPU (Graphics Processing Unit) or MMPA (Massively Parallel Processor Array) type accelerators, for example an NVIDIA model. These accelerators are particularly well-suited for handling intensive computing workloads, such as signal processing algorithms or matrix computations.
- Versal VCK190: The architecture is based on a modular organization integrating both advanced digital processing capabilities and a complete radio chain. At the heart of the system lies the AMD Versal component, which brings together several complementary units:
  - AI Engine: VLIW/SIMD vector processors dedicated to intensive processing tasks.
  - Programmable Logic (PL): reconfigurable logic enabling the implementation of specialized processing and adaptation to different operational profiles.
  - Processor: control unit responsible for centralized management and coordination of the sub-modules.
- AD9988: RF DAC/RF ADC/4T4R set, managing four transmit paths.
- RF path: Front-End module handling analog processing across the frequency bands.
- Channel emulator: the channel will be emulated by the F8 PROPSIM engineered by Keysight Technologies.
- Antennas: for over the air testing, monopole antennas will be used.

The architecture of the receiver is symmetrical to the one of the transmitter, with the difference that the former uses only one antenna / RF path.

Mapping of functions (Figure 7) to the architecture:

- Processing Unit: waveform generation (transmitter), waveform processing (receiver), channel estimation, CSI autoencoder (compression at transmitter, decompression at receiver), model training.
- Versal: FFT / IFFT, synchronization, digital filtering
- AD9988: DACs / ADC
- RF path: duplexing, filtering, amplification, transposition.

3.4.3 Interfaces

The physical interfaces of the demonstrator (Figure 10) are:

- Processing unit / Versal: 10-GigaByte Ethernet (GBE)
- CPU / GPU: internal to the processing unit
- Versal / AD9988: FMC
- AD9988 / RF path: GPIO for configuration and RF connector for signal.
- Tx/Rx: either over the air or channel emulator (SMA)

3.4.4 Capabilities

- Typical modulation parameters (bandwidth 5, 10, 20, 40 MHz, modulation from BPSK up to 256-QAM, turbo-coding rate from 1/3 up to 5/6)
- Frequency bands: over the air 6.6 GHz (experimental license from French Regulator), with channel emulator between 2.6 GHz and 6.0 GHz.
- Channel emulation: the Propsim F8 channel emulator supports 5G radio channel models defined in 3GPP TR38.900 and custom. RF interface channel frequency ranges from 220 to 6000 MHz. RF interface channel signal bandwidth goes up to 160 MHz.
- Training and inference: exploitation of capabilities of the GPU
- GUI: a Graphical User Interface has been designed to track and monitor performances in real-time. A snapshot of this interface is provided below. The GUI displays: constellation for payload and control links, estimated channel power, Control Redundancy Check (i.e., similar to packet error rate) – bottom-left figure.

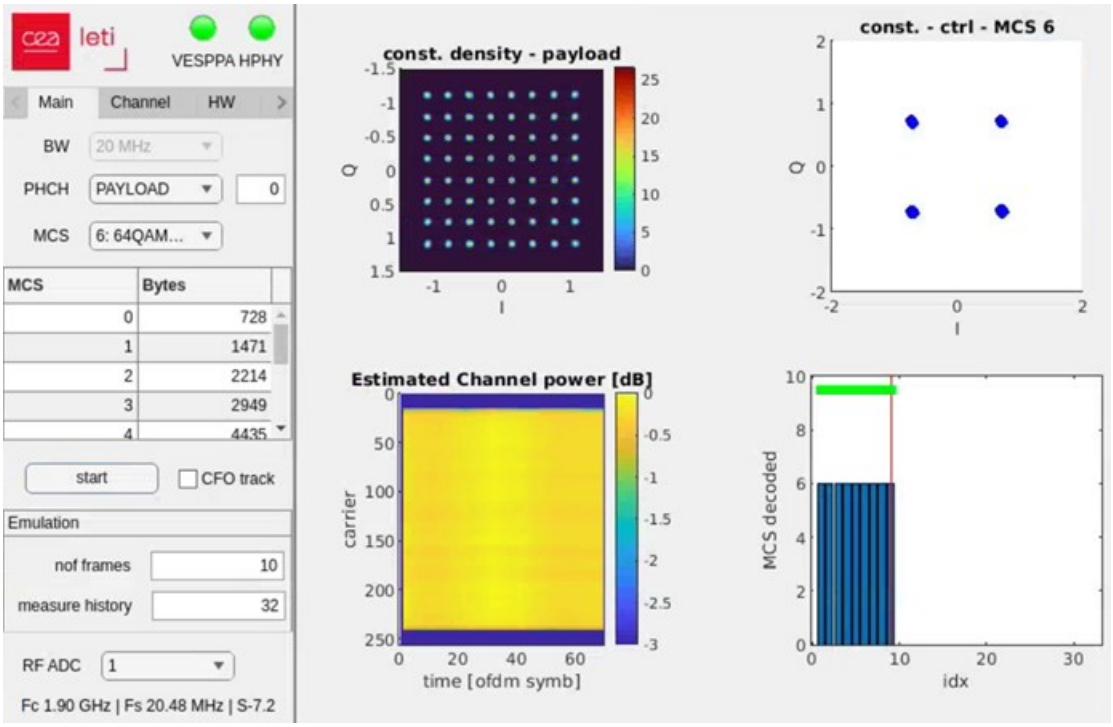

Figure 10: Snapshot of the GUI.

- Feedback monitoring: a monitoring system will be developed to assess the CSI information that goes through the UL.

3.4.5 Validation Targets & KPI

The following validations will be run:

- Validation on software simulation. This work is currently done in the context of other work package and involves computer simulations only. The goal is to confirm the feasibility of CSI compression.
- Training and inference on embedded GPU. Channel database collection, model training, compression and decompression on ‘process unit’ embedded GPUs. This step will confirm the dimension of the platform computational capacities.
- Validation of the transmission with compressed CSI via GUI (channel decoding and constellation). This is the assessment of the quality (compression and aging) of the CSI used for beamforming signal.
- Performance of GPU (number of compression / decompression per second, power consumption, latency).
- Throughput of feedback. This will allow to compare UL rates with compressed and non-compressed CSI.
- Performance comparison between compressed and non-compressed CSI feedback (PER, SNR).

4 An Outlook towards a unifying Demonstration Environment for Collaborative Edge Inference in RAN

In this section we address the mapping between the described demonstrations and the 6GARROW functional architecture. As an exemple, we further provide an outlook towards a unifying demonstration environment for collaborative edge inference in radio access networks. The defined demonstration blocks are explicitly mapped to the layers of the functional architecture.

For this purpose, we start with a high-level representation of the 6GARROW functional architecture, which can be summarized as a layered architecture, as shown in Fig 11 (from D2.3).

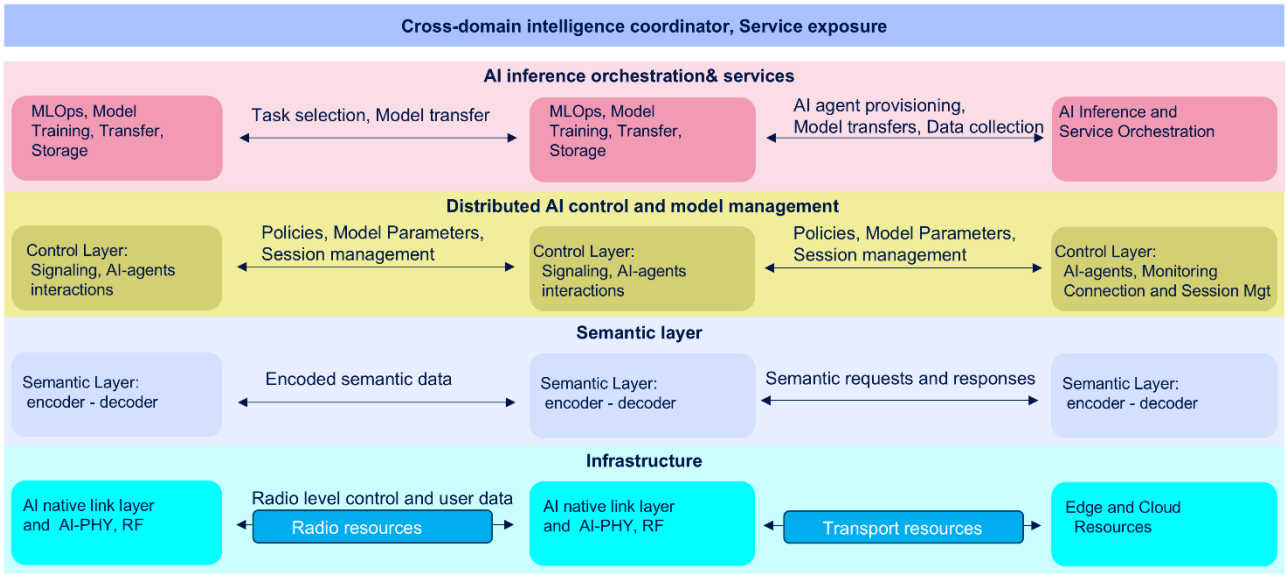

Figure 11: 6GARROW High-level view on the functional architecture.

As described in more detail in D2.3, the infrastructure layer, encompassing radio, transport and cloud resources, is expected to be AI-enabled to provide full flexibility in resource and task orchestration for joint RAN-CN optimizations.

The semantic layer contains three enablers: (i) non-arbitrary semantic representations, (ii) dedicated interfaces to negotiate semantic-centric SLAs or SSLAs and exchange associated information and metrics, and (iii) a semantic control plane to configure network functions accordingly. Moving towards semantic communication networks involves adopting a new data representation that can be managed by ML models to meet diverse application-specific requirements.

The Distributed AI Control and Model Management layer covers AI model and AI agent runtime operations, such as model selection, policy control, and data collection. The AI Inference orchestration and services layer manage the deployment and execution of models for prediction and exhibits service exposure towards external actors. The Service layer exhibits service exposure and intent-based services towards external actors and cross-domain coordination.

4.1 Mapping between the 6GARROW Demonstrations and the Functional Architecture

In Table 2 we provide a mapping between the planned demonstrations and the 6GARROW Functional Architecture, by providing relations between the demonstrated concepts and the blocks of the Functional Architecture. In addition, we highlight the use case families addressed by each of the planned demonstrations.

Table 2: Mapping between the demonstrations, the use cases and the functional architecture

Demonstrations	Use Case Families			Cross-domain coordination and service exposure	AI inference orchestration and services	Distributed AI control and model management	Semantic layer	Infrastructure
	F1	F2	F3					
Demonstration 1	✓				✓	✓	✓	✓
Demonstration 2		✓		✓				✓
Demonstration 3			✓					✓
Demonstration 4		✓			✓	✓		✓

* Use Case Family 1 (F1): AI native 6G design empowering new network services for stakeholders.

* Use Case Family 2 (F2): Resource usage (lower layers).

* Use Case Family 3 (F3): Automation & failure recovery (network functions).

* In the context of the Use Case Families, "✓" indicates that the demonstration addresses aspects present in some of the Use Cases belonging to the Use Case Family.

* In the context of the Functional Architecture, "✓" indicates that the demonstration involves concepts in the domain of the corresponding block/layer of the Functional Architecture (see Figure 11).

4.2 A Demonstration Environment for Collaborative Edge Inference

To further illustrate the mapping between the proposed demonstrations and the functional architecture, this subsection takes Demonstration 1 as a representative example and details how its functional modules are mapped onto the different architectural layers. Based on the mapping, a more general demonstration environment for collaborative edge inference in radio access networks is presented, where the device and the edge server collaborate to jointly perform an inference task under communication and computation constraints, considering the varying network conditions. Figure 12 describes the involved blocks of a such unifying demonstration environment, targeting edge intelligence applications. As we can see from Figure 12, the involved blocks are mapped to the layers of the 6GARROW functional architecture from Figure 1.

Concrete examples covered by this demonstration environment include scenarios where mobile robots, potentially equipped with multiple sensors (e.g., RGB cameras, radar and/or LiDAR), communicate with an edge server to jointly perform perception and control under communication and computation constraints. Rather than transmitting raw sensor streams to the network, the robots first perform on-device semantic encoding, where a (potentially) lightweight AI model extracts task-relevant information in the form of compact semantic embeddings. These semantic embeddings are transmitted through the communication infrastructure to a nearby edge server, where larger foundation models or world models fuse them with digital twin data, infrastructure sensors, environment maps and multi-robot context. The semantic encoders are trained offline at a dedicated training server, based on the task and figures of merit supplied by the inference server, which retains a repository of inference models associated with the different tasks. Learning and inference are performed according to a criterion that captures trade-offs between implementation complexity/efficiency, communication overhead and inference accuracy.

The demonstration environment includes the following blocks:

- The information flow begins at the AI Inference Orchestration and Services (L1) layer, where an application task triggers the workflow orchestrator to establish an inference session. The

orchestrator creates the service context, requests the required network resources, and coordinates the inference process through the service session manager. In parallel, the UE-side monitor reports the current device capability and task execution status, while the model repository provides the available encoder-decoder models and version information required for deployment.

- At the Distributed AI Control and Model Management (L2) layer, the edge performs AI model configuration using information collected from both the UE the RAN. Device capability reports are combined with network measurements, including CQI, SNR, latency, packet loss, and available bandwidth, to determine the appropriate model parameters. The selected encoder-decoder pair is then synchronized and deployed across the UE and the edge server.
- The Semantic Layer (L3) carries the application data path. Sensor observations are first processed by the semantic encoder on the UE to generate compact semantic representations, which are packed together with session information and transmitted through the RAN transport. The RAN maps the semantic packets onto the available radio resources and provides transport feedback to the upper-layer monitoring functions. At the edge, the received semantic representations are decoded by the inference engine to generate the application output, such as a robot control command.
- Finally, the Infrastructure Layer (L4) provides the physical execution platform supporting the upper-layer functions. On the UE side, local computing resources perform sensor processing and semantic encoding before transmission through the AI-enabled physical layer. The RAN provides radio resource control and scheduling, while the edge infrastructure supplies the computing resources, inference runtime, storage, and service hosting environment required for semantic decoding and AI inference.

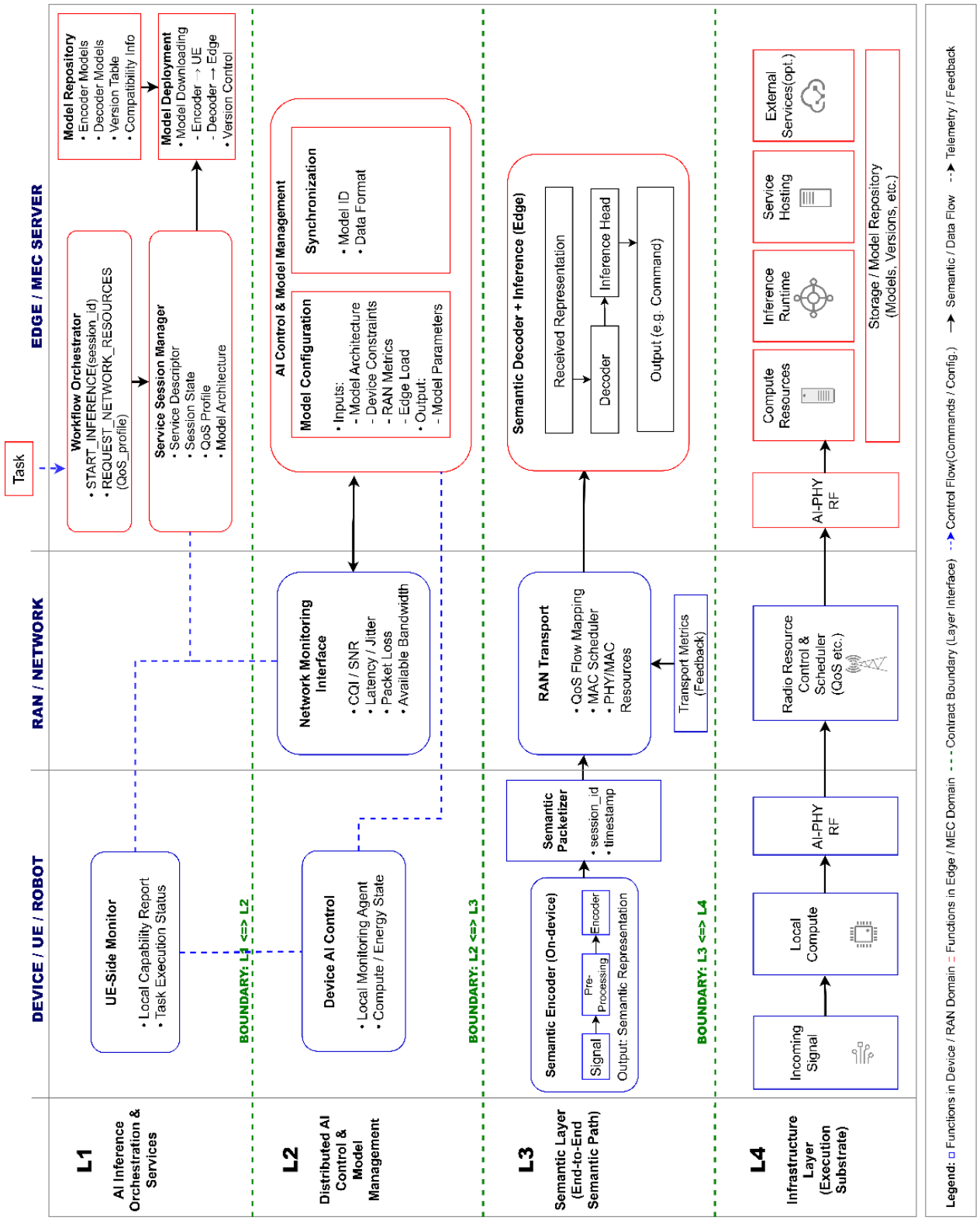

Figure 12: A unifying demonstration environment for collaborative edge inference, mapped to the 6GARROW functional architecture.

5 Conclusions

This deliverable provides details of the four laboratory demonstrations developed within WP5 of the 6GARROW project. Building upon the AI-native system architecture defined in D2.2 and the AI/ML technologies developed in WP3 and WP4, it establishes the foundation for the implementation, integration, and testing of the proposed AI/ML solutions. In this way, the deliverable directly contributes to Objective 4 of the 6GARROW project: "Develop and Validate Experimental Testbeds and Proof of Concept Platforms to Showcase the Key 6GARROW Innovations."

In addition to the definition of the planned demonstrations, this deliverable presents the relationship between the proposed demonstrations and the overall 6GARROW functional architecture. Furthermore, the proposed collaborative edge inference environment provides a detailed architectural view that illustrates the interaction between distributed AI functions, semantic communication, and wireless networking, serving as a reference architecture for future AI-native edge intelligence applications.

The descriptions presented in this deliverable provide the basis for the next phase of WP5. The proposed demonstration platforms offer controlled environments for validating AI-native technologies across the device, RAN, and CN domains, while supporting the assessment of system integration and interoperability. Future work will focus on implementing and integrating the proposed demonstration platforms, validating the developed AI/ML technologies under realistic operating conditions, and identifying potential issues while assessing their practicality and effectiveness within AI-native 6G networks.

References

[6GARROW-D21] 6GARROW, Deliverable D2.1 “6GARROW Scenarios, Use Cases and related KPIs/KVIs”, 08/2025.

[6GARROW-D22] 6GARROW, Deliverable D2.2 “Initial System Architecture”, 31/10/2025.

[6GARROW-D23] 6GARROW, Deliverable D2.3 “6GARROW Refined Scenarios, Use Cases and related KPIs/KVIs”, 05/2026.

[NV26] NVIDIA Corporation, NVIDIA Jetson Embedded Computing Platform. [Online]. Available: https://developer.nvidia.com/embedded-computing. Accessed: Jul. 22, 2026.

[GMH+24] H. A. Gonzalez, J. Huang, F. Kelber, K. K. Nazeer, T. Langer, C. Liu, M. Lohrmann, A. Rostami, M. Schöne, B. Vogginger, T. C. Wunderlich, Y. Yan, M. Akl, and C. Mayr, “SpiNNaker2: A Large-Scale Neuromorphic System for Event-Based and Asynchronous Machine Learning,” arXiv preprint arXiv:2401.04491, 2024.

[PRO26] Prophesee, Metavision Event-Based Vision Technologies. [Online]. Available: https://www.prophesee.ai/event-based-sensors/. Accessed: Jul. 22, 2026.