Distributed Artificial Intelligence-driven open and programmable architecture for 6G networks

D5.4 ADROIT6G technology validation in lab environments - Final

Document Summary Information

Project Identifier	HORIZON-JU-SNS-2022. Project 101095363	**Acronym**	ADROIT6G
Full Title	Distributed Artificial Intelligence-driven open and programmable architecture for 6G networks		
Start Date	01.01.2023	**Duration**	36 months
Project URL	https://adroit6g.eu/		
Deliverable	D5.4: ADROIT6G technology validation in lab environments - Final		
Work Package	WP5: Integration & Validation		
Contractual due date	30.11.2025	**Actual submission date**	07.05.2026
Type	Report	**Dissemination Level**	Public
Lead Beneficiary	CYENS		

Co-funded by the European Union

'ADROIT6G project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101095363. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.'

6GSNS

Responsible Author	Christophoros Christophorou (CYENS)
Contributions from	Iacovos Ioannou (CYENS), Christophoros Christophorou (CYENS), Vasos Vassiliou (CYENS), Kleanthis Neokleous (CYENS), Fotos Frangoudes (CYENS), Eleni Keliri (eBOS), Marios Sofokleous (eBOS), Carla Fabiana Chiasserini (CNIT), Antonella Molinaro (CNIT), Sergio Barbarossa (CNIT), Mounssif Krouka (OULU), Giulio Franzese (EURECOM), Ilias Chatzistefanidis (EURECOM), Navid Nikaein (EURECOM), Hans-Peter Huth (Siemens), Florian Zeiger (Siemens), Markus Sauer (Siemens), Carlos Guimarães (Siemens), Elizaveta Boriskova (Siemens), Erin Seder (NXW), Giacomo Bernini (NXW), Konstantinos Togias (ISI/ATH), Ioannis Koligliatis (ISI/ATH), Maria Trigka (ISI/ATH), Dimitris Selis (ISI/ATH), Poulecheria Zervou (ISI/ATH), Ioannis Kyriakopoulos (ISI/ATH), Anastasios Pandis (ISI/ATH), Swastika Roy (IQU), Navideh Ghafouri (IQU) , John Vardakas (IQU), Marius Iordache (ORO), Cătălin Brezeanu (ORO), Thanasis Charemis (NOVA), Juan Jose Vegas Olmos (NVIDIA), Tanel Järvet (CAFA), Märten Rannu (CAFA)
Peer reviewers	ORO & OULU

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete	Changes	Contributor(s)
V1.0	19.09.2025	1%	Initial Table of Contents	Christophoros Christophorou (CYENS), Iacovos Ioannou (CYENS), Vasos Vassiliou (CYENS), Christos Verikoukis (ISI/ATH)
V1.1	16.10.2025	5%	Baseline scenario description for all PoCs.	All partners
V1.2	20.11.2025	40%	First version of contributions of experiments' descriptions for all PoCs.	All partners
V1.3	18.12.2025	50%	Updates on experiments' descriptions for all PoCs.	All partners
V1.4	23.01.2026	60%	First version of contributions of experiments' operationalization	All partners

© ADROIT6G, 2023

			into a concrete workflow for all PoCs.	
V1.6	13.02.2026	70%	Updates on PoCs experiments' operationalization into a concrete workflow, and description of datasets generated.	All partners
V1.7	23.02.2026	80%	Updates on PoCs chapters and definition of KPIs and KVIs measurement tools.	All partners
V1.8	19.03.2026	90%	Finalization of Experiments' description for all PoCs	All partners
V1.9	03.04.2026	95%	Updates in all chapters of the deliverable.	Christophoros Christophorou (CYENS), Iacovos Ioannou (CYENS), Vasos Vassiliou (CYENS), All Partners.
N/A	07.04.2026	95%	Deliverable forwarded to ORO and OULU for review	Christophoros Christophorou (CYENS)
N/A	13.04.2026	95%	Content screening (1st review round)	Mounssif Krouka (OULU) Alexandru Oprea (ORO)
V1.10	21.04.2026	99%	Partners' updates after 1st review.	All Partners
N/A	27.04.2026	99%	QAM Review	Eleni Keliri (eBOS)
N/A	04.05.2026	99%	PC Review	Christos Verikoukis (ISI/ATH)
V1.11	05.05.2026	100%	Partners updates after QAM and PC review.	All Partners
N/A	07.05.2026	100%	Submitted Version – End deliverable creation process	Christophoros Christophorou (CYENS)

© ADROIT6G, 2023

Disclaimer

The content of the publication herein is the sole responsibility of the publishers and it does not necessarily represent the views expressed by the European Commission or its services.

While the information contained in the documents is believed to be accurate, the author or any other participant in the ADROIT6G consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the ADROIT6G 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 ADROIT6G 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

© ADROIT6G Consortium, 2023-2025. 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

Executive Summary 12
1 Introduction 14
1.1 Mapping ADROIT6G Outputs 15
1.2 Deliverable Overview and Report Structure 17
2 PoC #1 – Holographic Teaching 18
2.1 ADROIT6G Components Used in the Experiments of PoC1 18
2.2 Baseline Scenario Description 21
2.2.1 Baseline Scenario Overview (Generic Structure) 22
2.2.2 Baseline System Actors and Components 22
2.2.3 Baseline Requirements for PoC1 24
2.2.4 Baseline Limitations and 5G Challenges 24
2.3 Experiments Description 25
2.3.1 Experiment #1: Infrastructure Setup Phase 26
2.3.2 Experiment #2: Resources Setup Phase for Holographic Teaching session 33
2.3.3 Experiment #3: Real-time Immersive Holographic Teaching Communication 37
2.3.4 Experiment #3.1: MVLD-based Missing View Reconstruction (Local Simulation) 42
2.4 Application of Simulators and Emulators in Testing and Validation 43
2.5 Associated KPIs and KVIs 43
3 PoC #2a – Terrestrial 6G IIoT with extreme mMTC 48
3.1 ADROIT6G Components Used in the Experiments of PoC2a 48
3.2 Baseline Scenario Description 49
3.2.1 Baseline Scenario Overview (Generic Structure) 49
3.2.2 Baseline System Actors and Components 49
3.2.3 Baseline Requirements for PoC2a 50
3.2.4 Baseline Limitations and 5G Challenges 51
3.3 Experiments Description 51
3.3.1 Experiment #1: Federated Predictive Analytics (Application + AI/ML Level) 52
3.3.2 Experiment #2: Integrated Teleoperation and Federated Analytics with BDIx/UE-VBS (End-to-End Integration Level) 57
3.4 Application of Simulators and Emulators in Testing and Validation 60
3.5 Associated KPIs and KVIs 60
4 PoC #2b – Rail Automation Support by Non-Terrestrial Networks 65
4.1 ADROIT6G Components Used in the Experiments of PoC2b 65
4.2 Baseline Scenario Description 66
4.2.1 Baseline Scenario Overview 66
4.2.2 Baseline Requirements PoC2b 67
4.2.3 Baseline Limitations and 5G challenges 68
4.3 Experiments Description 68
4.3.1 Experiment #1: AI-Operations for Service Migration 69
4.3.2 Experiment #2: UE connection enhancement 72
4.3.3 Experiment #3: Automated selective SLA monitoring 75

© ADROIT6G, 2023

4.4 Application of Simulators and Emulators in Testing and Validation....77
4.5 Associated KPIs and KVIs....77
5 PoC #3 – Collaborative Construction Robots....80
5.1 ADROIT6G Components Used in the Experiments of PoC3....80
5.2 Baseline Scenario....81
5.2.1 Baseline Scenario Overview....81
5.2.2 Baseline System Actors and Components....81
5.2.3 Baseline requirements for PoC3....82
5.2.4 Limitations and 5G challenges....83
5.3 Experiments Description....83
5.3.1 Experiment #1: Autonomous Transport with Dynamic Re-Routing Loading....84
5.3.2 Experiment #2: Drone–AGV Collaborative working with D2D....88
5.4 Application of Simulators and Emulators in Testing and Validation....94
5.5 Associated KPIs and KVIs....94
6 Generated Datasets....98
6.1 PoC #1 Datasets Generated....98
6.1.1 Data Acquisition Parameters....98
6.1.2 Dataset Components....98
6.1.3 Data Utility and Evaluation....99
6.1.4 Dataset Specifications....99
6.2 PoC #2b Datasets Generated....99
6.2.1 Satellite Connectivity Data for Dataset generation....99
6.2.2 Application Performance Data with Background Traffic for Dataset generation....100
6.3 PoC2a and PoC #3 Datasets Generated....101
6.3.1 Description of data logged by CAFA Robot app during the experiments....102
6.3.2 Description of the files....102
7 Conclusions....104
References....106

© ADROIT6G, 2023

List of Figures

Figure 1-1: Final version of the ADROIT6G architecture. 15
Figure 2-1: PoC.1 Holographic Teaching 18
Figure 2-2: Baseline schematic representation of POC1 - Experiment 3 21
Figure 2-3: Baseline schematic representation the volumetric capture on the teacher's side 22
Figure 2-4: PoC1 Experiment #1 Infrastructure Setup Phase 26
Figure 2-5: PoC1 Experiment #1 Sequence Diagram 28
Figure 2-6: PoC1 Experiment #1 Testbed Deployment 32
Figure 2-7: PoC1 Experiment #2 Resources Setup Phase for Holographic Teaching session 33
Figure 2-8: PoC1 Experiment #2 Sequence Diagram 35
Figure 2-9: PoC1 Experiment #2 Testbed Deployment 36
Figure 2-10: PoC1 Experiment #3 - Real-time Immersive Holographic Teaching example of the teacher location setup 37
Figure 2-11: PoC1 Experiment #3 Real-time Immersive Holographic Teaching Communication 38
Figure 2-12: PoC1 Experiment #3 Sequence Diagram 39
Figure 3-1: CAFA Robot app architecture showing data and process flow 50
Figure 3-2: PoC2a Experiment #1 sequence diagram (provisioning, runtime, and baseline closed-loop adaptation) 54
Figure 3-3: PoC2a Experiment #1 testbed setup 56
Figure 3-4: PoC2a Experiment #2 end-to-end interaction flow covering distributed resource discovery, xApp-based closed-loop control, and dynamic-load validation 59
Figure 4-1: SCI-Points and Interlocking Service Signalling 67
Figure 4-2: PoC2b Experiment 1: component interaction – training phase 70
Figure 4-3: PoC2b Experiment 1: component interaction – closed loop operations phase 71
Figure 4-4: PoC2b Experiment 1 closed loop operations phase: component interaction 72
Figure 4-5: PoC2b Experiment 2: component interaction 73
Figure 4-6: PoC2b Experiment 2: sequence diagram 74
Figure 4-7: PoC2b Experiment 3: component interaction 76
Figure 4-8: PoC2b Experiment 3: sequence diagram 77
Figure 5-1: CAFA Robot app - the dashboard. 82
Figure 5-2: PoC3 experimental architecture 84
Figure 5-3: PoC3 Experiment #1 Sequence Diagram 86

© ADROIT6G, 2023

Figure 5-4: PoC3 Experiment #2 Sequence Diagram 91
Figure 6-1: Emulated experiment of Satellite Connectivity.....100

List of Tables

Table 1-1: Adherence to ADROIT6G's GA Deliverable & Tasks Descriptions 15
Table 1-2: Links to Other Project's Outputs / Work 17
Table 2-1: ADROIT6G Components used in the Experiments of PoC-1 19
Table 2-2: Specification for one stream rendering summary 24
Table 2-3: Summary of Experiments for PoC.1 during Testing Cycle 2 25
Table 2-4: PoC1 Service KPIs 43
Table 2-5: PoC1 Network KPIs 44
Table 2-6: PoC1 Service KVIs 45
Table 2-7: PoC1 Network KVIs 46
Table 3-1: ADROIT6G Components used in the Experiments of PoC-2a 48
Table 3-2: Summary of Experiments for PoC.2a during Testing Cycle 2 51
Table 3-3: ADROIT6G components involved in PoC2a Exp #1 and the role of each component in the overall storyline 53
Table 3-4: ADROIT6G components involved in PoC2a Exp #2 and the role of each component in the overall storyline 58
Table 3-5: Service KPIs for PoC2a 60
Table 3-6: Network KPIs for PoC2a 62
Table 3-7. KVIs for PoC2a 63
Table 4-1: ADROIT6G Components used in the Experiments of PoC-2b 65
Table 4-2: Summary of Experiments for PoC.2b during Testing Cycle 2 69
Table 4-3: Service KPIs for PoC2b 78
Table 4-4: Network KPIs for PoC2b 78
Table 4-5: Service KVIs for PoC2b 78
Table 5-1: ADROIT6G Components used in the Experiments of PoC-3 80
Table 5-2: Summary of Experiments for PoC.3 during Testing Cycle 2 83
Table 5-3: Service KPIs for PoC3 94
Table 5-4: Network KPIs for PoC3 95
Table 5-5: KVIs for PoC3 95

© ADROIT6G, 2023

Glossary of terms and abbreviations

Abbreviation / Term	Description
AGV	Automated Ground Vehicle.
AI	Artificial Intelligence.
AI/ML	Artificial Intelligence / Machine Learning.
API	Application Programming Interface.
AR/VR	Augmented Reality / Virtual Reality.
BDIx	Belief–Desire–Intention eXtended.
CLG	Closed-Loop Governance.
CPU	Central Processing Unit.
CS	Control Station.
CSH	Computing & Storage Hub.
CSP	Computing & Storage Powerhouse.
D2D	Device-to-Device.
E2E	End-to-End.
FL	Federated Learning.
FR1	Frequency Range 1 in 5G NR, covering sub-6 GHz spectrum.
GA	Grant Agreement.
gNodeB	Next Generation Node B; a 5G base station.
GNSS	Global Navigation Satellite System.
GPU	Graphics Processing Unit.
HD	High Definition.
ICE	Interactive Connectivity Establishment.
IIoT	Industrial Internet of Things.
IoT	Internet of Things.
KPI	Key Performance Indicator.
KVI	Key Value Indicator.
LPIPS	Learned Perceptual Image Patch Similarity.
LTE	Long Term Evolution.
M&O	Management and Orchestration.
ML	Machine Learning.

© ADROIT6G, 2023

MLD	Multimodal Latent Diffusion.
mMTC	massive Machine-Type Communications.
MOS	Mean Opinion Score.
MQTT	Message Queuing Telemetry Transport.
MVLD	Multi-view Latent Diffusion.
NIC	Network Interface Controller.
NR	New Radio.
NTN	Non-Terrestrial Network.
PoC	Proof of Concept.
QoE	Quality of Experience.
QoS	Quality of Service.
RAM	Random Access Memory.
RAN	Radio Access Network.
RGBD	Red, Green, Blue, and Depth.
RIC	RAN Intelligent Controller.
ROS2	Robot Operating System 2.
SCP	Service Communication Proxy.
SDP	Session Description Protocol.
SLA	Service Level Agreement.
SMF	Session Management Function in the 5G core.
TCP	Transmission Control Protocol.
TFLOPS	Tera Floating-Point Operations Per Second.
UDP	User Datagram Protocol.
UE	User Equipment.
UE-VBS	User Equipment as Virtual Base Station.
URLLC	Ultra-Reliable Low-Latency Communications.
VNF	Virtual Network Function.
VRo	Virtual Robot.
WiFi Direct	Peer-to-peer Wi-Fi technology for direct device-to-device communication.
XR	Extended Reality.

© ADROIT6G, 2023

Executive Summary

The work performed in Task 5.2 and presented in this deliverable represents the last critical phase of the ADROIT6G project, focusing on the final and complete validation of the ADROIT6G system components developed under Work Package (WP) 3 and WP4. This D5.4 deliverable is the second report produced under Task 5.2. It documents the technology validation activities conducted during Testing Cycle 2, marking the final stage of system integration and validation prior to the performance evaluation phase in Task 5.3.

Building upon the functional block validation activities performed in Testing Cycle 1 and reported in D5.3, which concentrated on validating smaller, cohesive groups of components, i.e., functional blocks, the work performed during Testing Cycle 2 advances to the integration of multiple ADROIT6G components across frameworks to form operational workflows. These workflows correspond to the user stories defined in D5.3, which have been translated into experiments that operationalise each story into concrete, testable scenarios designed to validate specific phases of the end-to-end Proof-of-Concepts (PoCs). Each experiment thus represents a cohesive integration path that combines different ADROIT6G components across different frameworks, correctly synchronised and capable of executing the intended logic to deliver the expected outcomes within the use case context.

The primary goal of this phase is to demonstrate that the ADROIT6G components, developed within the AI-Driven Management & Orchestration Framework, the Fully Distributed and secure AI/ML Framework for CrowdSourcing AI, and the BDI - & AI-driven Unified & Open Control Operations in WP3 and WP4, and integrated under WP5, operate seamlessly together to realise well-defined parts of the overall end-to-end (E2E) scenarios. Rather than focusing on isolated functional groups, Testing Cycle 2 evaluates the PoCs as complete demonstrations of ADROIT6G capabilities. This ensures that all components interact and communicate correctly within each PoC workflow, consistently providing the expected functionalities, and the ADROIT6G platform operates as a cohesive and stable system. Through this process, Testing Cycle 2 confirms that the PoCs are ready for quantitative performance evaluation against the project's Key Performance Indicators (KPIs) and Key Value Indicators (KVIs).

The main outcome of this deliverable is the definition, execution, and validation of the final integrated workflows for the ADROIT6G PoCs in lab environments. For each PoC, the deliverable identifies the ADROIT6G components involved, describes the baseline scenario and limitations, defines the Testing Cycle 2 experiments, presents the operational workflows and component interactions, and specifies the associated KPIs and KVIs together with the tools and methods used for their measurement. In this way, D5.4 provides a consolidated validation reference showing how the ADROIT6G technological enablers are integrated and exercised within realistic PoC-specific scenarios. In addition, the deliverable reports the datasets generated during Testing Cycle 2, including data produced in the context of PoC #1, PoC #2b, PoC #2a, and PoC #3. These datasets support the validation activities and provide the basis for subsequent analysis, AI/ML model support, and performance evaluation.

It should be noted that D5.4 focuses on the preparation and execution of the validation experiments for Testing Cycle 2 but does not include any performance evaluation results. The performance

© ADROIT6G, 2023

assessment, where the PoCs will be evaluated against the established KPIs and KVIs to demonstrate efficiency, effectiveness, and alignment with the ADROIT6G objectives, is carried out in Task 5.3 and reported in Deliverable D5.6, Performance Evaluation and Replication Guidelines – Final.

© ADROIT6G, 2023

1 Introduction

This ADROIT6G D5.4 deliverable reports on the validation efforts conducted within Task 5.2 during Testing Cycle 2, as part of the ADROIT6G technology validation process. In this final validation phase the primary focus is on evaluating the end-to-end Proof-of-Concepts (PoCs). The objective was to ensure that all ADROIT6G components and algorithms, when combined into complete PoC workflows, operated as intended and delivered the functionalities described in the user stories defined in D5.3. Thus, while Testing Cycle 1 focused on validating smaller, cohesive groups of components, Testing Cycle 2 shifted towards evaluating the PoCs as fully integrated demonstrations of ADROIT6G capabilities. This phase ensured that: i) all components interacted and communicated correctly within each PoC workflow, consistently providing the expected functionalities; and ii) the ADROIT6G platform operated as a cohesive, stable, and fully integrated system, ready to progress to the quantitative evaluation phase.

Thus, building upon the functional block validation activities reported in D5.3, where groups of components were tested together to verify their collective behaviour, performance, and ability to deliver specific functionalities within the system, the present work advances to the integration of multiple components across frameworks to form operational workflows. The validation activities reported in this deliverable focus on the integration and validation of cohesive ADROIT6G workflows, which represent the final stage of system integration prior to performance evaluation. It validates cohesive, integrated workflows that combine multiple ADROIT6G components to realise well-scoped portions of the end-to-end (E2E) PoC scenarios.

The goal of this stage is to demonstrate the transition from isolated functional block capabilities to integrated operational workflows that collectively realise a portion of the targeted E2E PoC scenarios. These workflows correspond to the user stories defined in D5.3 that served as the foundation of this validation activity, guiding the definition of the experiments that operationalise each story into concrete and testable scenarios, designed to validate specific phases of each PoC scenario. Each workflow thus represents a cohesive integration path that combines orchestration, AI-driven control, and far-edge computing functionalities. Through this validation, the project confirms that the designed interfaces, APIs, and closed-loop communication mechanisms among the frameworks (see Figure 1-1) are interoperable, correctly synchronised, and capable of executing the intended logic to deliver the expected outcomes within the use case context.

In summary, this deliverable marks the culmination of the integration and validation activities within ADROIT6G. It demonstrates the operational maturity of the developed technologies through experiment-driven validation of integrated workflows, paving the way for the quantitative performance assessment phase. It is important to note that D5.4 focuses on the preparation and execution of the validation experiments conducted during Testing Cycle 2. The deliverable does not present any performance evaluation results. The assessment of the system's ability to meet the project's Key Performance Indicators (KPIs) and Key Value Indicators (KVIs) will be carried out in Task 5.3 and documented in Deliverable D5.6 (Performance Evaluation and Replication Guidelines – Final).

© ADROIT6G, 2023

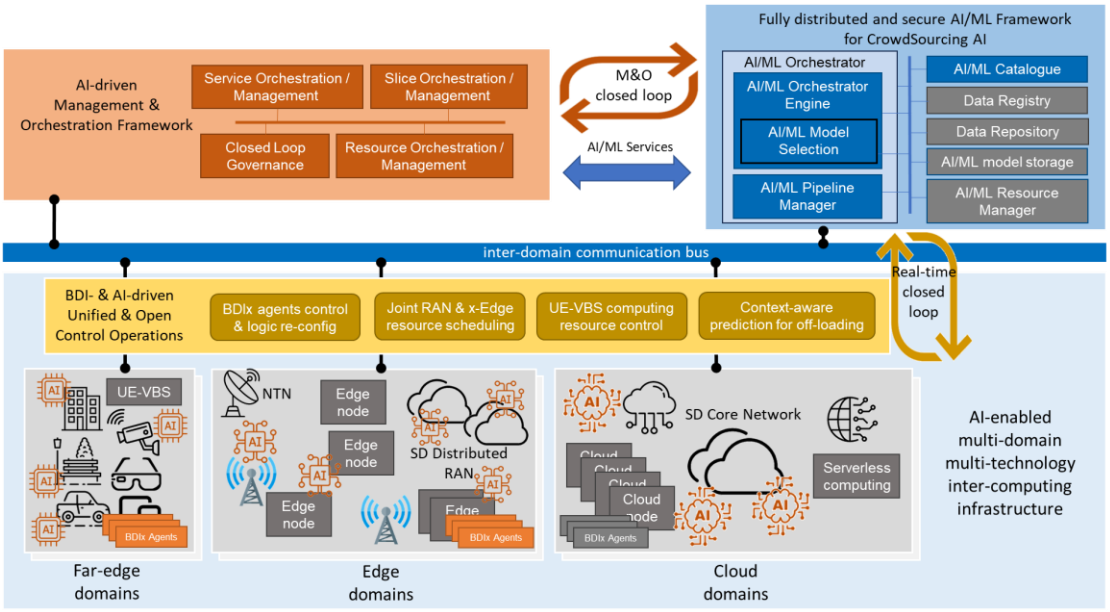

Figure 1-1: Final version of the ADROIT6G architecture.

1.1 Mapping ADROIT6G Outputs

This section presents the ADROIT6G's Grant Agreement (GA) commitments, as defined in the project's formal task descriptions, with respect to their outputs and work to be performed. Specifically, Table 1-1 describes how the work performed during Testing Cycle 2 (TC2) and presented in this deliverable, aligns with Task 5.2 of WP5, Integration & Validation, and the broader project objectives.

Table 1-1: Adherence to ADROIT6G's GA Deliverable & Tasks Descriptions

ADROIT6G GA Component Title	ADROIT6G GA Component Outline	Respective Document Chapter(s)	Justification
DELIVERABLE			
D5.3: ADROIT6G technology validation in lab environments - Initial Report containing the intermediate results of ADROIT6G performance validated through the use cases in lab settings.			
TASKS			
Task 5.2: Technology validation in lab environments	In this task, the four use cases, refined in Task 2.1 will be used to validate, through **four relevant PoCs** in specific testbeds, the ADROIT6G performance improvement with respect to 5G, its innovative features and capabilities in a lab environment. The **necessary end-user devices**,	**Chapter 2, Chapter 3, Chapter 4, and Chapter 5**	**Chapter 2 to Chapter 5** focuses on the validation efforts conducted during Testing Cycle 2, which serves as the final critical phase on validating the end-to-end PoCs. The aim was to ensure that all ADROIT6G components and algorithms, when combined into complete PoC workflows, operated as intended. This ensured that: i) all

	emulators and **traffic simulators** will also be setup, configured and interconnected with the ADROIT6G architecture to enable the execution of the use cases from end to end. Emphasis will be put on technical aspects, such as e.g., **interoperability** of **components**, use of **suitable data and generation of PoC settings**. **Synthetic data** will be produced as needed to **supply the training** of the **AI algorithms** and the tests in the initial absence of live data streams. **Any security issues identified in the PoCs will be logged and analysed**.		components interacted and communicated correctly within the PoC workflows, consistently providing the expected functionalities; and ii) the ADROIT6G platform was validated as a cohesive and stable system providing the final confirmation that PoCs are ready for large-scale performance evaluation against the project's KPI and KVIs in Task 5.3.
	Two open datasets will be **produced**: (i) One in PoC1 to include full-body RGBD images of an instructor, the skeleton data, and the camera the data were collected from, along with the merged 3D Point Cloud; (ii) The other in PoC2a and PoC3 to validate the distributed and energy-efficient AI/ML approaches of ADROIT6G. The dataset in PoC2a and PoC3 will contain, for example, the number and type of robots, their actions, their position, the CPU and memory consumption of the virtual network functions (VNFs) used to control the robots, the robots' application latency, the packet error rate, the number of instances of the VNFs. *The roles of each of the involved participants is presented in the PoC descriptions in Section 1.4.3.2 of the DoW. All technical partners (ATH, CNIT, EURE, OULU, MLNX, NOVA, ORO, EBOS, IQU, NXW) will support the use case partners (CYENS, SIEMENS, CAFA) by providing technical support for the testbeds they provide as well as for the architectural elements they will co-develop. (Output: D5.3, D5.4)*	**Chapter 6**	**Chapter 6** reports on the generated datasets during Testing Cycle 2.

1.2 Deliverable Overview and Report Structure

The remaining document is structured as follows:

- **Chapter 2** focuses on the validation efforts, specific to PoC1, undertaken during TC2.
- **Chapter 3** focuses on the validation efforts, specific to PoC2a, undertaken during TC2.
- **Chapter 4** focuses on the validation efforts, specific to PoC2b, undertaken during TC2.
- **Chapter 5** focuses on the validation efforts, specific to PoC3, undertaken during TC2.
- **Chapter 6** describes the datasets generated during TC2.
- **Chapter 7** draws the main conclusions.

Table 1-2 provides a summary of the linkage to other project outputs, specific deliverables, and their respective contributions.

Table 1-2: Links to Other Project's Outputs / Work

WP #	Task #	Deliverable No related	Content
WP2	Task 2.1 – Task 2.4	D2.3	D2.3 describes the final ADROIT6G architecture design and the PoCs definition, considered in D5.4.
WP3	Task 3.1 – Task 3.4	D3.2 and D3.3	Technologies validated in D5.4 originate from WP3.
WP4	Task 4.1 – Task 4.4	D4.1 and D4.2	Technologies validated in D5.4 originate from WP4.
WP5	Task 5.1	D5.2	D5.2 reports the integration and deployment status of testbeds, preparing the PoCs for Testing Cycle 2 and enabling the end-to-end Proof-of-Concepts (PoCs)' validation performed in D5.4.
WP5	Task 5.3	D5.6	Insights from D5.4 support performance evaluation. Specifically, D5.4 provides intermediate validation results to D5.6 that will further critically analyse them against the project's Key Performance Indicators (KPIs) and Key Value Indicators (KVIs).
WP6	Task 6.1 – Task 6.3	D6.3	D5.4 validation results can contribute to dissemination and communication activities and may support contributions to standardisation.

© ADROIT6G, 2023

2 PoC #1 – Holographic Teaching

This chapter focuses on the validation efforts specific to PoC1 (see Figure 2-1), conducted during Testing Cycle 2. It provides an overview of the PoC's experiments, their specific objectives and their associated KPIs and KVIs, already defined in D2.3 [1]. Prior to the execution of each experiment, the execution plan and detailed validation steps were specified, guaranteeing the correct configuration of all required components and ensuring the reliability of the experiment setup. Specifically, for each experiment all the details regarding how the scenario will be executed from end-to-end, as well as specifying the whole flow of interaction between the different components that take part in the specific experiment, are described.

Additionally, the application of simulators and/or emulators during testing and validation (where applicable), including details on the setup of devices, and synthetic data generation used to support testing and AI training, are provided. Finally, the specific KPIs and KVIs used to evaluate performance (as those described in Chapter 8 of D2.3 [1]) are provided. In addition, for each KPI and KVI the tools and methods used to measure and assess the performance are explicitly described.

The outcomes of PoC1's technology validation activities are reported in D5.6, evaluated against the specific KPIs and KVIs presented in section 2.5.

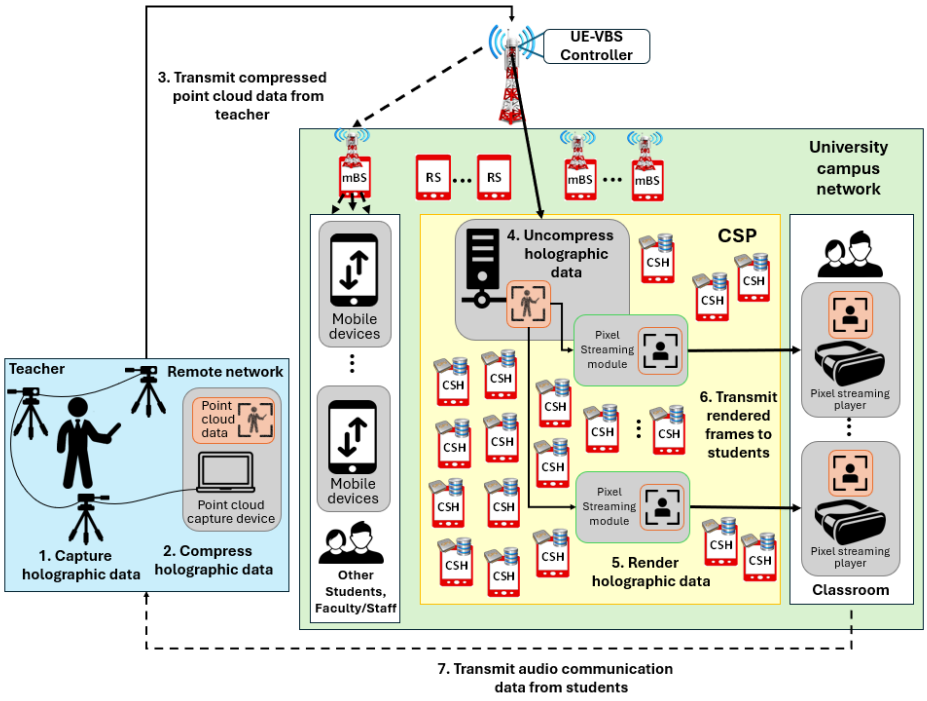

Figure 2-1: PoC.1 Holographic Teaching

This PoC involves capturing, transmitting, processing, and rendering high-quality 3D holographic content (point cloud data) in real-time using the ADROIT6G architectural elements and the local UE-VBS computing resources formed by harnessing the networking, storage, and processing resources of mobile devices.

2.1 ADROIT6G Components Used in the Experiments of PoC1

Table 2-1 below provides a view of the ADROIT6G components that are used in PoC1 and maps each of them explicitly to the three PoC1 experiments. Its purpose is to make clear, for each experiment,

which ADROIT6G components are involved in the corresponding operational workflow, across the different frameworks. In doing so, the table clarifies for each experiment the specific building blocks that must be deployed on the ISI/ATH testbed and correctly interconnected to execute the operational workflow end-to-end. In the context of D5.4, the table also functions as a traceability reference for the PoC1 validation activities. By showing which components are actually used per experiment, it supports coordination among component owners and testbed team, and serves as a reference for aligning the operational workflow descriptions presented in the following sections with the set of components expected to be involved in each experiment.

Table 2-1: ADROIT6G Components used in the Experiments of PoC-1

PoC Component	Exp. 1	Exp. 2	Exp. 3
AI-Driven Management & Orchestration Framework			
Closed-loop Governance (CLG)	X	X	X
Resource Orchestration & Management • SLA-driven Adaptive Learning Rate-based Federated Learning Policy for Slice-level Resource Provisioning **Reason for excluding this component from PoC1:** The algorithm is not included in PoC1, as no technically relevant integration point exists within the current PoC1 experiment workflows. In the previous Testing Cycle 1 (Test Case 3.2 in Deliverable 5.3), this solution was used for FL-driven slice-level computing resource prediction in a standalone manner. However, it does not handle actual resource allocation and orchestration, which are the primary objectives of the current PoC1 workflows, and therefore does not fully align with the POC1 user stories. Based on the description, the solution under **Service/Slice Orchestration & Management** can support slice creation based on the requested service type and allocate resources according to slice requirements. This component aligns with the current POC1 user stories and is sufficient for Testing Cycle 2 demonstrations. At the same time, it also corresponds to the objectives of the **Resource Orchestration & Management** component as defined in the Adroit6G architecture.	**Excluded from PoC1 validation** Note that, this algorithm is being validated in PoC2b (Experiment #3), where its role and measurable impact are clearly defined.		
Service Orchestration & Management	X	X	
Slice Orchestration & Management • RL for Dynamic Service/Slice Orchestration and Management		X	
Fully Distributed & Secure AI/ML Framework for CrowdSourcing AI			
AI/ML Pipeline Manager	X	X	

© ADROIT6G, 2023

<table>
<tr><td>AI/ML Catalogue</td><td>X</td><td>X</td><td></td></tr>
<tr><td>AI/ML Orchestrator Engine - Multimodal Diffusion Models for Synthetic Data Generation: Multimodal Latent Diffusion (MLD) & Multi-view Latent Diffusion (MVLD) methods</td><td></td><td></td><td>X</td></tr>
<tr><td>AI/ML Models for Predictive Monitoring</td><td>X</td><td></td><td></td></tr>
<tr><td>AI/ML Orchestrator Engine – ML model compression through Quantization for efficient and faster training using Collaborative Learning: G-DNGAM and DIN algorithms

Reason for excluding this component from PoC1: The algorithm is not included in PoC1, as no technically relevant integration point has been identified within the current PoC1 experiment workflows. In the previous Testing Cycle 1 (Test Case 2.3 in deliverable 5.3 [2]), the algorithm was used to support the evaluation of another component’s KPIs (Pipeline Registration and Execution time), rather than to validate its own performance. The component’s primary contribution concerns energy-efficient distributed ML training through quantization, with Energy Efficiency as the main KPI. This objective does not align with the scope of the current PoC1 experiments.</td><td colspan="3">Excluded from PoC1 validation.

Note that the algorithm is being validated in PoC2a (Experiment #1) and PoC2b (Experiment #1), where its role and measurable impact are clearly defined, including validation on two distinct datasets: one publicly available benchmark dataset and one industrial dataset provided within PoC2b, ensuring a robust assessment of its performance</td></tr>
<tr><td>AI/ML Orchestrator Engine/ AI/ML Model Selection</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>• MERGE: Meta-learning approach for Model selection, creation & adaptation

Reason for excluding MERGE from PoC1: MERGE currently does not fit in any of the planned experiments for PoC1 since there are not different versions of the ML model trained with statistically different datasets. Without such datasets, it is not possible to create different versions of a ML model that make sense for MERGE.</td><td colspan="3">Excluded from PoC1 validation.

Note that the algorithm is being validated in PoC3. In PoC3, different versions of a model available in the literature (e.g., DeepBeam) can be stored in the model repository and used to validate the functional pipeline.</td></tr>
<tr><td colspan="4">UE Side</td></tr>
<tr><td>BDIx Agents</td><td>X</td><td>X</td><td></td></tr>
</table>

© ADROIT6G, 2023

BDI - & AI-driven Unified & Open Control Operations			
BDIx Agents Control & Logic Re-config (Real-Time Close Loop)	X		
UE-VBS Computing Resource Control (Real-Time Close Loop).	X	X	X
Joint RAN & x-Edge Resource scheduling **Note:** This component is implemented on the **Topology Module** and on the **BDIx agents**.	X		X

2.2 Baseline Scenario Description

The baseline scenario represents the **state-of-the-art implementation of the Holographic Teaching PoC without the use of the ADROIT6G infrastructure**. The aim is to establish a **reference point** against which the benefits of the ADROIT6G-enabled solution can be evaluated.

Its purpose is to establish a baseline configuration against which the performance, scalability, and Quality of Experience (QoE) improvements enabled by the ADROIT6G system can be evaluated. The baseline scenario exposes inherent architectural limitations related to latency, bandwidth consumption, centralized processing, and scalability, which will later be contrasted with the results of controlled emulations performed within the ADROIT6G testbed.

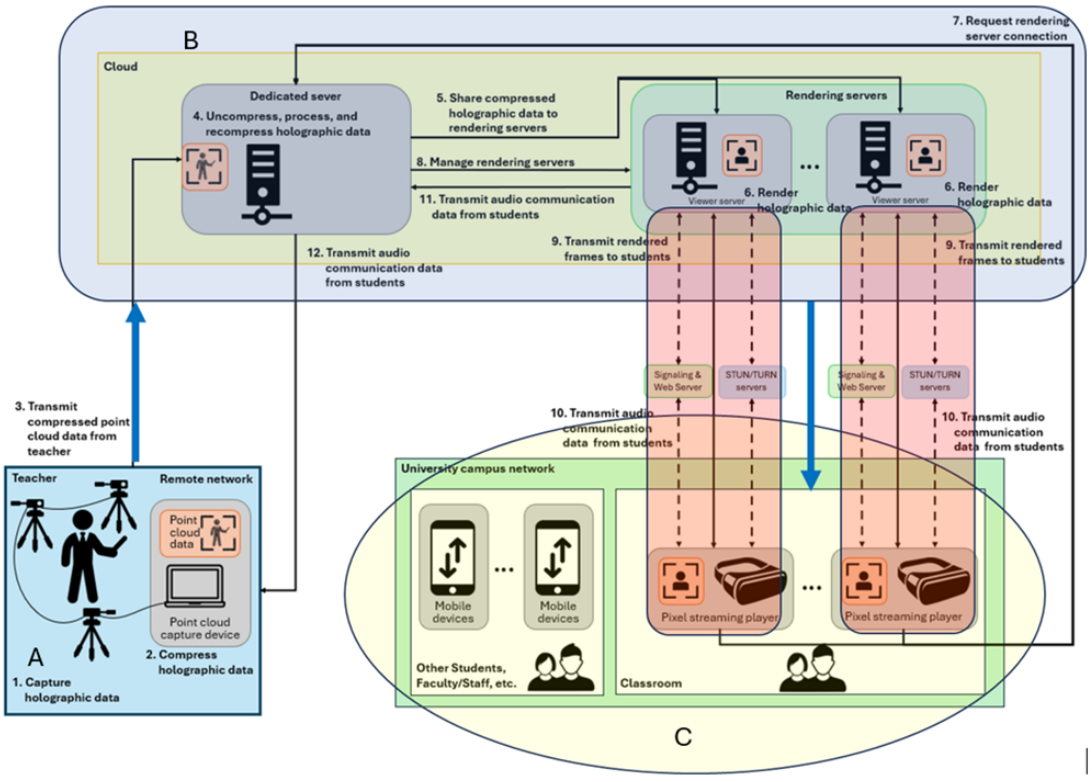

Figure 2-2: Baseline schematic representation of POC1 - Experiment 3

2.2.1 Baseline Scenario Overview (Generic Structure)

In the baseline scenario the following will be done (see also Figure 2-2):

- **Data capture and application logic** will be performed at the source (for example where the teacher will be; Point A)
- **Compute-intensive processing and rendering** are performed in a **centralized cloud environment** (Point B)
- **Processed content** is streamed from the cloud to end users (Point C)
- **Scalability** is achieved through cloud replication

This architecture is representative of **current cloud-centric XR and immersive communication systems** as can be seen in Figure 2-2 below.

2.2.2 Baseline System Actors and Components

This baseline scenario represents **current practice** in cloud-based immersive holographic teaching and will serve as a **benchmark** for latency, bandwidth consumption, scalability and KPI/KVI metrics. The objective is to present and identify **structural limitations** that cannot be overcome without the ADROIT6G infrastructure.

Figure 2-3: Baseline schematic representation the volumetric capture on the teacher's side

2.2.2.1 Actors and System Components

Teacher Side (point A in Figure 2-2**)**

The teacher location acts as the **content capture and uplink endpoint** of the holographic teaching service with the following setup:

- Instructor located in a dedicated teaching space
- Multi-camera volumetric capture system (Microsoft Kinect RGB + depth cameras)
- Local capture workstation for synchronization and encoding
- Standard internet connectivity (fiber or 4G/5G)

© ADROIT6G, 2023

Cloud Infrastructure (Point B in Figure 2-2**)**

The centralized cloud represents the **core processing and rendering domain** of the baseline scenario with the following setup:

- Centralized cloud data center
- GPU-enabled virtual machines for:
  - 3D reconstruction
  - Rendering of holographic views
- Media servers for streaming rendered content

Students Side (Point C in Figure 2-2**)**

The student location represents the **content consumption and interaction endpoint** with the following setup:

- Multiple students equipped with AR/VR headsets
- Each student connected individually to the cloud
- Standard consumer network connectivity
- Optional audio/video communication via third-party platforms (e.g., Microsoft Teams)

Baseline Architectural Characteristics:

- **Centralized processing**: All heavy computation occurs in the cloud
- **Long network paths**: Data traverses wide-area networks multiple times
- **Cloud-driven scalability**: Each additional student requires:
  - Additional rendering workload
  - Additional streaming resources

Baseline End-to-End Information Flow:

The end-to-end information flow in the baseline scenario is as follows:

1. The instructor's motion and appearance are captured at the teacher location using multiple RGB and depth cameras.
2. Volumetric data is synchronized, pre-processed, and compressed locally.
3. Compressed volumetric streams are transmitted over the public network to the centralized cloud.
4. The cloud performs real-time 3D reconstruction and holographic rendering.
5. Multiple viewpoint-dependent streams are generated, one per student.
6. Rendered and encoded streams are transmitted from the cloud to each student's AR/VR device.
7. Audio communication between teacher and students is handled via conventional conferencing services operating in parallel.

This flow requires multiple long-distance network traversals and centralized processing stages.

© ADROIT6G, 2023

Summary of Baseline Processing, Rendering, and Networking Requirements:

For each **personalized holographic stream**, the centralized cloud must allocate dedicated compute, memory, and network resources. summarizes indicative requirements per stream for different quality levels.

Table 2-2: Specification for one stream rendering summary

Aspect	Specification (for one stream rendering)		
Resolution	**1280x720 (HD)**	**1920x1080 (Full HD)**	**3840x2160 (4K UHD)**
Number of Views	3 (180° coverage)	3 (180° coverage)	3 (180° coverage)
GPU Compute Power	~10 TFLOPS per view (~30 TFLOPS total)	~40 TFLOPS per view (~120 TFLOPS total)	~80 TFLOPS per view (~240 TFLOPS total)
GPU Memory (VRAM)	At least 8 GB	At least 16 GB	At least 24 GB
CPU Compute Power	Equivalent to ~37.5 GHz total	Equivalent to ~150 GHz total	Equivalent to ~300 GHz total
RAM (System Memory)	32 GB	64 GB	128 GB
Storage	500 GB NVMe SSD	1 TB NVMe SSD	1 TB+ NVMe SSD
Networking	2–5 Gbps	5–10 Gbps	10+ Gbps
Power Consumption	~0.8–1.0 kW	~1.5 kW	~2.5 kW

As the number of students increases, these requirements scale **linearly**, resulting in significant cumulative compute and bandwidth demand.

2.2.3 Baseline Requirements for PoC1

The Real-time Immersive Holographic Teaching application imposes the following requirements:

- **Low end-to-end latency:** Target: < 300 ms for acceptable holographic interaction.
- **High computational capacity:** At least 10 TFLOPs required for real-time volumetric reconstruction and multi-view rendering.
- **High downlink data rates:** Approximately ~85 Mbps per student for compressed 720p-based holographic streams.
- **Network scalability:** Support for up to ~50 concurrent students receiving personalised streams. Each student receives one of the 10 different available streams, based on their virtual location within the 3D environment/classroom.

2.2.4 Baseline Limitations and 5G Challenges

The baseline scenario exposes **fundamental limitations** when deployed over conventional 5G and cloud-centric infrastructures:

- **Insufficient far-edge computing capabilities**, forcing all processing into centralized cloud servers.
- **High end-to-end latency** due to volumetric data upload, cloud processing, and downstream streaming.
- **Scalability constraints**, as personalized rendering pipelines must be replicated per student.
- **Excessive bandwidth consumption**, leading to congestion in radio access and core networks.

© ADROIT6G, 2023

- **QoE degradation** for both holographic teaching participants and other users sharing the same network infrastructure.

2.3 Experiments Description

This section outlines the experiments whose goal is to drive the validation efforts planned for this PoC. These experiments were agreed upon during structured PoC workshops held on 18th of November 2025 as part of the ADROIT6G plenary meeting in Copenhagen (Denmark), and were subsequently finalised in a series of PoC1 technical meetings, co-led by the respective Testbed owner (ISI/ATH) and PoC owner (CYENS) with the active participation of all involved component owners (NXW, IQU, CNIT, NOVA, EURE and OULU). Each experiment includes a clear description of its objective, its operationalization into a concrete workflow describing the whole flow of interaction between the different components that take part in the experiment, and the description of testbed set up.

Table 2-3 summarizes the final set of experiments for PoC1, whose goal is to drive the validation efforts planned for this PoC.

Table 2-3: Summary of Experiments for PoC.1 during Testing Cycle 2

Experiment	Objective	Associated with User Story (described in D5.3)
#1 Infrastructure Setup Phase (Network Level)	The objective of this experiment (see section 2.3.1) is to validate the automated infrastructure initialization and configuration process that occurs when a new User Equipment (with a UE-VBS Component installed and a BDIx running of the Device) connects to the ADROIT6G network. The goal is to ensure that the AI-driven Management and Orchestration (M&O) framework, the Crowdsourcing AI framework, the BDIx agents, and UE-VBS Computing Resource Control collaboratively enable onboarding of devices into the network. This process includes the detection of new devices, deployment of the relevant ML models, configuration of orchestration policies, and initialization of the UE-VBS components. This experiment establishes the foundational setup phase for all subsequent tests within PoC #1, ensuring that every connected UE can dynamically integrate as a potential UE-VBS and augment connectivity and computing infrastructure in the far-edge domain.	User Story #0
#2 Resource Setup Phase for the Holographic Teaching session (Network + Application Level)	The objective of this experiment (see section 2.3.2) is to validate the automatic resource observation and prediction workflow and orchestration mechanisms required to support a real-time holographic teaching session. Specifically, it aims to demonstrate how the ADROIT6G system detects the start of a session, monitors computational and network conditions, and allocates the required resources in slices ensuring the required QoS for the Holographic Teaching session. This experiment bridges the gap between UE onboarding (validated in Experiment #1) and real-time holographic teaching (Experiment	User Story #1

© ADROIT6G, 2023

	#3), confirming that predictive insights from the FL-based workflow enable the system to adapt dynamically to the demands of immersive XR-based holographic teaching.	
#3 **Real-time Immersive Holographic Teaching and** (Application Level)	The objective of this experiment (see section 2.3.3) is to validate the ability of the ADROIT6G system to support, at application level, a real-time immersive holographic teaching session with stringent end-to-end performance requirements. In this experiment, the holographic teaching application at the Teacher's location is responsible for capturing and encoding the instructor's 3D holographic representation. The role of ADROIT6G is to ensure that these are efficiently transmitted and delivered to the students over the underlying 6G infrastructure in a way that preserves real-time interactivity and high Quality of Experience (QoE). Another key aspect investigated in this experiment is scalability. As the number of concurrently connected students increases in the classroom, multiple viewpoint-dependent rendering streams will need to be generated, where each student (or group of students) receives a personalized point of view that reflects their position and orientation relative to the holographic instructor. The experiment will therefore assess the capacity of ADROIT6G to sustain this scalability while still meeting the stringent latency, bandwidth, and QoE requirements of real-time XR-based holographic teaching.	User Story #2 & User Story #3

2.3.1 Experiment #1: Infrastructure Setup Phase

This experiment verifies the **infrastructure setup capability** of the ADROIT6G system. The objective is to validate that the ADROIT6G architecture can automatically **onboard** and **prepare** the **UEs** so that they become **UE-VBS-capable**, be correctly configured (e.g., with the required **policies, behavioural rules, AI models, and orchestration integration)**, and contribute their resources to the distributed far-edge intelligence infrastructure. This experiment establishes the foundational setup phase required for Experiments #2 and #3.

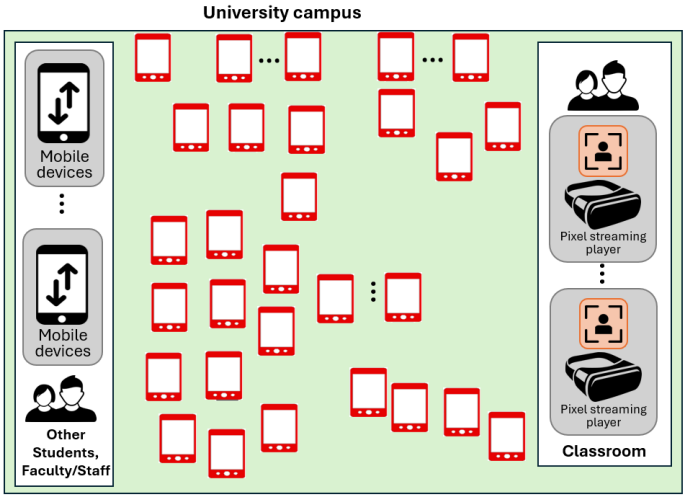

Figure 2-4: PoC1 Experiment #1 Infrastructure Setup Phase

UEs integrating as UE-VBSs and augment connectivity and computing infrastructure in the far-edge domain

The experiment starts when a **new UE device,** equipped with a **UE-VBS Component** and **a BDIx Agent**, connects to the ADROIT6G 6G network. The **BDIx Agent**, which is pre-installed on the UE, *detects the connection/attach event* and initiates the onboarding process *sending a message* (i.e., an informative request message with device and network details) to the **Topology module.** From this point onward, the setup proceeds in an automated sequence coordinated among the **Crowdsourcing AI framework** (AI/ML Pipeline Manager, AI/ML Catalogue), **Topology Module**, **BDIx Agents**, the **UE-VBS Computing Resource Control**, and the **Closed-Loop Governance** (CLG).

The complete workflow sequence for this infrastructure setup phase is described in section 0.

2.3.1.1 Operationalization of experiment into a concrete Workflow

This section explains the complete workflow that operationalises this experiment. It explains, from **start trigger** to **end,** the overall storyline as a sequence of actions that occur when a new device connects to the 6G network. The workflow shows how the architecture onboards and prepares a UE to act as a UE-VBS with minimal manual intervention, starting from its initial detection and progressing through model deployment, policy configuration, UE-VBS setup, and full integration into the ADROIT6G system, so the UE can participate as a UE-VBS node in subsequent holographic teaching experiments (Experiment #2 and Experiment #3). The sequence diagram presented in section 2.3.1.2 complement this narrative by formalizing the workflow at the interface level (i.e., specifying the concrete inter-component interactions, interfaces/APIs invoked, and key messages/payloads exchanged).

The process begins in **Step 1**, when the device running the **BDIx Agent** joins the 6G network. Once joined, the **BDIx agent** sends a message (i.e., an informative request message) with device and network details to the **Topology module**, which is under the BDIx agent's concept. After receiving the message, the **Topology module** detects the new device that is connected to the network and **registers (logs) the device** in the ADROIT6G system. As part of this initial action, the UE device is **registered in the Topology Module**, creating the initial UE entry (e.g., ID, serving BS, basic capabilities and initial localization). *This initial detection and registration event is what triggers the automated onboarding pipeline for the new UE.*

In **Step 2a, the UE (through the BDIx agent) sends a request message** (referred to as a detection and registration event) to the **Closed-Loop Governance (CLG)** to obtain device-specific policies, plans, and behavioural rules for the device to use to support reliable XR streaming (in subsequent PoC1 experiments).

In Step 2b, the **BDIx agent** interacts with the **AI/ML Catalogue** (through the **AI/ML Pipeline Manager** that exposes the REST APIs) to receive the proper AI/ML model (**Note:** the specific AI/ML is already pretrained and available in the **AI/ML Catalogue**). In addition, the **Prediction Monitoring tool** manages to generate forecasts upon the time-series data of the testbed analytics. The workflow of the Prediction Monitoring Tool can be operationalised into a coherent end-to-end AI-driven orchestration workflow that connects monitoring, forecasting, system controls and adaptive resource management, while directly supporting the Holographic Teaching PoC1. This AI/ML component generates foresights on the future behaviour of network services, infrastructure performance indicators in the form of time series, using InfluxDB. The workflow begins with the

selection of AI models and monitoring indicators from InfluxDB, and time-series databases. After preprocessing and feature forecasting models, generate predictions of network behaviour. These predictions are available to user/testbed for enabling proactive scaling, resource rebalancing and policy enforcement actions, as defined by relevant components and APIs.

In **Step 3**, (triggered by the BDIx agent) the **UE-VBS component** on the device is configured, taking into account device capabilities and current network conditions, enabling the device to **participate as a UE-VBS** in later PoC1 experiments. This configuration **enables the UE** to **contribute compute** and **networking resources** as part of the far-edge infrastructure and prepares it for role assignment in later experiments.

Then, in **Step 4**, the CLG delivers all configuration files to **the BDIx agent at the UE,** and the **BDIx agent** synchronizes localization information with the **Topology module.**

Finally in **Step 5**, the **Topology Module** registers the new onboarded UE-VBS and accounts its network and compute resources within the overall ADROIT6G resource pool. This ensures that the newly onboarded UE-VBS-capable node can be dynamically considered in resource allocation and role assignment actions in subsequent experiments.

Through these five steps, the workflow ensures that the new UE becomes a **UE-VBS** and is able to support the subsequent holographic teaching and XR-oriented experiments defined for PoC1.

2.3.1.2 Sequence Diagrams

The step-wise workflow described above explains the experiment at a procedural level (i.e., what happens in each step and the role of each component in the overall storyline).

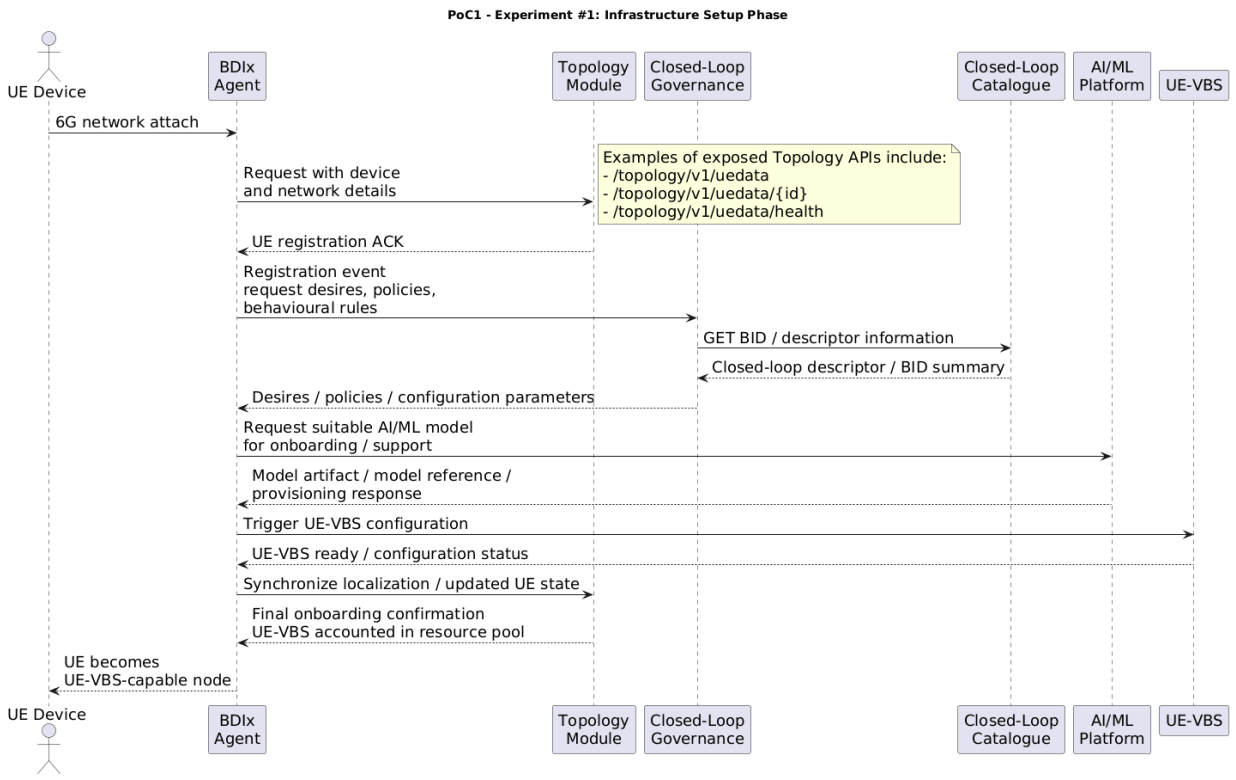

Figure 2-5: PoC1 Experiment #1 Sequence Diagram

The **sequence diagram** presented in Figure 2-5 complements this narrative by depicting the **same workflow from an interaction perspective,** providing an implementation-oriented reference for testbed integration**.** Consequently, the diagrams help validate the concrete interfaces (APIs) and dependencies required for implementation on the testbed and provide a clear reference for integration and troubleshooting during experiment execution.

2.3.1.3 Testbed Setup/Configuration

This subsection documents the concrete ISI/ATH testbed configuration for Experiment #1, based on the actual deployment state of the Kubernetes cluster. It captures the live topology, including namespaces, deployed workloads, service exposure, and monitoring components.

A consolidated view of the deployment and communication model is presented in Figure 2-6.

2.3.1.3.1 Deployed components

The following ADROIT6G components are deployed and operational in the ISI/ATH testbed:

- **BDIx Agents + Topology Module** (including UE-VBS Computing Resource Control)
- **Closed-Loop Governance (CLG)**
- **AI/ML Platform components** (Catalogue, Pipeline Manager, orchestration services)
- **5G Core (Open5GS)**

2.3.1.3.2 Namespaces and deployed workloads

The deployment is organised across multiple Kubernetes namespaces, reflecting functional separation:

- **bdix-agent**: BDIx agents (agent1–3) and topology module
- **closed-loop-functions**: CLG analytics (analysis-scl-pmp)
- **ml-platform**: AI/ML services (FastAPI, MLflow, MySQL, Postgres, Prefect, Nexus, MinIO, rec-exec, nginx)
- **odin**: orchestration services (cl-*, service-lcm, resource-allocation, rec-exec, vsb-catalogue, postgres, emqx, nginx)
- **open5gs-core**: 5G Core functions (amf, smf, upf, ausf, udm, udr, pcf, nrf, nssf, scp, mongo, webui, gnb1, ue)
- **default**: supporting services (FL clients/servers, monitoring stack)
- **ingress-nginx**: ingress controller
- **kube-system**: Kubernetes system services

2.3.1.3.3 Workload inventory

This subsection provides a consolidated view of the deployed workloads in the ISI/ATH testbed.

BDIx Layer (bdix-agent namespace):

- **BDIx Agents**
  - Deployments: agent1, agent2, agent3
  - Service exposure: ClusterIP 8088/TCP

- **Topology Module**
  - Deployment: topology
  - Service exposure: ClusterIP 8080/TCP

Closed-Loop Governance (closed-loop-functions namespace):

- **CLG Analytics Component**
  - Deployment: analysis-scl-pmp
  - Service exposure: ClusterIP

AI/ML Platform (ml-platform namespace):

- **FastAPI Service**
  - Deployment: ml-platform-fastapi
  - Service exposure: ClusterIP 8000/TCP
- **MLflow**
  - Deployment: ml-platform-mlflow
  - Service exposure: ClusterIP 5000/TCP
- **MySQL Database**
  - Deployment: ml-platform-mysql
  - Service exposure: ClusterIP 3306/TCP
- **PostgreSQL Database**
  - Deployment: postgres
  - Service exposure: ClusterIP
- **Execution Engine**
  - Deployment: rec-exec
  - Service exposure: ClusterIP 8080/TCP
- **NGINX API Gateway**
  - Deployment: nginx
  - Service exposure: ClusterIP 80/TCP
- **Supporting Services**:
  - minio — object storage
    - Deployment: minio
    - Service exposure: ClusterIP 9000/TCP
  - nexus — artifact repository
    - Deployment: nexus
    - Service exposure: ClusterIP 8081/TCP
  - prefect — workflow orchestration
    - Deployment: prefect
    - Service exposure: ClusterIP 4200/TCP

Orchestration Layer (odin namespace):

- **Catalogue & Coordination Services**:
  - Closed Loop Catalogue
    - Deployment: cl-catalogue

        - Service exposure: ClusterIP 8080/TCP
    - Closed Loop Coordinator
        - Deployment: cl-coordinator
        - Service exposure: ClusterIP 8080/TCP
    - VSB Catalogue
        - Deployment: vsb-catalogue
        - Service exposure: ClusterIP 8085/TCP
- **Lifecycle & Resource Management**:
    - Closed Loop Life Cycle Management
        - Deployment: cl-lcm
        - Service exposure: ClusterIP 8080/TCP
    - Service Life Cycle Management
        - Deployment: service-lcm
        - Service exposure: ClusterIP 8080/TCP
    - Resource Allocation
        - Deployment: resource-allocation
        - Service exposure: ClusterIP 8080/TCP
- **Execution Engine**:
    - Deployment: rec-exec
    - Service exposure: ClusterIP 8080/TCP
- **Databases, Storage & Messaging**:
    - Postgres relational DB
        - Deployment: postgres
        - Service exposure: ClusterIP 5432/TCP
    - MQTT broker
        - Deployment: emqx
        - Service exposure: ClusterIP 1883/TCP, 18083/TCP
    - Minio
        - Deployment: minio
        - Service exposure: ClusterIP 9000/TCP
- **NGINX Gateway**
    - Deployment: nginx
    - Service exposure: NodePort 30000/TCP

5G Core (open5gs-core namespace):

Each network function is deployed as a separate Kubernetes Deployment:

- **Core Network Functions**:
    - amf (ClusterIP 38412/TCP), smf (ClusterIP 7777/TCP), upf (ClusterIP 2152/UDP, 8805/UDP), ausf, udm, udr, pcf, nrf, nssf, scp
- **Mongo Database**:
    - Deployment: mongo
    - Service exposure: ClusterIP 27017/TCP

- **Access & Simulation**:
  - gNB emulator
    - Deployment: gnb1
  - UE emulator
    - Deployment: ue
- **Management & UI**:
  - Deployment: webui
  - Service exposure: ClusterIP 3000/TCP
- **Monitoring Stack**
  - **Prometheus** — metrics collection
    - Deployment: prometheus
    - Service exposure: ClusterIP 9090/TCP
  - **Grafana** — visualization
    - Deployment: grafana
    - Service exposure: ClusterIP 3000/TCP
  - **Thanos** — long-term metrics aggregation
    - Deployment: thanos-query
    - Service exposure: ClusterIP 10902/TCP
  - **Vector** — telemetry/log pipeline
    - Deployment: vector

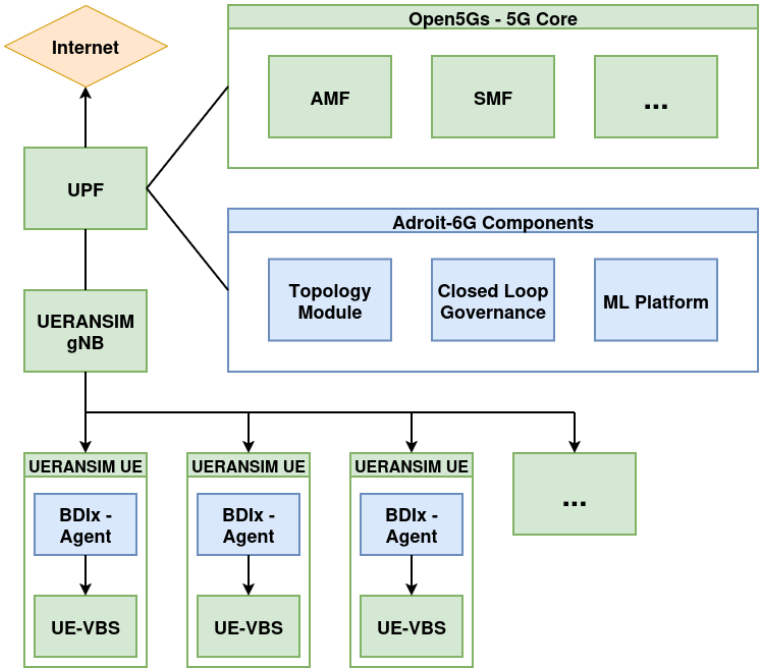

Figure 2-6: PoC1 Experiment #1 Testbed Deployment

Platform and Infrastructure:

- **Ingress Controller**:
  - Namespace: ingress-nginx
  - Deployment: nginx-ingress-controller

- Service exposure: LoadBalancer / NodePort 80, 443
- **Kubernetes System Components**:
  - Namespace: kube-system
  - coredns, metrics-server, traefik, storage provisioners

2.3.2 Experiment #2: Resources Setup Phase for Holographic Teaching session

This experiment validates the **automated orchestration and intelligent resource setup process** that enables the initiation of a real-time holographic teaching session within the ADROIT6G system. Specifically, the experiment aims to demonstrate how the ADROIT6G system: i) instantiates the holographic teaching service; ii) allocates and configures the required resources across the cloud–edge–far-edge continuum; and iii) prepares the UE-VBSs at the far-edge (by the **BDIx agents** and the **UE-VBS Components** on the Devices) to support the holographic teaching session.

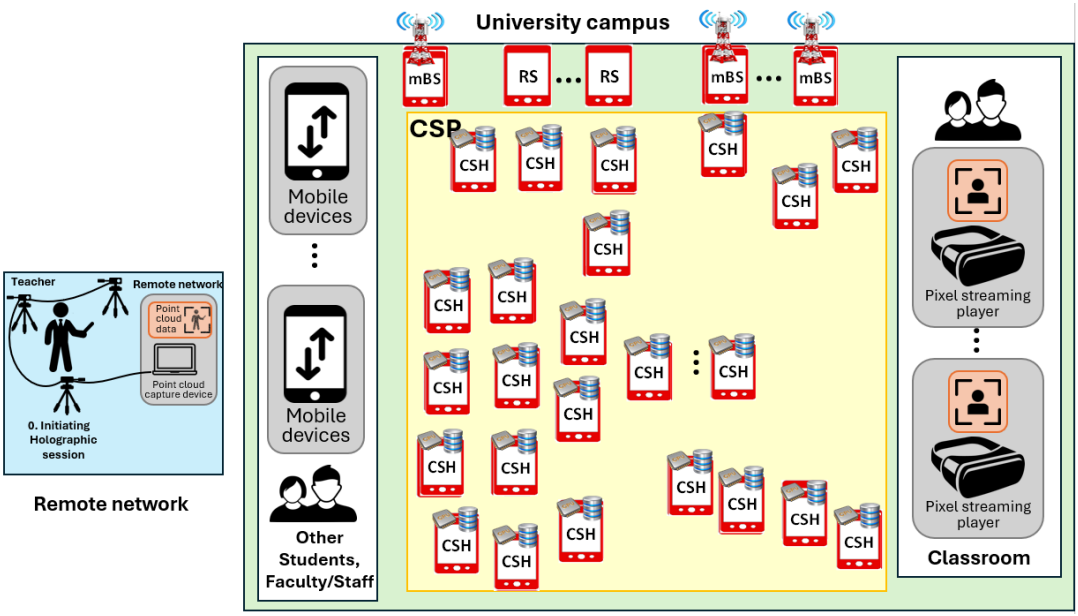

Figure 2-7: PoC1 Experiment #2 Resources Setup Phase for Holographic Teaching session

Network and computational resources at the far-edge domain dynamically adapt to the demands of immersive XR-based Holographic teaching

The process begins when **the Teacher's holographic teaching application** issues a **session start / service request** (via the Teacher frontend). This request triggers **Service Orchestration & Management** to create a service instance based on the corresponding **service descriptor**, and to coordinate the required actions with **Slice Orchestration & Management**, **UE-VBS Components, BDIx agents, Closed-Loop Governance (CLG) and CrowdSourcing AI framework**. The service descriptor will include the following data:

- maximum number of students expected to join the session,
- the frame rate and resolution of the point cloud data being sent,
- the number of individual views available to the students, and
- the IP address of the teacher's device.

Thus, once the session request is identified by the **Service Orchestrator**, an orchestration workflow is activated to allocate the necessary network and computational resources with the help of **Slice Orchestrator** and to prepare the UE-VBSs at the far-edge to support the holographic teaching session. The complete workflow sequence is described in the section 2.3.2.1.

2.3.2.1 Operationalization experiment into a concrete workflow

This section outlines the resource setup workflow required to initialize a real-time holographic teaching session. It explains, from **start trigger** to **end,** the overall storyline as a sequence of actions that occur when the **XR application running** on the teacher's device initiates the holographic teaching session. The workflow shows how the ADROIT6G system detects the start of a session, monitors computational and network conditions, and guides the preparation of UE-VBS and edge/far-edge nodes before the actual holographic content delivery starts (see Experiment #3). The sequence diagram presented in section 2.3.2.2 complement this narrative by formalizing the workflow at the interface level (i.e., specifying the concrete inter-component interactions, interfaces/APIs invoked, and key messages/payloads exchanged).

The process begins in **Step 1**, when the **XR application running** on the teacher's device initiates the holographic teaching session and sends a **session start (i.e., service request) message** to the **Service Orchestration & Management** (including session ID, the characteristics of the stream, such as the frame rate and resolution of the point cloud data, maximum number of participants, the number of supported views available to participants, and destination IP). This session start message is the event that triggers the **service/resource setup orchestration pipeline** for PoC1.

In **Step 2a**, the **Service Orchestrator** (which is under the **AI Driven M&O framework)**, based on the holographic-teaching service descriptor, and specifically the expected number of students and the characteristics of the point cloud data, determines which service functions must be activated (e.g., streaming rendering/processing, etc. functions) and which infrastructure domains will be involved (e.g., cloud, edge, far-edge).

In Step 2b, the **Service Orchestrator** triggers the **Closed-Loop Governance (CLG)** to instantiate the corresponding closed-loop instance(s) (e.g., select the relevant closed-loop descriptor/policy for the session, etc.). The **CLG** initiates the deployment **of the required AI model/policy artifact** through the AI/ML framework (i.e., **BDIx-related ML models)** required to govern agent behaviour and decision-making during the session, matching the processing demands of volumetric XR content.

In Step 3, the **Service Orchestrator** interacts with the **Slice Orchestrator** to inform the initialization of the session. Once informed, the **Slice Orchestrator** decides about the resources that should be assign to that session and a slice of isolated resources is created for the user traffic. At the end of this step, the **required resources are allocated for the session** and the UE-VBSs at the far-edge (i.e., from those UE-VBS devices already onboarded in Experiment #1) are prepared (by the **BDIx agents** and the **UE-VBS Components** on the Devices) to support the holographic teaching session.

Finally, in **Step 4**, the **Service Orchestrator** issues a **system-readiness signal**, confirming that all components are correctly configured and that the infrastructure is prepared and is in a stable state to support the subsequent **Experiment #3** (Real-time holographic teaching communication).

© ADROIT6G, 2023

Through these four steps, the workflow ensures that the required compute and network resources across the cloud–edge–far-edge continuum are allocated and configured to support the holographic teaching session.

2.3.2.2 Sequence Diagrams

The step-wise workflow described above explains the experiment at a procedural level (i.e., what happens in each step and the role of each component in the overall storyline). The **sequence diagram** presented in this section (see Figure 2-8) complements this narrative by depicting the **same workflow from an interaction perspective,** providing an implementation-oriented reference for testbed integration**.**

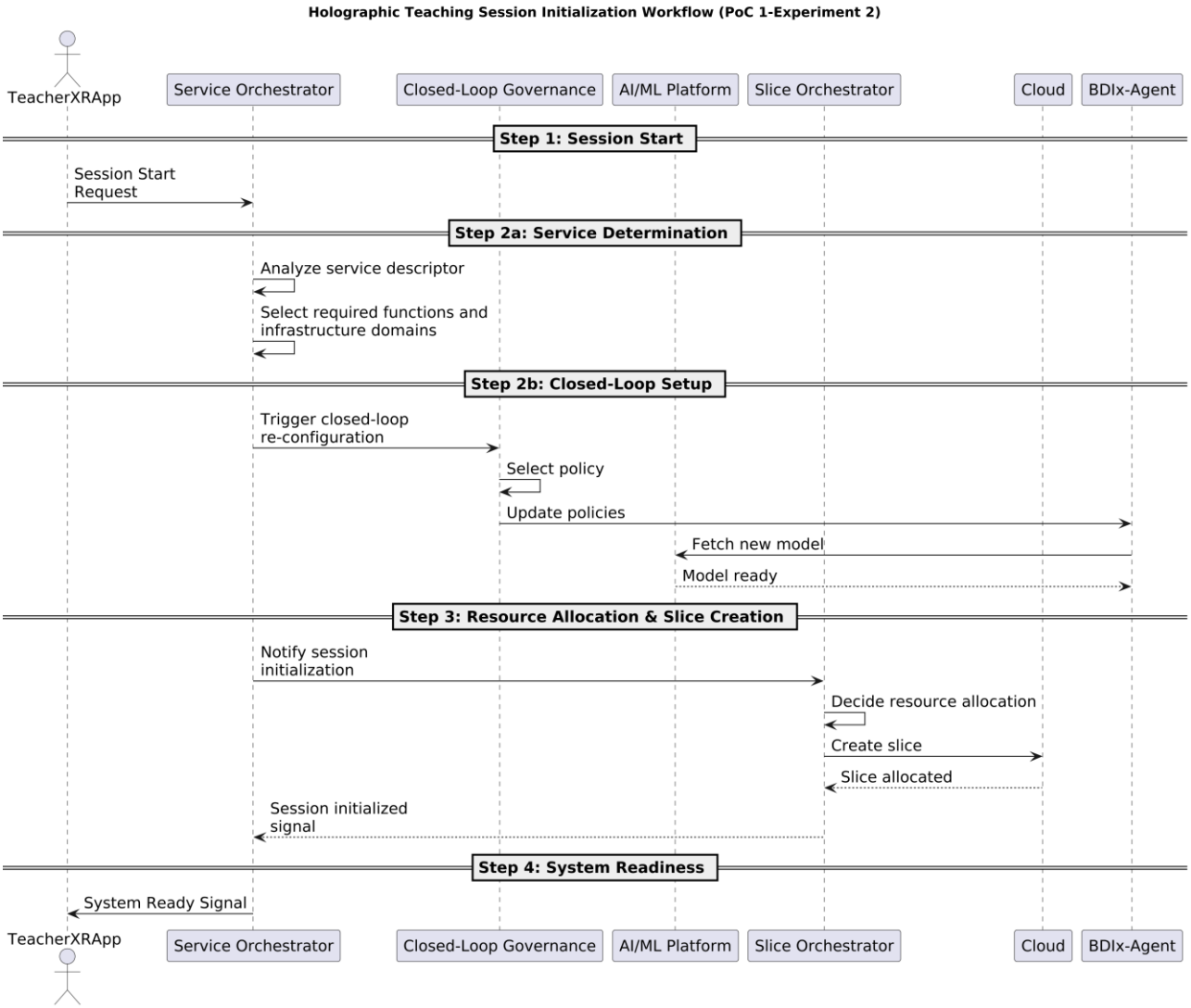

Figure 2-8: PoC1 Experiment #2 Sequence Diagram

2.3.2.3 Testbed Setup/Configuration

This section documents the **ISI/ATH testbed configuration for Experiment #2** described from two complementary perspectives: **i) network-level;** and **ii) application-level.** From the network-level

perspective, it captures the live topology with all relevant ADROIT6G components deployed and running, while from the application-level perspective it describes the Teacher's location setup required to initiate the experiment.

2.3.2.3.1 Network-Level Configuration

This section captures the live topology with all components deployed and running (namespaces, workloads, service endpoints).

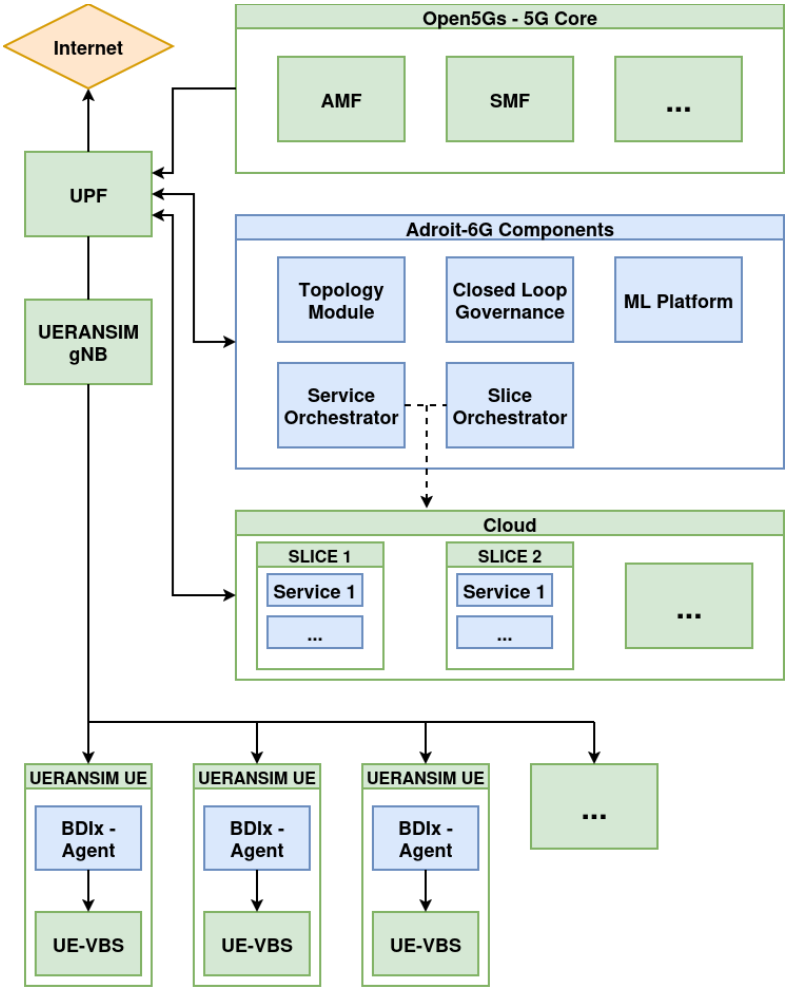

Figure 2-9: PoC1 Experiment #2 Testbed Deployment

2.3.2.3.1.1 Deployed Components

The following ADROIT6G components are deployed on top of those already available in Experiment 1. The Service Orchestrator is part of the orchestration layer deployed in Experiment 1:

- Slice Orchestrator

2.3.2.3.1.2 Namespaces and deployed workloads

The deployment is organised across Kubernetes namespaces, reflecting functional separation:

- **slice-orchestrator**: Slice Orchestrator and supporting services
- **odin:** Service orchestrator and supporting services

2.3.2.3.1.3 Workload inventory

This subsection provides a consolidated view of the deployed workloads associated with the additional components in this experiment.

Slice Orchestrator Layer (slice-orchestrator namespace):

- Slice Orchestrator
  - Deployment: slice-orchestrator
  - Service exposure: ClusterIP 8080/TCP

2.3.2.3.2 Application-Level Configuration

Teacher Location Setup: The teacher's location represents the content creation and interaction endpoint of the holographic teaching experience. The setup comprises the following hardware components. The setup can be seen in Figure 2-10.

- **Volumetric Capture Cameras**: Three Azure Kinect DK depth cameras that support RGB and infrared data arranged around the instructor to capture full-body motion and appearance.
- **Audio Capture Equipment**: Microphone or headset to capture the instructor's voice in real time.
- Volumetric Capture Computer: The audio-visual data recorded by the cameras and microphone are received by the computer of the teacher which processes the data. Through this device the teacher starts a new session and sends the service descriptor for the allocations of the resources.

Figure 2-10: PoC1 Experiment #3 - Real-time Immersive Holographic Teaching example of the teacher location setup

2.3.3 Experiment #3: Real-time Immersive Holographic Teaching Communication

The objective of this experiment is to validate the **end-to-end real-time holographic teaching experience** enabled by applications running on top of the ADROIT6G infrastructure. In this setup, the **holographic teaching application** at the teacher's location is responsible for **capturing and encoding** the instructor's 3D holographic representation using an external volumetric capture system (e.g., multiple cameras and associated processing). The experiment focuses on verifying the ability of the ADROIT6G system to **transmit the compressed volumetric data** efficiently over the high-speed, low-latency 6G infrastructure to a designated Computing and Storage Powerhouse

© ADROIT6G, 2023

(CSP, emulated in this PoC), support **real-time processing and reconstruction** of the holographic content at the far edge (within the CSP; see section 2.3.4), and **deliver the rendered holographic views** to students equipped with AR/VR devices, while meeting stringent latency, throughput, and QoE requirements.

A key aspect of this experiment is **scalability**. As the number of concurrently connected students increases, multiple **viewpoint-dependent rendering streams** need to be generated so that each student (or group of students) receives a personalised point of view that reflects their position and orientation relative to the holographic instructor. The experiment therefore assesses the capacity of the ADROIT6G system to sustain this scalability while still meeting the strict latency, bandwidth, and QoE requirements of real-time XR-based holographic teaching.

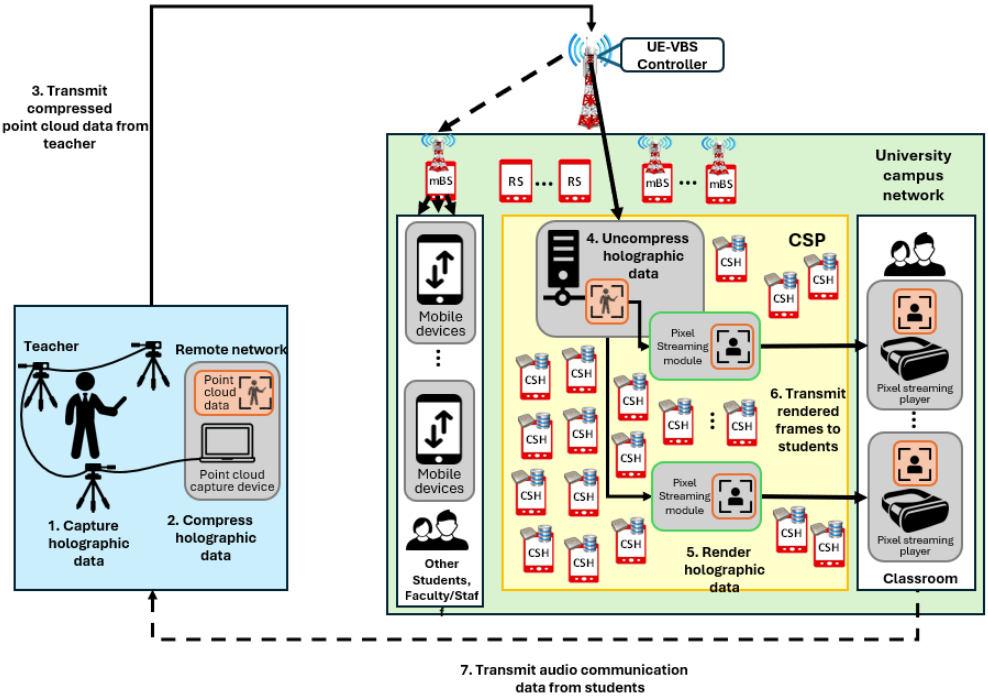

Figure 2-11: PoC1 Experiment #3 Real-time Immersive Holographic Teaching Communication

The process begins once the **Service Orchestration/Management component** (see section 2.3.2.1 Experiment #2) confirms that all components are correctly configured and that the infrastructure is in a stable state to support the subsequent holographic teaching experiment. Following this trigger, the **AI/ML Orchestrator** activates a service blueprint specifically designed for immersive holographic teaching. Based on prior orchestration results (from Experiment #2), the designated UE-VBSs already acting as CSHs forming a **Computing & Storage Powerhouse (CSP; emulated)** are used to handle the volumetric data and assist in real-time holographic rendering. The complete workflow sequence is described in section 2.3.3.1.

2.3.3.1 Operationalization of experiment into a concrete Workflow

This section describes the **workflow** that supports the immersive holographic teaching session in PoC1, showing how volumetric data and compute workloads are handled over the ADROIT6G architecture. The sequence diagram presented in section 2.3.3.2 complement this narrative by

formalizing the workflow at the interface level (i.e., specifying the concrete inter-component interactions, interfaces/APIs invoked, and key messages/payloads exchanged).

In **Step 1** the XR application establishes communication between the teacher and the students, and the teacher transmits the **compressed volumetric data** produced by the external capture system to the designated **Computing & Storage Powerhouse (CSP)** at the far edge and close to the classroom that the holographic teaching sessions is taking place. The **volumetric data** is **processed** and **integrated** within **a Unity XR environment**, where it is **encoded into real-time video streams suitable for transmission**. These streams are then delivered to connected students using the WebRTC real-time communication protocol. A **dedicated Signalling Server** coordinates the connection establishment between the CSP and student clients.

In **Step 2**, when a **student connects using a VR headset or desktop client**, the client registers with the signaling server and initiates a WebRTC negotiation process. During this process, **session descriptions (SDP offers and answers)** and **network connectivity information** (**ICE candidates**) are exchanged, allowing a direct and secure media channel to be established.

2.3.3.2 Sequence Diagrams

The step-wise workflow described above explains the experiment at a procedural level (i.e., what happens in each step and the role of each component in the overall storyline). The **sequence diagram** presented in Figure 2-12 complements this narrative by depicting the **same workflow from an interaction perspective.**

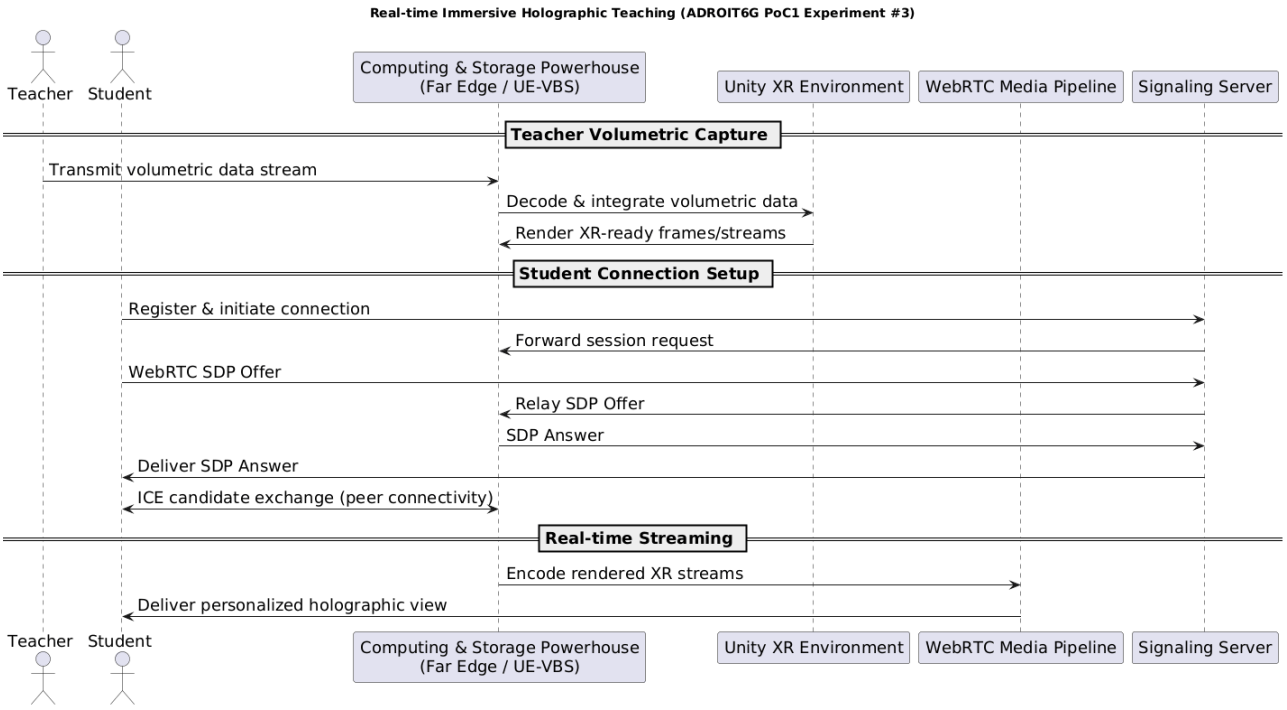

Figure 2-12: PoC1 Experiment #3 Sequence Diagram

2.3.3.3 Testbed Setup/Configuration

This section documents the concrete **ISI/ATH testbed configuration for Experiment #3**, described from two complementary perspectives: **i) network-level;** and **ii) application-level**. From the network-level perspective, it captures the live topology with all relevant ADROIT6G components deployed and running, including Kubernetes **namespaces**, **workloads**, container **image tags/versions**, exposed **service endpoints**, and the associated **monitoring/telemetry** configuration. From the application-level perspective, it describes the end-to-end setup required to execute the experiment, including the application-layer software stack and the end-user hardware deployed at both the **teacher** and **student** locations. It is worth highlighting here that during this experiment, the cloud-side rendering and processing (i.e., the CSP) will be **emulated**.

2.3.3.3.1 Network-Level Configuration

This section captures the live topology with all components deployed and running (namespaces, workloads, image tags, service endpoints, and monitoring).

The topology of the network level components remains the same and no further components have been deployed. The slices contain the necessary applications/services have been created in experiment 2 and are ready to handle the load assigned to the cloud.

2.3.3.3.2 Application-Level Configuration

At the application layer, the testbed supports an end-to-end holographic teaching pipeline consisting of the following:

- Volumetric capture and encoding at the Teacher Location
- Real-time volumetric processing and reconstruction within the Unity engine
- Encoding of holographic or multi-view video streams using WebRTC-based streaming
- Transmission of compressed volumetric data over the ADROIT6G-enabled network
- Real-time processing, reconstruction, and rendering within the CSP (emulated cloud)
- Delivery of personalized holographic views to students' XR devices

To enable end-to-end real-time holographic teaching, the following application-level software components are deployed across the teacher location, CSP, and student devices:

Teacher-Side Software

- Unity Real-Time Rendering Engine
  - Integrates volumetric capture input from RGB depth cameras
  - Performs real-time 3D reconstruction and rendering of the instructor
  - Manages virtual classroom scene and interaction logic
- Unity Render Streaming Package
  - Encodes rendered camera views and holographic content into compressed video streams (VP8 codec)
  - Streams media using WebRTC protocol
- WebRTC Streaming Module
  - Establishes peer-to-peer media connections with student clients
  - Handles video, audio, and input data transmission

CSP-Side Software (Emulated Cloud / Edge)

- Signaling Server
  - Manages connection establishment between teacher and student clients
  - Exchanges SDP offers/answers and ICE candidates
  - Maintains session coordination
- Streaming Session Manager
  - Tracks connected students
  - Assigns video streams
  - Manages multiple concurrent connections
  - Generates personalized viewpoints

Student-Side Software

- Unity XR Client Application
  - Runs on VR headsets or student computers
  - Connects to signaling server
  - Receives real-time WebRTC streams based on position in the virtual classroom
- WebRTC Client Module
  - Negotiates connection with CSP
  - Receives and decodes video streams
  - Handles audio communication
- XR Rendering Module
  - Displays holographic instructor in immersive 3D space

Teacher Location Setup: The setup for the teacher will be the same as in Experiment #2 and as described in Section 2.3.3.3.2.

Student Location Setup: The student location represents the consumption and interaction endpoint of the holographic teaching experience.

At the student location, the setup represents the **consumption and interaction endpoint** of the real-time immersive holographic teaching experience. Each student is equipped with an **AR/VR head-mounted display** capable of rendering real-time 3D holographic content, supporting head tracking and spatial rendering to ensure that the holographic instructor is visualized from a viewpoint consistent with the student's position and orientation. **Optional input devices**, such as handheld controllers or hand-tracking sensors, enable interaction within the virtual environment, while **audio equipment**—either integrated into the headset or provided through external microphones and headsets—supports natural, real-time voice communication with the instructor. Student devices connect to the ADROIT6G testbed through **wired or wireless network connectivity**, receiving personalized holographic streams generated by the system. Each student device operates as an **independent endpoint**, allowing the testbed to evaluate scalability and performance as the number of concurrently connected students increases (as illustrated in Figure 2-11).

Each student device acts as an individual endpoint, allowing the testbed to assess scalability as the number of connected students increases.

© ADROIT6G, 2023

Hardware Components:

- **AR/VR Head-Mounted Displays:** XR devices capable of displaying real-time 3D holographic content that support head tracking and spatial rendering.
- **Optional Input Devices:** Controllers or hand-tracking sensors for interaction.
- **Audio Equipment:** Built-in or external microphones and speakers/headsets.
- **Network Connectivity:** Wireless or wired connection to the ADROIT6G testbed

2.3.4 Experiment #3.1: MVLD-based Missing View Reconstruction (Local Simulation)

This experiment extends PoC1 Experiment #3 focusing on the reconstruction of missing viewpoint-dependent streams using MVLD. It targets the multi-view rendering aspect of the holographic teaching scenario, where each student receives a viewpoint-dependent stream. The objective is to assess the capability of MVLD to reconstruct a missing view from neighbouring available views, under controlled conditions, while remaining aligned with the overall PoC1 workflow.

2.3.4.1 Operationalization of experiment into a concrete Workflow

The experiment follows a simplified version of the Experiment #3 workflow, with a controlled "missing view" condition:

1. A short recording (10 sec) of the holographic teaching scenario is used as input, containing all available viewpoint-dependent streams (RGB encoding).
2. A missing-view condition is simulated by removing (or corrupting) one of the viewpoint streams at the receiver side.
3. The remaining neighbouring views are used as input to the MVLD model.
4. MVLD performs reconstruction of the missing view through cross-view generation.
5. The reconstructed view is compared against the original ground-truth view for evaluation purposes.

2.3.4.2 Simulation Configuration

The experiment is executed locally using the pretrained MVLD model already developed within the project (see D5.2 [3]). Input data consists of multi-view video recordings of the holographic teaching scenario, with all viewpoint streams available. The recordings are processed offline, and missing-view conditions are simulated programmatically. MVLD is executed as a standalone inference component on local compute resources (GPU-enabled where available), without integration into the testbed, due to the current availability of computational resources on the testbed.

2.3.4.3 Expected role and measurable effect

The role of MVLD in this experiment is the reconstruction of missing viewpoints ("fill-in" of a removed camera/view stream). The evaluation focuses on:

- Robustness to view loss (ability to reconstruct plausible missing views)
- Visual consistency with neighbouring viewpoints
- Reconstruction quality against ground truth (e.g., PSNR/SSIM/LPIPS where applicable)

© ADROIT6G, 2023

- In relation to PoC1 KPIs/KVIs defined in D5.4, this experiment provides indicative insights (not full system-level validation) on:
    - SR-HT-01 (End-to-End Latency), through the additional processing time introduced by MVLD
    - SR-HT-03 (Reliability), in terms of robustness under simulated view loss
    - NR-HT-01 (Throughput), through the potential reduction in transmitted data if one view is reconstructed instead of transmitted
    - NR-HT-02 (Latency/Jitter), indirectly through the trade-off between transmission and local reconstruction

2.4 Application of Simulators and Emulators in Testing and Validation

Due to the limitations of current mobile devices, both in terms of hardware resources and software support, as well as the significant complexity involved in implementing the Computing and Storage Powerhouse (CSP) functionality, direct realization of this component on local devices was not feasible in the present setup. The CSP is not only expected to handle computationally intensive tasks but also as data decompression and real-time rendering, that relies on the Unity engine, which cannot be efficiently supported by the mobile devices used in the testbed. In addition, a large number of mobile devices would be required to realistically realize this functionality in a fully distributed manner, which was not feasible within the current testbed setup.

For these reasons, the CSP functionality is emulated through a dedicated local server deployed at the ISI/ATH testbed. This server acts as the CSP and performs the decompression and rendering tasks on behalf of the mobile devices, thereby enabling the experimental validation of the overall system workflow while preserving the functional role that the CSP would ideally have in the target architecture.

2.5 Associated KPIs and KVIs

This section presents the KPIs and KVIs associated with PoC1, which will be used to evaluate the performance of the PoC1 experiments. For consistency across the project, the selected KPIs/KVIs are aligned with those defined in D2.3. In addition, for each KPI and KVI, this section specifies the measurement approach, including the tools, data sources, and methods that will be used on the testbed to compute and assess the reported results. The definitions and measurement methodologies follow the 6G-IA / SNS JU TMV WG KPIs and KVIs framework and the IMT-2030 guidelines, fully-aligned with the 3GPP and ETSI relevant standards for KPIs, as described analytically within D2.1 and D2.3.

Table 2-4: PoC1 Service KPIs

KPI	Description	Target Value	Measured in
SR-HT-01	**End-to-End Latency**	**< 350 ms**	**Exp. 3**
	How to measure: Measured by the XR application/service: stamp frames at capture and at render and compute Tdisplay − Tcapture.		
SR-HT-03	**Reliability**	**> 95% uptime**	**Exp. 2 and/or 3**

	How to measure: Holographic service pods (Service Orchestrator, UE-VBS/BDIx instances, CSP renderer) provide readiness/liveness probes. Use the ISI/ATH Kubernetes monitoring stack (Prometheus scraping pod readiness) to count the minutes where all holographic-service pods remain Ready; Divide that by the total observation window to report SR-HT-03 reliability (target ≥95 %).		
SR-HT-04	**System Scalability**	**Up to 50 devices**	**Exp. 3**
	How to measure: Scale the uevbs-bdix deployment up to 50 replicas within the ISI/ATH Kubernetes cluster. Prometheus exposes kube_pod_status_ready for those pods. Count the minutes where ≥50 UE-VBS/BDIx pods are Ready while the holographic service stays available.		
SR-HT-05	**AI Model Selection Latency**	**< 100 ms**	**Exp. 1**
	How to measure: Measured by AI/ML Orchestrator and CLG: trace the "select/provision model" call from request receipt to selection response.		
SR-HT-06	**Orchestration Efficiency**	**< 200 ms**	**Exp. 2**
	How to measure: Measured by Service Orchestrator		
SR-HT-07	**Service Recovery Time**	**< 1 minute**	**Exp. 2 and/or 3**
	How to measure: Delete one holographic-service pod to trigger Kubernetes' self-healing. Prometheus scrapes kube_pod_status_ready and kube_pod_container_status_restarts_total; record the timestamp when the pod transitions to Ready=0 and when the replacement pod reports Ready=1. The difference is the service recovery time.		
SR-HT-08	**Computing power needed per stream rendering (aggregate processing power in CSP)**	**~30 TFLOPS**	**Exp. 1 and/or 2**
	How to measure: Expose each rendering pod's CPU/GPU utilisation through k8s standard metrics (node exporter or vendor plugin), multiply the utilisation by the CPU/GPU nominal TFLOPS to estimate used computing power.		
SR-HT-09	**Ram needed per stream rendering**	**<32 GB**	**Exp. 1 and/or 2**
	How to measure: Use the Kubernetes metrics pipeline (Prometheus scraping container_memory_usage_bytes) to track each rendering pod's RAM consumption during stream rendering.		

Table 2-5: PoC1 Network KPIs

KPI	Description	Target Value	Measured in
NR-HT-01	**Peak Throughput**	**> 1 Gbps**	**Exp. 2 and/or 3**
	How to measure: Run an iperf3 session with the UE-VBS pod acting as client and the CSP pod as server.		
NR-HT-02	**Latency**	**< 150 ms**	**Exp. 2 and/or 3**

	How to measure: Add a simple probe pod that pings the CSP service every minute during experiment and record the reported round-trip.		
NR-HT-03	**Jitter**	**< 5 ms**	**Exp. 3**
	How to measure: From the probe pod's ping output, use the mdev value to report jitter.		
NR-HT-04	**Reliability**	**> 95%**	**Exp. 2 and/or 3**
	How to measure: From the probe pod's ping output, count packets transmitted vs. received; compute (received / transmitted) × 100 over the experiment window.		
NR-HT-05	**Orchestration Latency**	**< 250 ms**	**Exp. 2**
	How to measure: Within the ETSI NFV MANO framework, focusing on the time taken for LCM operations, such as VNF instantiation, scaling, or finalisation. Key measurements include KPIs for latency-aware slicing, utilizing real-time telemetry, 6G predictive mechanisms, and in comparison with standard-compliant 5G performance metrics.		
NR-HT-06	**Resource Utilization Efficiency**	**> 90% utilization**	**Exp 1 and 2**
	How to measure: Use Prometheus metrics (container_cpu_usage_seconds_total, container_memory_usage_bytes, NIC stats) together with the Kubernetes resource requests for each UE-VBS/BDIx and CSP pod; compute actual_usage / requested_capacity per resource.		
NR-HT-07	**Slice Adaptation Time**	**< 10 seconds**	**Exp 2**
	How to measure: Calculate the time elapsed between the Slice Orchestrator's trigger to pods reporting ready status.		

Table 2-6: PoC1 Service KVIs

KVI	Description	Values/Scaling
KVI-SR-HT-I1	**Safety**	**User-Centric perception Low / Medium / High**
	How to measure: (a) Through questionnaires and focused surveys of users and stakeholders on their perception of personal safety resulting from use of the tool. Safety can be measured with many types of metrics, such as % of stakeholders (via survey after participating in use case) they feel safe in using the tool and in the spaces where the tools are used.". (b) Additionally, at a PoC level, through the validation of the secure sharing of multi-source data and information, among domains by efficient transfer learning mechanisms performing an optimal synthesis between data-driven and model-based learning techniques.	
KVI-SR-HT-I2	**Security**	**User-Centric perception Low / Medium / High**
	How to measure: Security involves data protection and security features, user control over personal data, understanding responsibility in situations of	

	increased surveillance, building awareness of different privacy needs in different contexts and for different populations, transparency in what data is collected and how it is shared across complex ecosystems, and ensure compliance with all laws and relevant guidance. (a) Through questionnaires and focused surveys of users and stakeholders on their perception of personal safety resulting from use of the tool. % of the users expressing positive evaluation regarding the secureness of the devices and of the E2E system.	
KVI-SR-HT-I3	**Regulation**	**User-Centric perception Low / Medium / High**
	How to measure: (a) Through questionnaires and focused surveys of users and stakeholders on their perception of regulatory compliance and trust on the system and operations.	
KVI-SR-HT-I4	**Responsibility**	**0 user manipulation incidents**
	How to measure: (a) Through questionnaires and focused surveys of users and stakeholders on their perception of user manipulation resulting from use of the tool. (b) Ensuring that AI models and data handling within 6G networks are accountable, transparent, and respect user privacy.	
KVI-SR-HT-D1	**Privacy**	**User-Centric perception Low / Medium / High**
	How to measure: (a) Through questionnaires and focused surveys of users and stakeholders on their perception of security on privacy issues resulting from use of the tool. % of users responses which consider that zero privacy issues have been occurred.	
KVI-SR-HT-D2	**Fairness**	**User-Centric perception Low / Medium / High**
	How to measure: (a) Through questionnaires and focused surveys of users and stakeholders on their perception of fairness resulting from use of the tool. % of high responeses. (b) Use of Compute- and Fairness-Aware Radio Resource Allocation Algorithms, KPI linked with Control-plane efficiency (%) - % lower signaling overhead compared to network core without SCP.	
KVI-SR-HT-E1	**Sustainability**	**Mean Opinion Score (MOS) of > 4 of the users expressing positive evaluation regarding the sustainability of the devices, network, developed applications and services.**
	How to measure: Sustainability is addressed through a more resource-efficient process. The closed-loop control system enables a process that generates less waste and scrap material, contributing to improved sustainability and the use of Energy Efficiency KPIs.	

Table 2-7: PoC1 Network KVIs

KVI	Description	Values
KVI-NR-HT-I1	**Energy efficiency**	**10% improved energy efficiency compared to current status**

	How to measure: Indeed, % improved energy efficiency compared to current status (5G mainly or SoTA measurements). Energy efficiency is measured as: (a) transmitted data per unit of energy (bit/Joule) which underpins strategies for reducing power consumption and extending equipment lifespan; (b) definition of Energy Efficiency from ITU-R M.2410-0, Peak data rate is the maximum achievable data rate under ideal conditions (in bit/s), i.e., the received data bits assuming error-free conditions assignable to a single mobile station, when all assignable radio resources for the corresponding link direction are utilized (i.e., excluding radio resources that are used for physical layer synchronization, reference signals or pilots, guard bands and guard times); (c) Total energy consumption in kWh or W.	
KVI-NR-HT-D1	**Digital Inclusion**	**50 students with connected devices and 99.9% communication service reliability**
	How to measure: Digital Inclusion serves the goal of providing access to service for all users of holographic teaching, covering the 100% of participants and also is connected with the reliability of the communication service.	
KVI-NR-HT-D2	**Trustworthiness**	**100% consistent coverage of the network**
	How to measure: Coverage is defined as the cell edge distance of a single cell through link budget analysis, essentially referring to “link budget” or “cell range”, without touching upon the statistical behaviour of the radio channel or the outage probability. Coverage >99,9%, as fraction of defined service space (in 3D) within latency bound.	
KVI-NR-HT-E1	**Open collaboration**	**50 connected devices able to interact**
	How to measure: Count the number of devices which can be connected (simulation process), strongly related with the KPIs for coverage.	

3 PoC #2a – Terrestrial 6G IIoT with extreme mMTC

PoC-2a addresses a terrestrial **6G Industrial IoT (IIoT)** factory scenario in which a large number of industrial devices (sensors/actuators), mobile robots and edge nodes generate **periodic** and **event-driven** traffic (extreme **mMTC**) to support production flows (transport/assembly), safety monitoring, and flexible automation.

The goal of PoC-2a is to validate that the ADROIT6G platform can: (i) **orchestrate slices and services** end-to-end, (ii) **manage distributed edge/RAN resources** under dynamic conditions (load spikes, failures, scaling), and (iii) maintain target **E2E performance** (latency, availability, scalability, energy efficiency) while supporting higher-level applications such as robot teleoperation and predictive maintenance analytics.

The PoC validation is driven by three user stories: **Robot Operator**, **Production Manager**, and **Integrator**, capturing the needs for safe teleoperation, data-driven production insights, and streamlined integration/deployment. The outcomes of PoC2a's technology validation activities are reported in D5.6, against the specific KPIs and KVIs presented in section 3.5.

3.1 ADROIT6G Components Used in the Experiments of PoC2a

The following table lists the PoC2a components/algorithms to validate and where they participate (mapped to the two TC2 experiments and to the PoC2a).

Table 3-1: ADROIT6G Components used in the Experiments of PoC-2a

PoC Component	Exp. 1	Exp. 2
AI-Driven Management & Orchestration Framework		
Closed-loop Governance (CLG)	X	X
Service Orchestration & Management	X	X
Resource Orchestration & Management / Slice management	X	X
Slice-Monitoring & Lifecycle-Management xApp	X	X
Fully Distributed & Secure AI/ML Framework for CrowdSourcing AI		
ML model compression through Quantization	X	X
BDI - & AI-driven Unified & Open Control Operations		
BDIx Agents	X	X
UE-VBS (UE-based Virtual Base Station) control component	X	X

**** Note (for consistency with D5.3):*** *PoC2a test cases explicitly validate (1)* ***BDIx+UE-VBS****, (2)* ***closed-loop service/slice orchestration & xApp scalability****, and (3)* ***end-to-end integration under dynamic load****.*

© ADROIT6G, 2023

3.2 Baseline Scenario Description

To simulate the operation and data flow of robots and sensors the CAFA Robot app, based on Gazebo robot simulator, is used as the base application for the PoC2a experiments. The simulator allows for repeatable and comparable experiment results when conducted with different parameters – for current case – with and without using the ADROIT6G components.

3.2.1 Baseline Scenario Overview (Generic Structure)

The Gazebo simulator runs in the server of the testbed in the Docker container which acts as the Control Station (CS). It has robot models with virtual sensors (colour cameras and lidar) whose data flow and robot movement commands in the virtual 3D world of the simulator are relayed through external Raspberry Pi Virtual Robots (VRo) to create measurable network traffic. For every command and video frame a timestamp is issued in the CS. When the command has completed its round-trip between VRo and CS dashboard, the time is measured and displayed as latency in the dashboard and logged to file. The commands for the virtual robots are orchestrated by predefined scenario scripts. Running these scripts can produce repeatable results in different network and scenario configurations and combinations. The results can be analysed by comparing the latency log files from different experiment sessions and therefore it allows to evaluate the efficiency of ADROIT6G components.

3.2.2 Baseline System Actors and Components

The main components of CAFA Robot app (see Figure 3-1):

- **The Control Station (CS)** – runs in Docker container in testbed's server. The software in CS:
  - Gazebo robot simulator – 3D world with robot models with virtual sensors. It allows to test robot behaviour virtually.
  - ROS2 (Robot Operating System) - publishes and subscribes to the robot movement and sensor topics between nodes across the subnet or between specific IP addresses. Using the ROS2 in simulator allows more easily to switch to real robots which can also use ROS2.
  - ROS2 to Gazebo bridge – translates Gazebo to ROS2.
  - Web server to serve the dashboard – displays video streams from robots and has buttons to control the robots and start the scenarios. Shows current latencies of commands and sensor streams.
- **Virtual Robot relay (VRo) –** running in Docker container in Raspberry Pi. The VRo has also ROS2 installed to be able to subscribe to robot topics from CS, and relay them back to CS.

© ADROIT6G, 2023

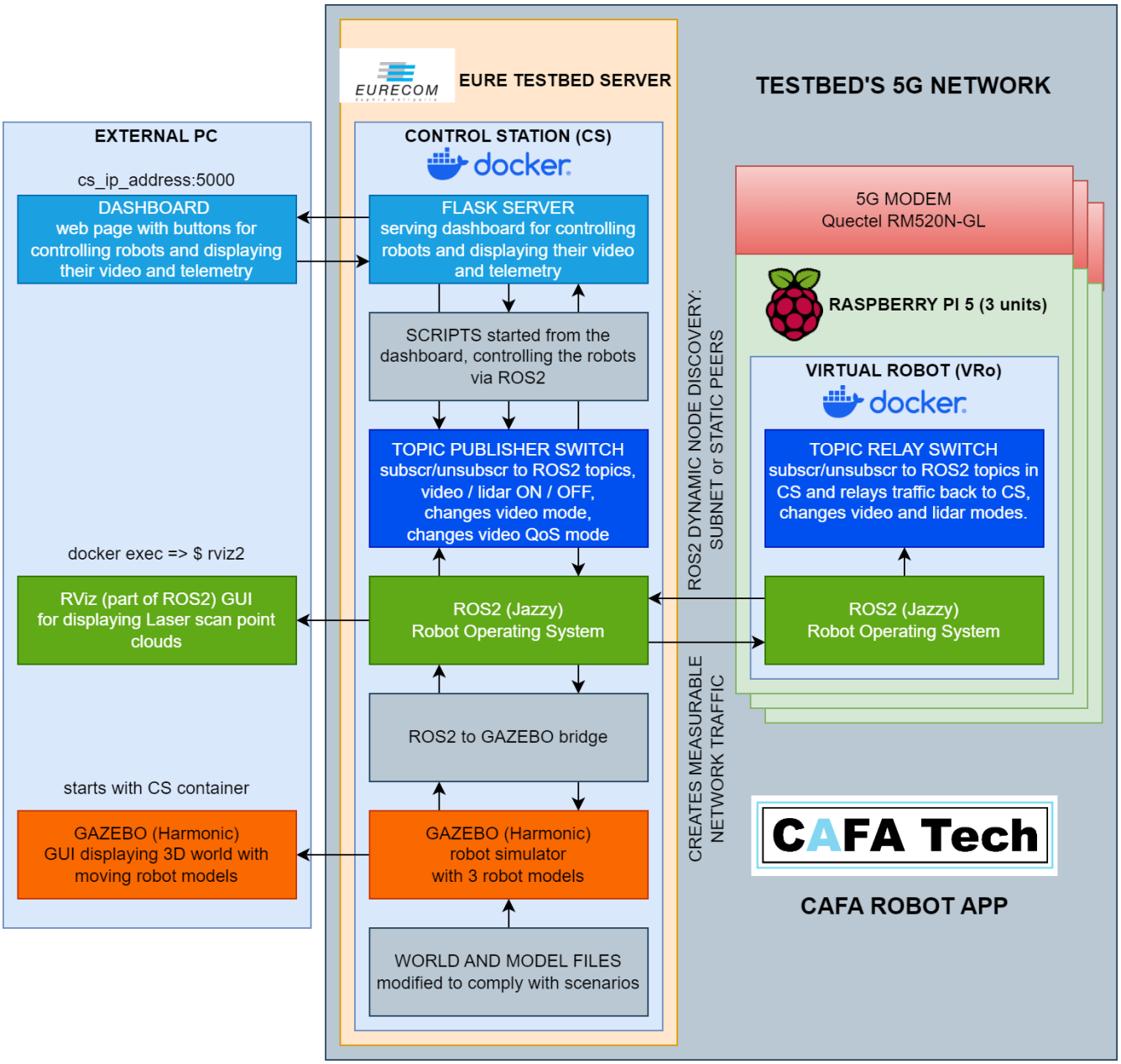

Figure 3-1: CAFA Robot app architecture showing data and process flow

3.2.3 Baseline Requirements for PoC2a

The application requirements stem from the need for seamless interaction, precise control, and efficient data processing in such a complex setting.

- **Extreme Ultra-Reliable Low-Latency Communication (URLLC):** Essential for real-time interaction and control among humans, robots, and sensors for autonomous and collaborative coordination.
- **Device-to-Device (D2D) Communication:** Primary method for information exchange between robots and sensors, provided by 6G.
- **Environmental Sensing:** The 6G network should enable the ability to sense the environment, people, and objects, determining their type, relative location, and velocity.

3.2.4 Baseline Limitations and 5G Challenges

- **Insufficient Data Rate/Throughput for Extreme Applications:** 5G's peak data rate is limited. This is explicitly stated as insufficient to support the extremely large volume of data required for future-looking applications.
- **Higher Application Latency:** actual networks typically offer application latencies of less than tens of milliseconds. For designed applications aim to achieve ultra-low latencies, targeting sub-millisecond levels is needed, crucial for real-time human-machine and machine-machine interactions and coordinated tasks.
- **Lower Reliability (Higher Packet Error Rate):** reliability with a packet error rate that is not sufficient for extreme Ultra-Reliable Low-Latency Communication (URLLC) applications.

3.3 Experiments Description

This section outlines the experiments which goal is to drive the validation efforts planned for this PoC. These experiments have been finalized and agreed upon during structured PoC workshops led by the PoC2a owners, with the co-leading of the Testbed owners and the participation of all involved component owners. Each experiment includes a clear description of its objective, its operationalization into a concrete workflow, a description of testbed set up, and the whole flow of interaction between the different components that take part in the experiment.

Table 3-2 summarizes the final set of experiments for PoC2a, whose goal is to drive the validation efforts planned for this PoC.

Table 3-2: Summary of Experiments for PoC.2a during Testing Cycle 2

Experiment	Objective	Associated with User Story (described in D5.3)
#1 Federated Predictive Analytics with CAFA **(Application + AI/ML Level)**	The objective of this experiment is to validate the use of AI-driven predictive analytics over the ADROIT6G platform, focusing on the deployment of the OULU federated learning algorithm together with the CAFA application. The experiment evaluates the ability of the testbed to ingest continuous data streams from IIoT sensors and mobile robots via the ADROIT6G network, train and update the federated model across distributed data sources, and expose predictions and maintenance recommendations through the CAFA analytics dashboard. This experiment establishes the baseline AI/ML capability for PoC2a. It demonstrates that the EURE Management and Orchestration components, the OULU federated algorithm and the CAFA backend can jointly support predictive maintenance and process optimisation, providing near real-time insights to the Production Manager without yet involving BDIx/UE-VBS components.	User Story #2: Production Manager

#2 Integrated Teleoperation and Federated Analytics with BDIx/UE-VBS (End-to-End Integration Level)	The objective of this experiment is to validate the end-to-end integration of mobile robot teleoperation, federated predictive analytics and adaptive infrastructure control in an Industry 5.0 factory cell, using all the components deployed in the PoC2a testbed. Multiple IIoT devices and mobile robots operate concurrently, generating both command-and-control (C2) traffic and sensor/telemetry streams. The experiment verifies that the EURE Closed-Loop Governance, Service Orchestration & Management, and Resource Orchestration & Management, together with the CYENS BDIx Agents and UE-VBS Components, the OULU federated algorithm and the CAFA application, can work together to support flexible and responsive manufacturing capabilities. This experiment builds on Experiment #1 and stresses the system under increasing device density and traffic load. It aims to demonstrate safe remote teleoperation of robots, robust D2D and D2CS communication enabled by BDIx and UE-VBS, and the ability to maintain key PoC2a KPIs (latency, reliability, energy efficiency, scalability) with slow degradation as the number of devices grows, while the federated analytics continues to deliver useful predictions to the Production Manager.	*User Story #1: Robot Operator* *User Story #2: Production Manager* *User Story #3: Integrator*

3.3.1 Experiment #1: Federated Predictive Analytics (Application + AI/ML Level)

This experiment validates the **Production Manager** user story: the Production Manager needs timely analytics (predictive maintenance / process optimisation) derived from continuously streamed IIoT and robot telemetry, delivered through a dashboard service. It demonstrates that ADROIT6G service/resource orchestration can deploy and sustain the analytics workflow, while federated training exchanges are made lighter via **quantized updates** to reduce uplink overhead and energy. The complete workflow sequence for this experiment is described in section 3.3.1.1.

3.3.1.1 Operationalization of experiment into a concrete Workflow

This section explains the workflow that operationalises this experiment. It explains, from **start trigger** to **end,** the overall storyline as a sequence of actions. These steps are provided below. Also, the ADROIT6G components involved in this experiment as well as the role of each component in the overall storyline is described in Table 3-3.

1. **Service request (Integrator/Operator action):** The Integrator onboards the CAFA analytics service (descriptor/blueprint) and requests deployment on the PoC2a slice(s).
2. **Service deployment:** Service Orchestration triggers Resource Orchestration to allocate edge compute + networking resources and deploy the CAFA backend, streaming ingestion endpoints, and (optionally) FL components (aggregator + clients).
3. **Data ingestion:** IIoT sensors and robots publish telemetry streams continuously; event bursts emulate alarms/rare events (extreme mMTC).

© ADROIT6G, 2023

4. **Federated training cycle:** Clients train locally on private data and send **quantized** model updates to the aggregator; the aggregator updates the global model and redistributes it.
5. **Analytics exposure:** CAFA dashboard exposes predictions, health indicators, and maintenance recommendations to the Production Manager.
6. **Closed-loop adaptation (baseline):** Monitoring collects service KPIs; CLG can trigger scale out/in or redeployment if thresholds are violated (e.g., ingestion lag, CPU saturation).

The sequence diagram presented in section 3.3.1.2 complement this narrative by formalizing the workflow at the interface level (i.e., specifying the concrete inter-component interactions, interfaces/APIs invoked, and key messages/payloads exchanged).

Table 3-3: ADROIT6G components involved in PoC2a Exp #1 and the role of each component in the overall storyline

Component / Algorithm	What is validated in PoC2a	Actions in Experiment #1
Service Orchestration & Management EURE	Automated deployment/lifecycle of PoC services (apps, analytics, xApps) on cloud-native infra; service instantiation + scaling workflows	Deploy CAFA analytics + FL components as a service; verify service instantiation and scaling
Resource Orchestration & Management / Slice management EURE	Joint resource provisioning across edge/RAN; slice creation / scaling; reaction under dynamic conditions	Allocate compute/network resources to FL clients/aggregator + CAFA backend; verify placement decisions
Closed-Loop Governance (CLG) EURE	Closed-loop policy enforcement using monitoring feedback; trigger adaptations	Monitor analytics service KPIs and trigger scale in/out; maintain SLA
Monitoring function / telemetry collection (Prometheus/Grafana stack as used in testbeds) EURE	KPI harvesting across services, edge nodes, and RAN; alerting and evidence collection	Collect KPIs for training overhead, model update cycles, and service performance
Slice-Monitoring & Lifecycle-Management xApp (near-RT RIC) EURE	O-RAN compliant slice KPI monitoring + closed-loop control actions	(Optional) Monitor slice KPI baseline supporting the analytics slice
BDIx Agents (distributed control) CYENS	Distributed resource discovery, topology awareness and decision logic for D2D/D2CS selection and resilient operation	N/A (baseline analytics experiment does not require distributed comms control)
UE-VBS (UE-based Virtual Base Station) control component CYENS	D2D relay / “far-edge” compute control; executing BDIx decisions in near real time	Optional if FL clients run inside UE-VBS containers

ML model compression via Quantization (used in federated workflows) OULU	Reduce uplink overhead/energy for distributed learning by sending quantized model updates	Apply quantization to FL model updates to reduce communication overhead and energy
CAFA applications (robot teleop + analytics dashboard)	Application-level E2E validation: robot operation + predictive insights on top of slices/services	Validate predictive analytics pipeline and dashboard consumption of streamed IIoT/robot data

3.3.1.2 Sequence Diagrams

The operational workflow of Experiment #1 is structured into three phases, shown in Figure 3-2.

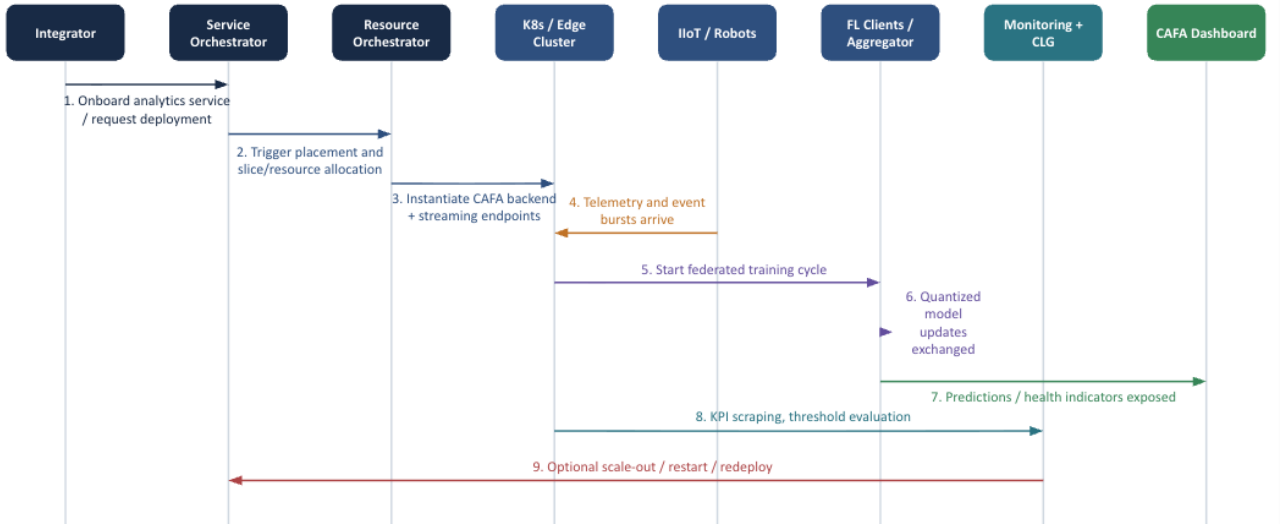

Figure 3-2: PoC2a Experiment #1 sequence diagram (provisioning, runtime, and baseline closed-loop adaptation)

(A) Provisioning phase: The Integrator requests deployment of the CAFA analytics service. The Service Orchestrator validates the service request and triggers the Resource Orchestrator to allocate the required edge/cloud compute and networking resources. Kubernetes then instantiates the required workloads, including CAFA backend components and, where enabled, the federated learning aggregator and clients. In parallel, the monitoring stack starts collecting infrastructure- and service-level metrics.

(B) Runtime phase: IIoT sensors and robot endpoints continuously generate telemetry and event-driven traffic, which is ingested by the CAFA backend. Federated learning clients train locally on their respective data partitions and send **quantized model updates** to the aggregator in order to reduce communication overhead and energy consumption. The aggregator computes the updated global model and redistributes it to the clients. The CAFA dashboard then exposes predictions, health indicators, and maintenance recommendations to the Production Manager.

(C) Baseline closed-loop phase: Prometheus/Grafana metrics are evaluated by the Closed-Loop Governance (CLG) function. When service thresholds are violated, such as ingestion lag, CPU saturation, or increased model update latency, CLG triggers corrective actions through the Service

© ADROIT6G, 2023

Orchestrator and/or Resource Orchestrator, such as scaling, restart, or redeployment. Updated KPIs are then re-harvested to verify that the analytics service returns to the expected operating region.

3.3.1.3 Testbed Setup/Configuration

This subsection documents the EURE testbed configuration used for PoC2a Experiment #1. The experiment is executed on the EURE cloud-native testbed, where ADROIT6G management and orchestration functions, the emulated RAN/Core environment, monitoring services, and the CAFA analytics application are deployed and interconnected. The purpose of this setup is to validate the deployment and runtime support of the predictive analytics workflow under terrestrial IIoT traffic conditions representative of extreme mMTC.

Deployed components

- Service Orchestrator
- Resource Orchestrator
- Closed-Loop Governance (CLG)
- AI/ML Orchestrator
- 5G Core (OAI)
- Emulated RAN (OAI RF-SIM)
- Monitoring/telemetry stack (Prometheus, Alertmanager, Grafana)
- CAFA analytics traffic patterns based on their backend and dashboard
- Federated learning components (aggregator and clients, where enabled)
- BDIx agents available in the testbed for PoC2a integration, although not actively used for distributed communications control in Experiment #1

Namespaces

- adroit-system: service-orchestrator, resource-orchestrator, clg-engine, BDIx agent(s), federated learning components
- trirematics: OAI network functions (RAN, core, UE)
- monitoring: prometheus, alertmanager, grafana

Workload inventory

- Service Orchestrator — Deployment (1/1), service endpoint exposed internally in the cluster
- Resource Orchestrator — Deployment (1/1), service endpoint exposed internally in the cluster
- CLG Engine — Deployment (1/1), service endpoint exposed internally in the cluster
- OAI core functions (AMF, SMF, UPF, AUSF, UDM, PCF, NRF) — Deployments/Services in namespace trirematics
- Emulated RAN components — Deployments/Services in namespace trirematics
- Monitoring stack — Prometheus, Alertmanager, Grafana in namespace monitoring
- FL components — aggregator and clients, deployed in namespace adroit-system when Experiment #1 includes federated training

© ADROIT6G, 2023

Networking and service exposure:

All management, orchestration, and 5G core services are exposed internally within the Kubernetes cluster. Monitoring services are also deployed internally, while Grafana may additionally be exposed externally through the testbed ingress or NodePort configuration for experiment supervision. The exact external endpoint is testbed-specific and therefore is not included here.

Setup summary:

The EURE testbed provides a Kubernetes-based service deployment environment with integrated monitoring and orchestration support. IIoT traffic is generated using periodic and event-driven traffic emulation patterns in order to reproduce extreme mMTC conditions. CAFA analytics components are producing these traffic patterns on the edge/cloud infrastructure, while federated learning clients and aggregator are distributed across testbed nodes when enabled. This setup allows validation of service instantiation, resource allocation, runtime observability, and baseline closed-loop adaptation for predictive analytics services in PoC2a.

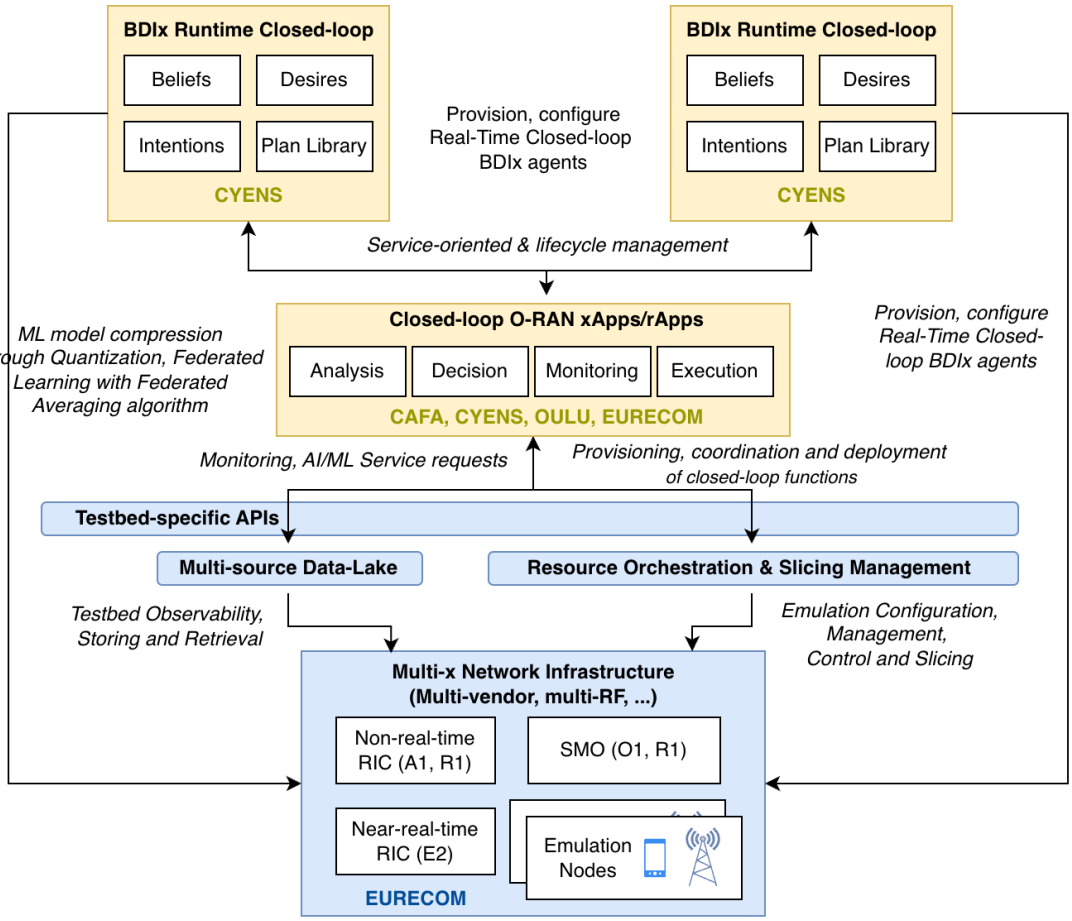

Figure 3-3: PoC2a Experiment #1 testbed setup

Federated analytics deployment in the EURE testbed: IIoT sensors and robots generate telemetry traffic through the emulated 5G/6G RAN/Core towards the Kubernetes-based edge/cloud cluster hosting CAFA ingestion, federated learning clients/aggregator, and the analytics dashboard. The management plane includes the Service Orchestrator, Resource Orchestrator, CLG, and the monitoring stack.

3.3.2 Experiment #2: Integrated Teleoperation and Federated Analytics with BDIx/UE-VBS (End-to-End Integration Level)

This experiment validates the Robot Operator, Production Manager, and Integrator user stories jointly: safe teleoperation requires low-latency reliable C2 traffic; production management requires continuous analytics; and the integrator requires rapid deployment and closed-loop governance under scaling and failures. It aligns with PoC2a test cases for (i) distributed resource discovery/management (BDIx+UE-VBS), (ii) closed-loop slice orchestration (xApp/CLG), and (iii) E2E operation under dynamic load. In addition, the experiment is designed to provide a quantitative assessment of scalability under increasing node and device density, which is particularly relevant for telecom-grade IIoT deployments involving large populations of industrial endpoints.

The complete workflow sequence for this experiment is described in section 3.3.2.1.

3.3.2.1 Operationalization of experiment into a concrete Workflow

This section explains the workflow that operationalises this experiment. It explains, from **start trigger** to **end,** the overall storyline as a sequence of actions. These steps are provided below. Also, the ADROIT6G components involved in this experiment as well as the role of each component in the overall storyline is described in Table 3-4.

1. **Deployment (Integrator):** Request deployment of (a) teleoperation service chain, (b) analytics service chain, (c) BDIx agents and UE-VBS control services, and (d) slice-monitoring xApp + closed-loop functions.
2. **Slice bring-up and policy loading:** Resource/Service Orchestrators instantiate slices; CLG loads KPI/policy thresholds for both teleoperation and analytics. Slice monitoring xApp connects to near-RT RIC and initializes service model(s).
3. **Teleoperation runtime:** Robot Operator controls one or more robots (C2 + video/telemetry). Traffic is generated concurrently with sensor streams (extreme mMTC).
4. **Distributed decisions (BDIx):** BDIx agents perform topology discovery and decide whether to use direct communication or D2D relay/UE-VBS support. Inputs include closed-loop policy files, learned models, and network resource data.
5. **UE-VBS actions:** UE-VBS enforces the selected connectivity/relay/offload path and grants radio-resource access accordingly, implementing BDIx decisions in near real time.
6. **Closed-loop slice control:** Slice monitoring xApp collects KPM/slice KPIs, pushes updated slice control policy to the RIC, and observes feedback effects (iterative loop).
7. **Stress & dynamics:** Increase device density (traffic patterns), introduce spikes/failures, and scale up to **10 emulated edge nodes with 4 UEs** while monitoring KPIs.

The sequence diagram presented in section 3.3.2.2 complement this narrative by formalizing the workflow at the interface level (i.e., specifying the concrete inter-component interactions, interfaces/APIs invoked, and key messages/payloads exchanged).

© ADROIT6G, 2023

Table 3-4: ADROIT6G components involved in PoC2a Exp #2 and the role of each component in the overall storyline

Component / Algorithm	What is validated in PoC2a	Actions in Experiment #2
Service Orchestration & Management EURE	Automated deployment/lifecycle of PoC services (apps, analytics, xApps) on cloud-native infra; service instantiation + scaling workflows	Deploy full E2E stack (teleop + analytics + BDIx/UE-VBS); enforce service chains under load
Resource Orchestration & Management / Slice management EURE	Joint resource provisioning across edge/RAN; slice creation / scaling; reaction under dynamic conditions	Dynamic resource re-allocation during load spikes/failures; validate slice instantiation time and scalability targets
Closed-Loop Governance (CLG) EURE	Closed-loop policy enforcement using monitoring feedback; trigger adaptations	Closed-loop orchestration for E2E scenario; coordinate with xApps and distributed control
Monitoring function / telemetry collection (Prometheus/Grafana stack as used in testbeds) EURE	KPI harvesting across services, edge nodes, and RAN; alerting and evidence collection	Real-time KPI collection under spikes/failures; alerting + post-analysis
Slice-Monitoring & Lifecycle-Management xApp (near-RT RIC) EURE	O-RAN compliant slice KPI monitoring + closed-loop control actions	Core validation: 5-stage xApp loop (deploy, connect, observe, control, finalize) and policy push to RIC
BDIx Agents (distributed control) CYENS	Distributed resource discovery, topology awareness and decision logic for D2D/D2CS selection and resilient operation	Core validation: BDIx internal logic + message exchange with UE-VBS and localisation; scale with >2 UEs; maintain availability under failures
UE-VBS (UE-based Virtual Base Station) control component CYENS	D2D relay / "far-edge" compute control; executing BDIx decisions in near real time	Core validation: UE-VBS - BDIx message ordering and radio-resource access control; support resilient teleoperation paths
ML model compression via Quantization (used in federated workflows) OULU	Reduce uplink overhead/energy for distributed learning by sending quantized model updates	Keep analytics running under higher load/device density while preserving energy targets
CAFA traffic patterns based on their applications (robot teleop + analytics dashboard)	Application-level E2E validation: robot operation + predictive insights on top of slices/services	Validate concurrent teleop + analytics under extreme mMTC and dynamic network conditions

© ADROIT6G, 2023

3.3.2.2 Sequence Diagrams

To remain consistent with the PoC2a validation logic introduced in D5.3, the end-to-end story of Experiment #2 is represented through three interacting sub-flows, shown in Figure 3-4.

(1) BDIx/UE-VBS distributed resource discovery and action:

A UE running BDIx boots and establishes its gRPC/data-layer configuration. It discovers neighbouring topology and available resources, evaluates policy and model inputs, and determines whether traffic should follow a direct path or a D2D relay path assisted by UE-VBS. The UE-VBS function then applies the selected relay and radio-resource configuration.

(2) Slice monitoring and closed-loop control through the xApp:

The slice-monitoring xApp is deployed and connected to the near-RT RIC. It initializes the slice service model, periodically receives KPM/KPI measurements, issues slice-control directives, and observes the effects of these actions through the feedback loop.

(3) Full end-to-end operation under dynamic load:

The complete setup combines BDIx, UE-VBS, xApps, and EURE management/orchestration functions. During runtime, UE-level and slice-level KPIs are continuously harvested, slice policies are updated when needed, and the impact of these changes is observed under traffic spikes, failures, and increasing node density. Final KPI harvesting provides the validation evidence for the end-to-end scenario.

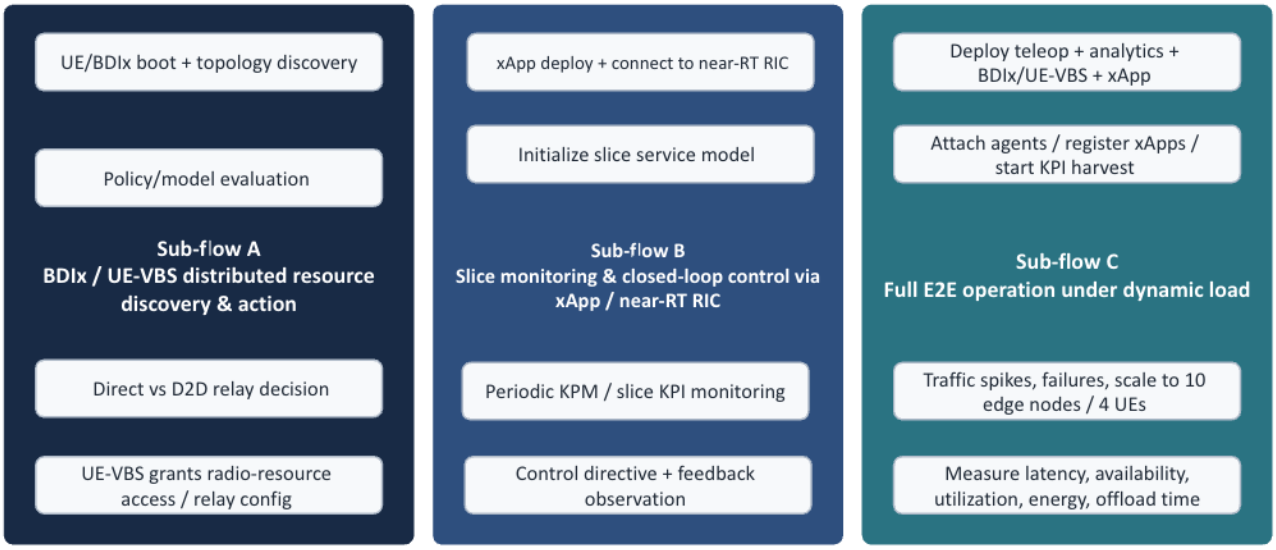

Figure 3-4: PoC2a Experiment #2 end-to-end interaction flow covering distributed resource discovery, xApp-based closed-loop control, and dynamic-load validation

3.3.2.3 Testbed Setup/Configuration

The configuration of the testbed for Experiment #2 is the same as that used for Experiment #1, with additional runtime enablement of BDIx/UE-VBS functions, xApp-based control, and dynamic load/failure scenarios.

3.4 Application of Simulators and Emulators in Testing and Validation

PoC2a relies on simulators/emulators to reach extreme mMTC conditions while keeping repeatability:

- **IoT Traffic Emulation Patterns** are applied in all PoC2a test cases to emulate the terrestrial IIoT environment (periodic + event-driven traffic and bursts).
- **Stress testing / benchmarking tools** are used to emulate node traffic and assess system response during scalability tests (notably for the closed-loop orchestration/xApp test case).
- **Dynamic workload/failure emulation** is included in the E2E integration test: traffic spikes, failures, and scaling up to 10 edge nodes and 4 UEs.
- **Real devices** are still used where needed to capture real-world KPI patterns (explicitly noted for PoC2a TC-03).
- **Gazebo robot simulation software** for repeatable robot behavior.

3.5 Associated KPIs and KVIs

This section presents the KPIs and KVIs associated with PoC2a and used to guide the validation and later performance evaluation activities. For consistency across the project, the selected KPIs/KVIs are aligned with D2.3 and with the PoC2a test-case validation logic defined in D5.3. In addition, for each KPI and KVI, the measurement strategy identifies the tools, data sources, and methods used on the EURE/CAFA testbed to collect the corresponding evidence. The quantitative performance evaluation results are reported in D5.6, while D5.4 focuses on the validation setup, observability, and measurement readiness of the PoC.

Table 3-5: Service KPIs for PoC2a

KPI	Description	Target Value	Experiment	Measurement Strategy
SR-Poc2a-01	Latency	Communication delay <10 ms	Exp. 2	Measured end-to-end using CAFA application patterns timestamps, and service/network telemetry collected through Prometheus/Grafana.
SR- Poc2a -02	Reliability	SLA 99.95%	Exp. 2	Estimated from service continuity, successful session completion ratio, and packet/service failure observations during repeated runs. Monitoring logs,

© ADROIT6G, 2023

				service restart events, and runtime alarms are used to assess whether the service remains available and stable under dynamic conditions.
SR- Poc2a -03	Scalability	Slow degradation for even higher numbers (e.g., >10 devices/m^2)	Exp. 1, Exp. 2	Evaluated by progressively increasing the number of active traffic sources, robots, UEs, and edge nodes, and observing the evolution of latency, throughput, and resource utilization. The target is not zero degradation, but bounded degradation as density increases.
SR- Poc2a -04	Energy Efficiency	Reduction of energy consumption by >30% compared to legacy	Exp. 1, Exp. 2	Assessed by comparing baseline and ADROIT6G-enabled runs, with special focus on federated learning communication cost and quantized model exchange overhead. Resource and traffic measurements are combined with OULU quantization metrics to estimate relative energy savings.

© ADROIT6G, 2023

Table 3-6: Network KPIs for PoC2a

KPI	Description	Target Value	Experiment	Measurement Strategy
NR- Poc2a - 01	Jitter	<1ms	Exp. 2	Measured from packet-level traces and time-series telemetry during teleoperation and concurrent analytics traffic. Jitter is computed on command/telemetry flows under normal and stressed runtime conditions.
NR- Poc2a - 02	Synchronized Timing accuracy	<500 microseconds	Exp. 2	Assessed through synchronized timestamp comparison across involved nodes/components, using testbed timing information and application/network traces where available.
NR- Poc2a - 03	Upload and download speeds	≥10 Mbps per device.	Exp. 1, Exp. 2	Measured through generated traffic patterns,

				application stream rates, and network monitoring counters collected during analytics and teleoperation scenarios. Device-level throughput is assessed under both nominal and loaded conditions.

Table 3-7. KVIs for PoC2a

KVI	Description	Values	Measurement Strategy
KVI-NR-Poc2a-I1	Safety	Safe operation maintained; accidents avoided	Assessed through successful execution of teleoperation scenarios without unsafe system behaviour, failed control actions, or loss of control during monitored runs. Operator observations and experiment logs provide supporting evidence.
KVI-NR-Poc2a-I2	Security	Ability of the monitoring and orchestration framework to detect anomalies, SLA violations, or abnormal behaviour	Assessed through the observability and monitoring capabilities of the testbed, combined with anomaly-related indicators when available. In D5.4,

			this is treated as validation readiness rather than a full security certification result.
KVI-NR-Poc2a-I3	Regulation	Number of standards involved	Number of standards, interfaces, or standard-aligned components applied or validated during the PoC
KVI-NR-Poc2a-D3	Digital Inclusion	Lower latency than current procedures	Communication delay < 10 ms
KVI-NR-Poc2a-D4	Trustworthiness	SLA 99.95% and latency comparison versus baseline	SLA stability and latency comparison
KVI-NR-Poc2a-E1	Sustainability	Energy-aware operation	Linked to SR-Poc2a-04
KVI-NR-Poc2a-E2	Business Value	Scalability improvement under higher density	Linked to SR-Poc2a-03
KVI-NR-Poc2a-E3	Economic Growth	Productivity increase or operational cost decrease	Before/after comparison

© ADROIT6G, 2023

4 PoC #2b – Rail Automation Support by Non-Terrestrial Networks

Rail Automation PoC Testing Cycle 2 was organized into 4 distinct experiments (Baseline, Exp.1, Exp.2, Exp.3). Contrary to Testing Cycle 1, where functional blocks have been validated, in Testing Cycle 2 tests have been structured to enhance or automate different fractions of the railway operations lifecycle. In both cases, a baseline experiment was also established to quantitatively control the improvements provided by the ADROIT6G stack.

4.1 ADROIT6G Components Used in the Experiments of PoC2b

PoC2b Testing Cycle 2 was organized into three validation experiments, complemented by one baseline experiment. Table 4-1 represents components used in each experiment.

Table 4-1: ADROIT6G Components used in the Experiments of PoC-2b

PoC Component	Baseline	Exp. 1	Exp. 2	Exp. 3
AI-Driven Management & Orchestration Framework				
Closed-loop Governance (CLG)		X		
Service Orchestration & Management		X		
Fully Distributed & Secure AI/ML Framework for CrowdSourcing AI				
AI/ML Model Selection: MERGE				
AI/ML Pipeline Manager		X		
SLA-driven Adaptive Learning Rate-based Federated Learning Policy				X
AI/ML Catalogue		X		
Predictive Monitoring and SLA Offering Solutions				
ML model compression through Quantization		X		
BDI - & AI-driven Unified & Open Control Operations				
BDIx Agents Control & Logic Re-config			X	
UE-VBS Computing Resource Control			X	
Edge & Cloud & Far Edge Domains				
Non-Terrestrial Networks	X	X	X	X

© ADROIT6G, 2023

Computing capabilities within Non-Terrestrial Networks	X	X	X	X

** Noticeably, not all components from the AI-Driven Management & Orchestration have been utilized. The models trained on the same PoC datasets served similar purposes, as a result, not all models needed to be used in the experiments. Thanks to uniform model development within the project, respective models of components can be interchanged with low to no effort. Model selection has been omitted for the same reason.*

4.2 Baseline Scenario Description

Baseline experiment is designed to not only provide a comparison for further Testing Cycle 2 experimentation, but also to demonstrate improvements made to the PoC application after receiving Testing Cycle 1 feedback. The PoC application has been improved in terms of visual interfaces and reliability upon service migration. Similarly to Test Case 1 in Testing Cycle 1 (D5.2 section 3.4.1.1), the baseline measurements remained consisting of reliability, availability and message loss KPIs. As the workflow of the experiment remained, sequence diagrams remained unchanged (see Figure 3-39 and Figure 3-40).

4.2.1 Baseline Scenario Overview

As per PoC application description (see D5.1 [2] section 3.3), the PoC application consists of the following software components:

1. **SCI-Points**[1] (in the order of thousands of devices) which are the field track devices that enable trains to be guided from one set of rails to another. Each point can either be with its status as “end position left”, “end position right”, or “no end position” when changing from one to the other.
2. **Element Controllers** (in the order of hundreds of devices) which consists of a fail-safe message broker usually on e.g., a 2-out-of-3 configuration. In this PoC, it is being deployed simply as a router or as a Pub/Sub broker, depending on the scenario being evaluated.
3. **Interlocking Services** (in the order of units of devices) which handle the interlocking logic in a replicated, parallel, and safe execution environment.

Sci-point to service communication forms the PoC2b application core, where delay of the messages is considered primary characteristic to be optimized. Service signalling, which forms the PoC traffic, is implemented in the following way (see figure below):

[1] SCI - Standard Communication Interface defined by the Eulynx industry consortium

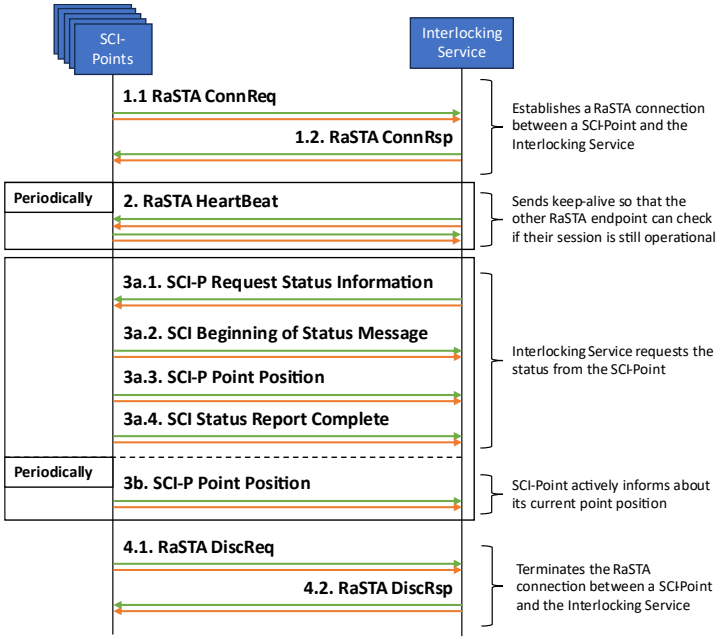

Figure 4-1: SCI-Points and Interlocking Service Signalling

SCI-points get assigned a service to communicate with. Hence, communication is established without service discovery. SCI-Points maintain a communication channel with heartbeat messages, as well as pass safety-critical status and position traffic upon a randomized frequency.

As Testing Cycle 1 already explored initial scenarios such as: *(i)* NTN as a complementary link to the existing TN links; and (ii) NTN as a primary and complementary link in remote areas, Testing Cycle 2 focuses on the remainder scenario: (iii) Interlocking Service deployment over the NTN-based computing. Specifically, the Interlocking server is offloaded to satellite asset, where on a static topology basic KPIs are measured.

4.2.2 Baseline Requirements PoC2b

The PoC requirements, identified in D2.3, which must be successfully fulfilled during the experimentation, are as follows:

- **Support of Message Priorities:** Most of the critical traffic uses messages and probably even a publish-subscribe based messaging middleware.
- **Redundancy Support:** The system must be as reliable and highly available as possible.
- **Place Service Components in Satellites:** Support computing capabilities within the satellites and the deployment of user payloads.
- **Smooth Handover Between Satellites:** Support continuous communication and data transfer with minimal interruption.
- **Edge Computing Must Consider the Geographical Location:** The service's payloads running in the Satellite Edge act differently depending on the country where it is flying over, or it may go into a power save mode over the ocean.
- **Energy Efficiency:** Prioritize energy-efficient communication.

- **Rates Around 10Mbps:** IoT applications typically create many data, so a symmetric bandwidth or even a higher upload rate is needed.
- **Round-Trip Delays < 100ms:** End-to-End delay from on-ground device to a far cloud should be under 100 milliseconds.
- **High Availability:** The network must be as reliable and highly available as possible (non-functional requirement).
- **High Security:** Security must be provided on the application layer, with zero trust principles and message integrity (non-functional requirement).

4.2.3 Baseline Limitations and 5G challenges

The following challenges, identified in D2.3 [1], are also exposed in the baseline scenario utilizing traditional 5G systems:

- Coverage and infrastructure cost is too high, especially in remote areas.
- High frequency 5G networks require frequent base stations, leading to expensive and complex setups.
- Rail automation's primary needs are reliability and safety, which 5G is limited in providing.
- 6G Non-Terrestrial Networks (NTNs) can provide wide coverage efficiently, complementing terrestrial networks and improving resilience, whereas 5G does not provide global coverage.
- NTNs offer backup or primary connectivity in underserved areas and enable edge/cloud computing for lower latency and robust operation, whereas 5G is limited to terrestrial links of the same technology, exposed to the same types of faults.
- 6G NTNs help save energy through less terrestrial infrastructure, dynamic satellite algorithms, onboard processing, and solar-powered equipment, whereas 5G does not utilize cost-free NTN.
- The hybrid 6G TN-NTN architecture ensures reliability and efficiency for safety-critical rail automation, expanding connectivity to previously inaccessible regions, which 5G as a standard does not support.

4.3 Experiments Description

Table 4-2 shows a brief description of each experiment's objectives, alongside user stories covered by the experiment. User Stories for PoC2b included: 1) Service Deployment; 2) Closed-Loop Governance and Service Update; 3) Service Termination; 4) Service Operation over NTN.

Table 4-2: Summary of Experiments for PoC.2b during Testing Cycle 2

Experiment	Objective	Associated with User Story (D5.3)
#0 Baseline	Baseline experiment targets baseline KPI measurement in traditional Railway Operations Systems. No ADROIT6G components are involved except the Edge & Far Edge emulation.	All user stories
#1 AI-Operations for Service Migration	Experiment 1 targets showcasing automation of service migration by the AI-Driven Management & Orchestration Framework. Not only automated training pipelines and automated model fetching and deployments are used, but also training process is enhanced with model compression, tailoring low-capacity NTN links where Railway application is also deployed. Moreover, an AI-closed loop is instantiated, where an already trained model allows for automated decisions of service deployment – as a result, PoC application is migrated and application KPIs are maintained even within a dynamic network with no interactions from the railway operator's side.	User Story #1 & User Story #2 & User Story #3
#2 UE connection enhancement	The objective of this experiment is to showcase UE agility in Railway operations. In this experiment, UEs are envisioned to share and relay traffic to enhance cumulative experience of all network users. Logically, passenger UEs used for on-board entertainment could leverage such functionality, however, in our experiments we envision railway elements to also utilize dynamic reconnections. In this scenario, one or several railway elements maintain a direct uplink to a satellite, allowing for all other railway elements to use NTN link as primary in case of this link providing better connection properties.	User Story #1 & User Story #3 & User Story #4
#3 Automated selective SLA monitoring	Experiment 3 is focused on railway operations, working alongside AI-workloads and logic, without conflicts or interference. Idle railway devices (with only lightweight safety-critical applications) run AI-models training to predict SLA metrics needed for operations within regulatory boundaries. As such devices are usually constrained, only available devices are used for training rounds. End to end latency for each railway element is predicted based on local element data. In the inference phase, the component automatically detects future SLA violations based on a given threshold. Such detections can be used for alerting or further automation with AI-closed loops.	User Story #1 & User Story #3 & User Story #4

4.3.1 Experiment #1: AI-Operations for Service Migration

The experiment is designed to demonstrate end to end functionality of AI-Driven Management & Orchestration, developed within the ADROIT6G project, applied to the PoC application. Specifically, a specialized AI model is trained on PoC data, and an AI-closed loop is instantiated and used for PoC application KPI enhancement by initiating service migration in an automated fashion.

© ADROIT6G, 2023

4.3.1.1 Operationalization of experiment into a concrete Workflow

The experiment revolves around AI-enabled operations, so it is organized into two stages: training phase (Figure 4-2) and closed loop operations phase (Figure 4-3). The end goal of the AI-enabled closed loop in this experiment is AI prediction of PoC application latency and predictive handover of the PoC service on mobile space assets (in case of, e.g., satellites going out of range or poor link quality).

The following components are involved in the experiment:

- **Closed-loop Governance (CLG):** allows for deployment and coordination of AI-closed loop.
- **Service Orchestration & Management:** accommodates PoC application deployment/resource management through in-testbed Resource Manager.
- **AI/ML Pipeline Manager:** facilitates AI model training and deployment, as well as long-term storage.
- **AI/ML Catalogue:** stores trained models.
- **ML model compression through Quantization:** compresses traffic of training phase to allow for FL aggregator offloading to space assets.
- **Non-Terrestrial Networks:** provides 6G service quality emulation.
- **Computing capabilities within Non-Terrestrial Networks:** provides service offloading capabilities.

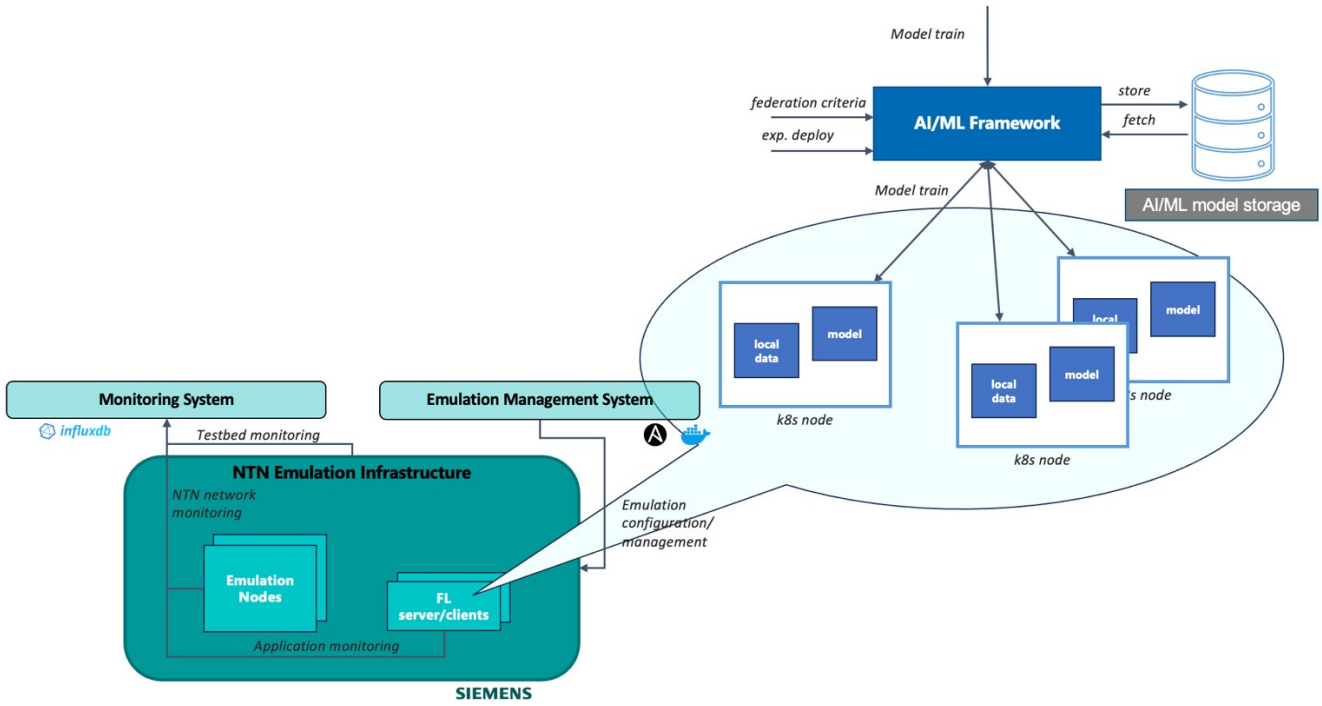

Figure 4-2: PoC2b Experiment 1: component interaction – training phase

Figure 4-2 and Figure 4-3 show how the components interact within the testbed. In short, in the training phase local federated learning clients are deployed by the AI/ML Framework besides PoC application nodes on terrestrial components. As the model training starts, clients use local PoC

application data to train each client and aggregate the global model on a satellite. Upon completion, the model is stored in AI/ML model storage component.

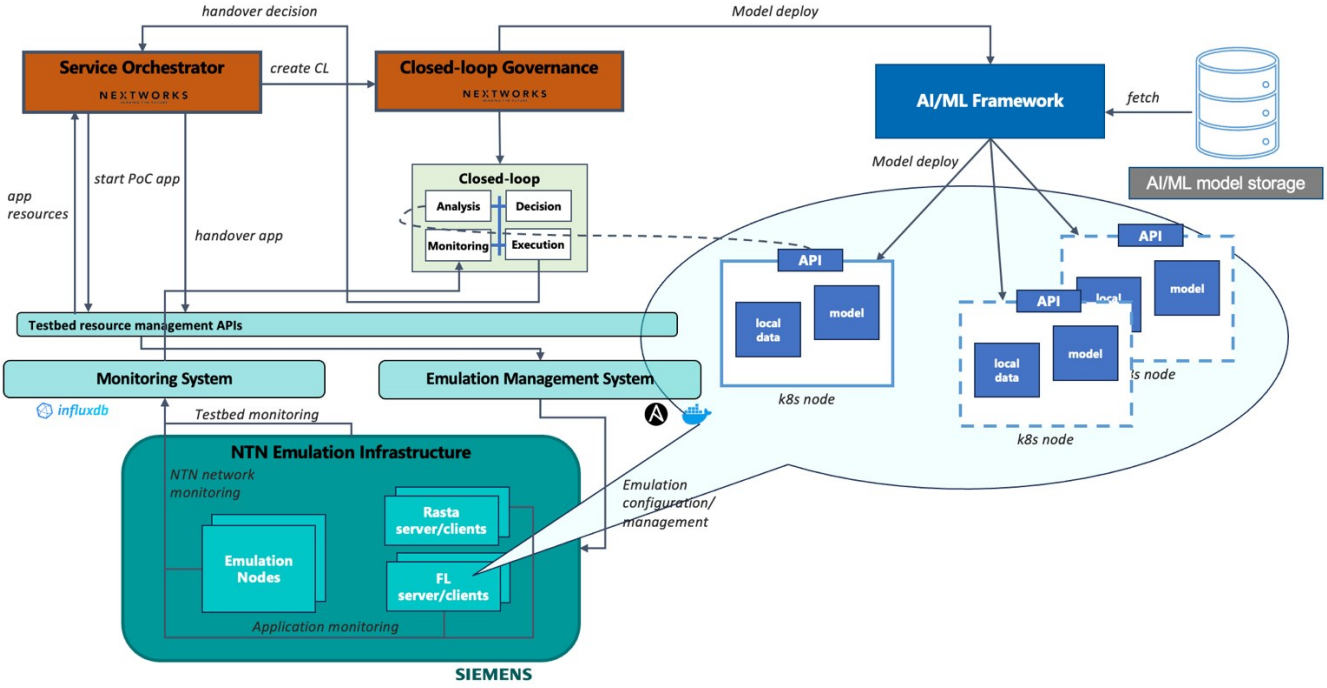

Figure 4-3: PoC2b Experiment 1: component interaction – closed loop operations phase

When it comes to inference phase, the Service Orchestrator initiates an AI-closed loop to be instantiated. Closed-Loop Governance then deploys a previously trained model on a terrestrial node via AI/ML Framework. The started closed loop then consists of 4 elements: Monitoring which fetches live PoC application data, Analysis of the incoming data by AI inference, Decision with a set threshold coming from PoC KPIs and Execution. Execution includes passing a handover decision back to Service Orchestration.

4.3.1.2 Sequence Diagrams

As mentioned above, the experiment is conducted in two phases. Phase I (training phase) has already been covered in Testing Cycle 1 - Experiment 5 (see D5.3 [2]). Phase II (closed-loop operations) is outlined in the Figure 4-4, where the following steps are taken:

1. **Service Initialization**: the user requests to start a service and provision it. The NTN testbed internal components provide applications and emulations (topologies) available for provisioning. Upon the Service Orchestrator's choice, the topology is initiated along with a service (in this case, the PoC application).
2. **Closed-loop Initialization**: Service Orchestrator initiates AI-enabled closed-loop operations. It requests a closed loop to be deployed by the CLG, which in turn deploys the model, trained in Phase I of this experiment, on the NTN testbed k8s cluster. The closed loop is then initialized, where the 4 functions (Monitoring, Analytics, Decision and Execution) cooperate with the deployed model to continuously monitor and improve PoC performance.

3. **Runtime operations**: The user then verifies the output – as satellites with PoC components deployed drift away from the ground rail track, the AI-enabled closed-loop performs PoC service handover in space to minimize PoC end-to-end latency.

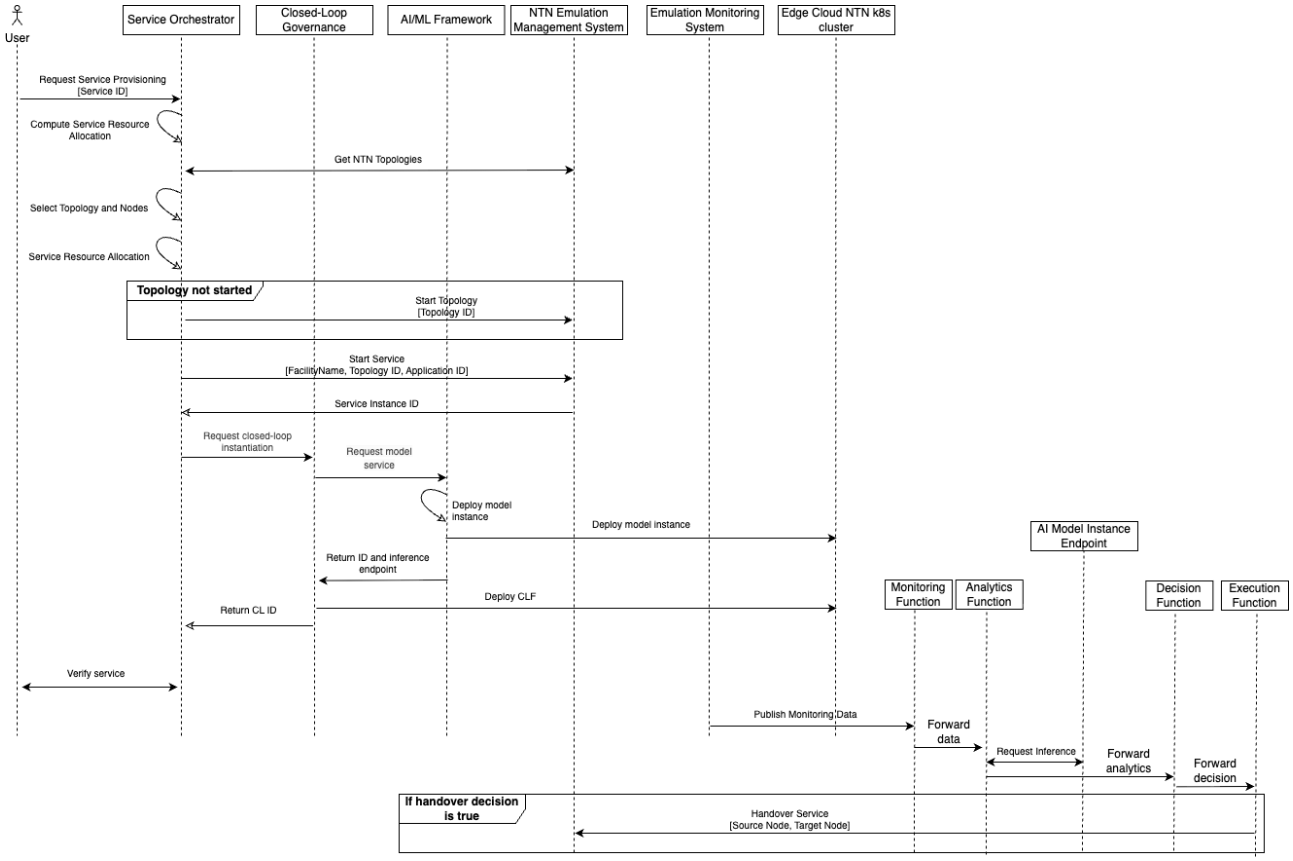

Figure 4-4: PoC2b Experiment 1 closed loop operations phase: component interaction

4.3.1.3 Testbed Setup/Configuration

The testbed setup is as described in the previous deliverable D5.2 [3]. More specifically, sections 3.3.2.1 and 3.3.2.2 describe the interfaces used for testbed interaction, whereas testbed configuration description can be found in section 2.2.2.

4.3.2 Experiment #2: UE connection enhancement

The following experiment targets UE connection enhancement in a dynamic network with mobile access points (e.g., satellites). UEs demonstrate the ability to self-organize and relay traffic for one another to maximize experience for the entire UE system. Satellites represent mobile access points, whereas different UEs maintain an uplink at different time slices. Throughout the satellite movement and uplink change the entire UE system maintains connection, as a result, maintaining KPIs of the PoC application, which utilizes links established by UEs.

4.3.2.1 Operationalization of experiment into a concrete Workflow

This experiment focuses on BDI - & AI-driven Unified & Open Control Operations. Figure 4-5 represents components and their interactions within the testbed for this test.

The following components are involved in the experiment:

- **BDIx Agents Control & Logic Re-config:** cognitive and decision-making software layer running on the device. It continuously reasons about the environment through its beliefs (e.g., channel state, neighbor topology, interference levels), sets and re-prioritizes goals through its desires (e.g., shifting from throughput maximization to energy conservation), and commits to concrete actions through its intentions and plans
- **UE-VBS Computing Resource Control:** the network function and radio capability that enables a device to act as a mobile base station, extending 6G coverage, relaying traffic, and contributing computing resources to the far-edge infrastructure
- **Non-Terrestrial Networks:** provides 6G service quality emulation, along with link creation and destruction
- **Computing capabilities within Non-Terrestrial Networks:** provides UE device emulation

Figure 4-5 showcases the interconnections between the components within the experiment. BDIx agents, deployed on the testbed as emulated UEs, retrieve LOS/NLOS information from Testbed monitoring systems. Based on this information, some UEs may decide to become UE-VBS, and a tree of connections shall be formed between the devices. As the topology is formed, the emulation management system reads the resulting tree from the topology module and updates the emulated links on the data plane in real time. As a result, BDIx agents' decisions take immediate effect on emulation links, thus enabling PoC application clients running alongside BDIx agents to get the fastest connection to the PoC application server, deployed on a satellite.

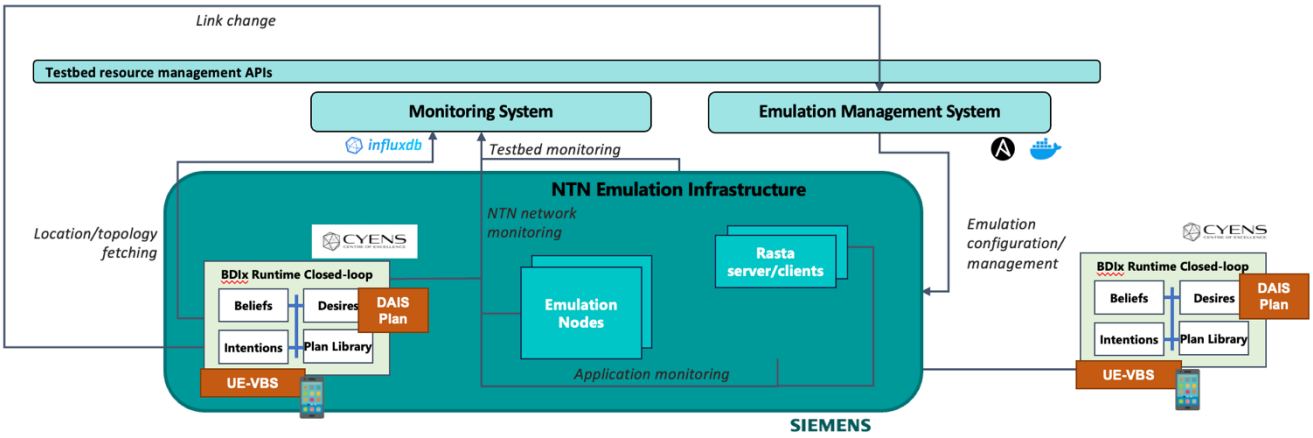

Figure 4-5: PoC2b Experiment 2: component interaction

4.3.2.2 Sequence Diagrams

Figure 4-6 outlines the stepwise flow of the experiment, in a sequence diagram form. The experiment consists of the following steps:

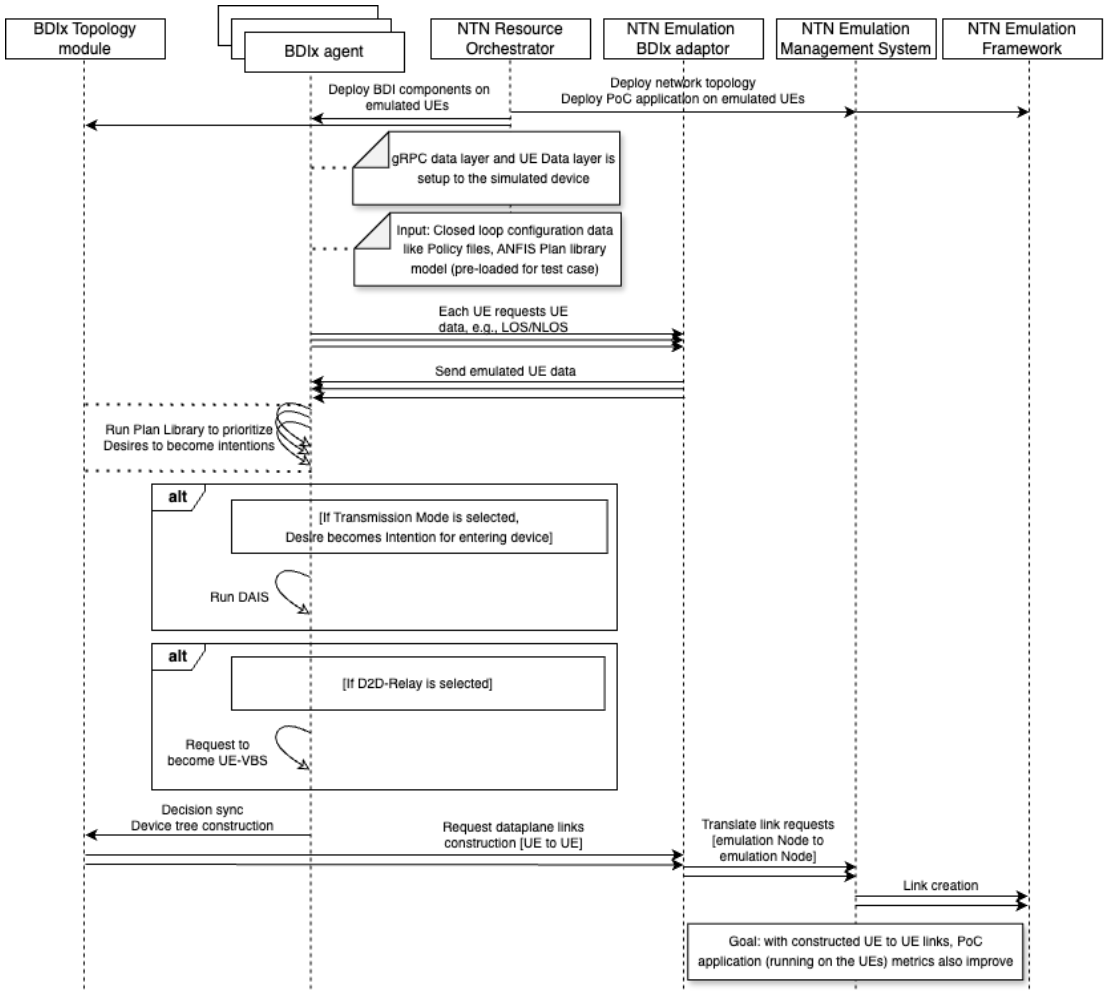

Figure 4-6: PoC2b Experiment 2: sequence diagram

1. **Setup**: NTN testbed internal components initialize a TN-NTN network session with emulated UEs. Each UE has both PoC client running, and BDIx agent components, namely, BDIx Agents Control & Logic Re-config and UE-VBS Computing Resource Control. The UEs connect to Topology module via terrestrial management plane links(high-latency). Satellite constellation is constructed with one mobile PoC server, with exactly one uplink from one of UEs (expected to become UE-VBS).
2. **Belief partial emulation**: as UEs are emulated, each UE inquires with NTN Emulation adaptor on their LOS/NLOS properties and geographical location to report their Beliefs to the Topology module.
3. **Topology tree construction:** as the experiment runs, UEs exchange management data with the Topology module, including current connectivity information as well as Desires. Topology module continuously runs Plan Library as a result constructing a hierarchical tree of UEs.
4. **Emulation feedback loop:** upon any tree update, NTN Emulation adaptor initiates link construction or destruction in order for emulated UEs to reflect the generated hierarchy.
5. **Full tree formation:** as all UEs successfully fulfil their Desires, a complete tree is formed, with some UEs as leaves, some as relay devices and some as UE-VBS (at least one device with

uplink). Consequently, PoC latency improves drastically, as dataplane UE to UE links are designed high speed.

4.3.2.3 Testbed Setup/Configuration

The testbed setup is as described in the previous deliverable D5.2 [3]. More specifically, section 3.3.2.4 describes the interface used for UE to testbed communication and the configuration of the testbed is described in section 2.2.2.

4.3.3 Experiment #3: Automated selective SLA monitoring

The last experiment, namely Experiment #3, focuses on native training and inference of workloads on railway operations hardware. It showcases how low-power devices can be utilized for both high priority applications (PoC application RaSTA) and low priority AI training/inference. In this context, the model is compact and anonymous to other training clients, meaning Federated learning is utilized.

4.3.3.1 Operationalization of experiment into a concrete Workflow

Experiment #3 focuses on native AI deployment, as a result, both stages of deployment and inference can be described with the same deployment schema (see Figure 4-7). This native approach is realistic in the context of industrial hardware, as it is not always exposed to third party components, e.g., model storage or centralized AI/ML pipeline manager. Moreover, private nature of Federated Learning used allows for this experiment to be even more appealing for real industrial deployments.

The following components are involved in the experiment:

- **SLA-driven Adaptive Learning Rate-based Federated Learning Policy:** allows for in-operations training without throttling the system, reports SLA violations for the system to take action
- **Non-Terrestrial Networks:** provides 6G service quality emulation
- **Computing capabilities within Non-Terrestrial Networks:** provides client emulation, allows for FL client deployment, training and inference

Figure 4-7 describes the general information flow between ADROIT6G components in this experiment. Federated Learning clients (of any project FL framework/model) are deployed alongside PoC application clients (RaSTA points). Local client data is then used for both training and inference, whereas model storage is local to the testbed as well as model deployment is performed natively. Training is initiated within a fraction of clients to not throttle the entire network with model traffic, where client selection is performed using SLA-driven Adaptive Learning Rate-based Federated Learning Policy. Inference is performed afterwards to detect SLA violations, which are then reported by the server back to the Emulation Management System of the testbed to perform future rebalancing (e.g., change of network slices or shutdown of low priority workloads).

© ADROIT6G, 2023

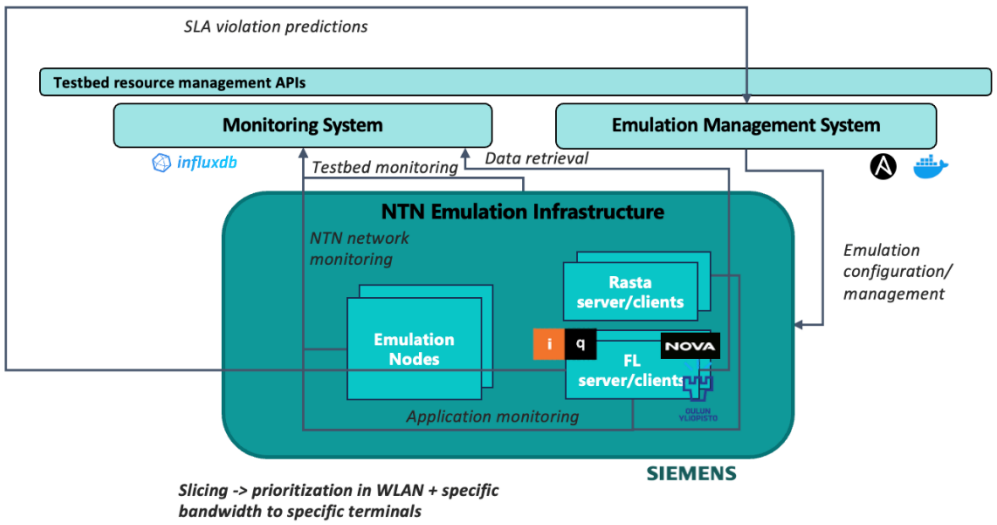

Figure 4-7: PoC2b Experiment 3: component interaction

4.3.3.2 Sequence Diagrams

Figure 4-8 describes the necessary steps to recreate experiment 3. The experiment consists of the following steps:

1. **Setup**: NTN testbed internal components initialize a TN-NTN network session with emulated Federated Learning clients and server. PoC server is deployed alongside the Federated Learning server (on a satellite node), PoC application clients and Federated Learning clients are co-located as well. Automated deployment of the model is not a part of this experiment, for that process please refer to Experiment #1.
2. **Model training phase**: The training is requested from the NTN Resource Orchestration upon constellation stabilization. Training is then performed in rounds, where clients fetch real-time local data from respective co-located PoC clients. Upon each round, clients exchange weight updates with the server according to the proposed algorithm. SLA of e2e PoC application latency is the training target.
3. **Transition phase**: As the training is finalized (upon reaching maximum rounds or target accuracy), the model is saved on the server and then deployed to the test client (used for inference).
4. **Model inference phase:** Eventually, inference on the test clients is called by the NTN Resource Orchestrator. The resulting SLAs are fetched periodically from each client and pulled to the NTN Emulation Monitoring System. Upon predicted SLA violation, not only can the NTN resource orchestrator manipulate network slices to prevent the violation, but also the operator of the system can react accordingly thanks to dynamic SLA prediction displaying per client.

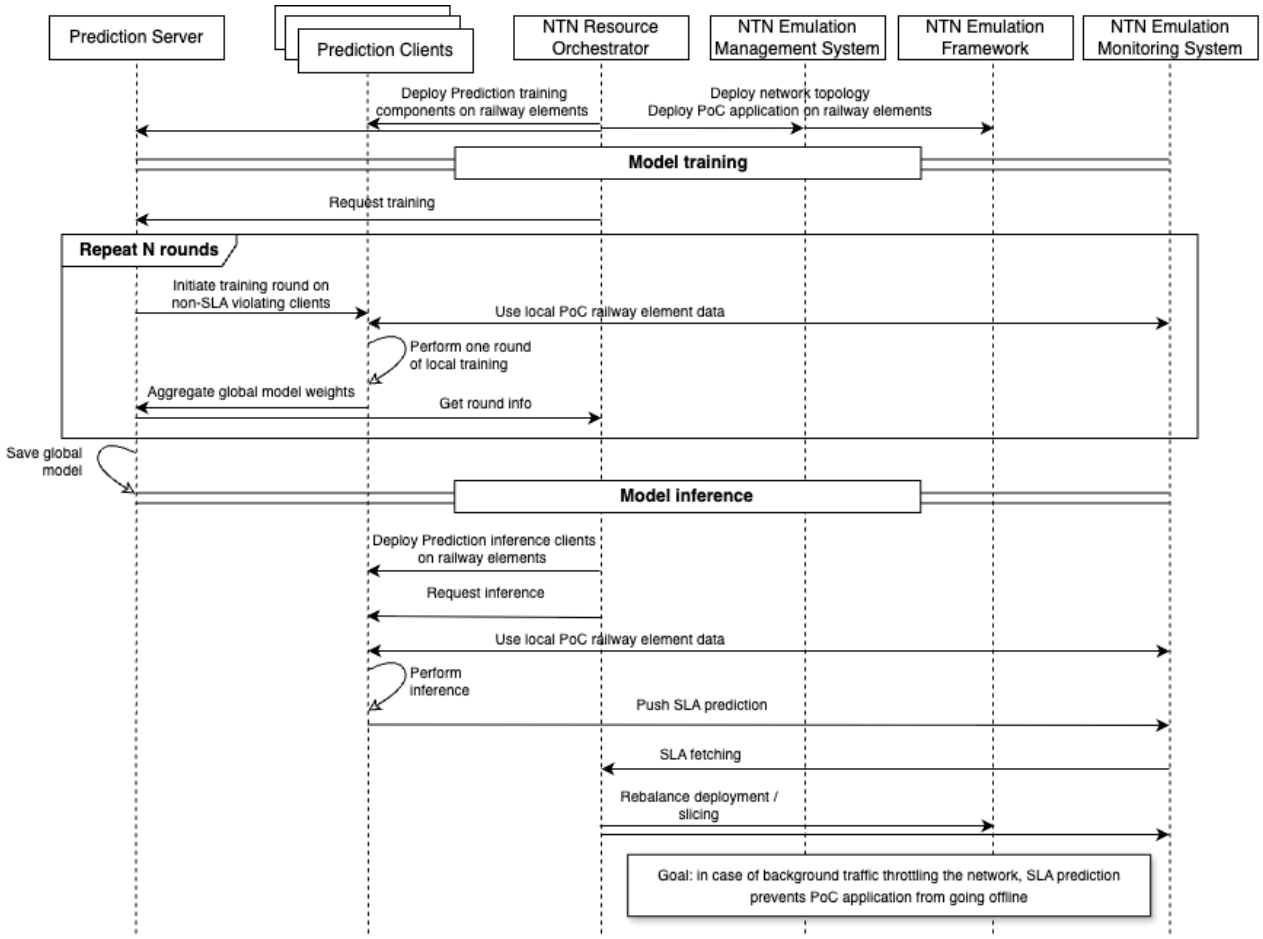

Figure 4-8: PoC2b Experiment 3: sequence diagram

4.3.3.3 Testbed Setup/Configuration

The testbed setup is as described in the previous deliverable D5.2 [3]. More specifically, section 3.3.2.3 outlines the influxDB interface used in the experiment, and section 2.2.2 iterates through the testbed configuration.

4.4 Application of Simulators and Emulators in Testing and Validation

All of the aforementioned experiments leveraged Satellite Communication and Computing Framework Emulation to emulate TN-NTN integrated network with 5G/envisioned 6G properties. Moreover, UEs for experiment #3 were represented with docker containers, to allow for smooth testbed and Satellite Communication and Computing Framework integration.

To successfully train two models used in Testing Cycle 2, two datasets have been generated, respectively. Each dataset is a logical extension of the previously generated dataset (see D5.3 [2] section 4.3.1 and 4.3.2).

4.5 Associated KPIs and KVIs

Below are the KPIs and KVIs, associated with the PoC2b, as initially claimed in D2.3, as well as respective means of measurement and control of their achievement.

Table 4-3: Service KPIs for PoC2b

KPI	Description	Target Value	Experiment	Measurement strategy
SR-NTN-01	Service availability	>99.999% over longer time frames	All	Continuous monitoring through PoC dashboard
SR-NTN-02	Reliability	Message loss on application layer $<10^{-5}$	All	Continuous monitoring through PoC dashboard
SR-NTN-03	Scalability	Slow degradation for even higher numbers (e.g., >10 devices/m^2)	Experiment #2 & #3	Setup of emulation with specified density
SR-NTN-04	Energy Efficiency	Reduction of energy consumption by >50% compared to legacy NTNs	Experiment #1	Theoretical assessment of network efficiency combined with Quantization savings

Table 4-4: Network KPIs for PoC2b

KPI	Description	Target Value	Experiment	Measurement strategy
NR-NTN-01	Latency / Jitter	<100 milliseconds / <50 milliseconds	All	Continuous monitoring through PoC dashboard
NR-NTN-02	Device density	<10 per m^2	Experiment #2 & #3	Setup of emulation with specified density
NR-NTN-03	Packet loss rate	$5 \cdot 10^{-6}$	All	Continuous monitoring through PoC dashboard
NR-NTN-04	QoE	>4.3	All	Relative assessment after experiment runs

The following KVIs would be measured through consortium-wide questionnaires in order to be assessed, after all experimentation results are disseminated within the consortium.

Table 4-5: Service KVIs for PoC2b

KVI	Description	Values	How to measure
KVI-NR-Poc2b-I1	Safety	Monitoring resources from different providers to identify anomalies and intrusion detection. PoC of anomaly	# of accidents prevented or the ability of the predictive monitoring and orchestration framework to detect anolmalies

© ADROIT6G, 2023

		detection in production-grade systems, dealing with the orchestration policies of resources across multiple domains, while maintaining service quality guarantees. Target value > 0.85 The KVI is related to KPIs on Anomaly detection recall and sensitivity	across resources, cyber-intrusions, SLA violations etc. KVI is related with KPIs, such as Recall, accuracy, False negative rate
KVI-NR-Poc2b-I3	Regulation	# of standards involved in the PoC Execution	# of standards applied, implemented or validated during the PoC.
KVI-NR-Poc2b-D3	Digital Inclusion	Lower latency than current procedures Communication delay <10 ms	
KVI-NR-Poc2b-D4	Trustworthiness	> 99.999% over longer time frames	% of the users expressing positive evaluation regarding the robustness of the devices, the trustworthiness of the used AI algorithms, the E2E privacy of the system.
KVI-NR-Poc2b-E1	Sustainability	SR- Poc2b -04 Reduction of energy consumption by >50% compared to legacy NTNs	Linked with the KPI of Energy Efficiency and Reduction
KVI-NR-Poc2b-E2	Business Value	Slow degradation for even higher numbers (e.g., >10 devices/m^2) and <10 per m^2	Use of the relevant KPI, as well for # of devices per m2
KVI-NR-Poc2b-E3	Economic Growth	% decrease in operational costs (target >10%)	Before-after comparison of the time required for service.

© ADROIT6G, 2023

5 PoC #3 – Collaborative Construction Robots

PoC-3 focuses on using 6G communication technology to automate modern construction sites by using mobile wheeled robots and aerial drones. A key innovation is the autonomous coordination between these robots and drones, enabled by 6G technology for precise communication and positioning. This coordination is crucial for synchronizing tasks like lifting, loading, and navigating, ensuring safety and efficiency.

The PoC 3 application involves autonomous construction robots and drones coordinating in a complex outdoor environment, requiring ultra-reliable low-latency communication, precise positioning, and real-time sensor data exchange. The system leverages AI-driven control frameworks and distributed edge computing to enable seamless, secure, and energy-efficient collaboration among multiple robotic agents, supporting high-precision tasks like lifting, moving, and assembling construction materials.

5.1 ADROIT6G Components Used in the Experiments of PoC3

The following table lists the PoC3 components/algorithms to validate and maps them to experiments.

Table 5-1: ADROIT6G Components used in the Experiments of PoC-3

PoC Component	Exp. 1	Exp. 2
AI-Driven Management & Orchestration Framework		
Closed-loop Governance (CLG)		X
Service Orchestration & Resource Management	X	
Fully Distributed & Secure AI/ML Framework for CrowdSourcing AI		
AI/ML Pipeline Manager	X	X
AI/ML Catalogue	X	X
BDI - & AI-driven Unified & Open Control Operations		
BDIx Agents		X
UE-VBS (UE-based Virtual Base Station)	X	X
Application Level		
CAFA Robot Application	X	X

© ADROIT6G, 2023

5.2 Baseline Scenario

PoC3 addresses the complex and dynamic environment of robotic construction, where extreme Ultra-Reliable Low-Latency Communication (URLLC) is essential for seamless interaction between humans, ground robots, and drones. In this scenario, multiple robots and drones operate and collaborate autonomously on a construction site. They exchange information primarily using 6G-provided Device-to-Device (D2D) communication. Communication between different entities (drones, mobile robots, static cameras) is managed across distinct network slices, tailored to their specific requirements.

PoC3 aims to demonstrate how 6G can transform individual robots into "cobots" that form symbiotic relationships to efficiently fulfill complex tasks and better cater to human needs. This fosters trustworthiness and digital inclusion, enabling new business models for verticals in robotic construction.

5.2.1 Baseline Scenario Overview

To simulate the operation and data flow of robots and sensors the CAFA Robot app based on Gazebo robot simulator is used as the base application for the PoC3 experiments. The simulator allows for repeatable and comparable experiment results when conducted with different parameters – for current case – with and without using the ADROIT6G components.

The Gazebo simulator runs in the server of the testbed in the Docker container which acts as the Control Station (CS). It has robot models with virtual sensors (colour cameras and lidar) whose data flow and robot movement commands in the virtual 3D world of the simulator are relayed through external Raspberry Pi Virtual Robots (VRo) to create measurable network traffic. For every command and video frame a timestamp is issued in the CS. When the command has completed its round-trip between VRo and CS dashboard, the time is measured and displayed as latency in the dashboard and logged to file. The commands for the virtual robots are orchestrated by predefined scenario scripts. Running these scripts can produce repeatable results in different network and scenario configurations and combinations. The results can be analysed by comparing the latency log files from different experiment sessions and therefore it allows to evaluate the efficiency of ADROIT6G components.

5.2.2 Baseline System Actors and Components

The main components of CAFA Robot app (this is the same as in PoC2a; see Figure 3-1):

- **The Control Station (CS)** – runs in Docker container in testbed's server. The software in CS:
  - Gazebo robot simulator – 3D world with robot models with virtual sensors. It allows to test robot behaviour virtually.
  - ROS2 (Robot Operating System) - publishes and subscribes to the robot movement and sensor topics between nodes across the subnet or between specific IP addresses. Using the ROS2 in simulator allows more easily to switch to real robots which can also use ROS2.
  - ROS2 to Gazebo bridge – translates Gazebo to ROS2.

© ADROIT6G, 2023

- Web server to serve the dashboard – displays video streams from robots and has buttons to control the robots and start the scenarios. Shows current latencies of commands and sensor streams.

- **Virtual Robot relay (VRo) –** running in Docker container in Raspberry Pi. The VRo has also ROS2 installed to be able to subscribe to robot topics from CS, and relay them back to CS.

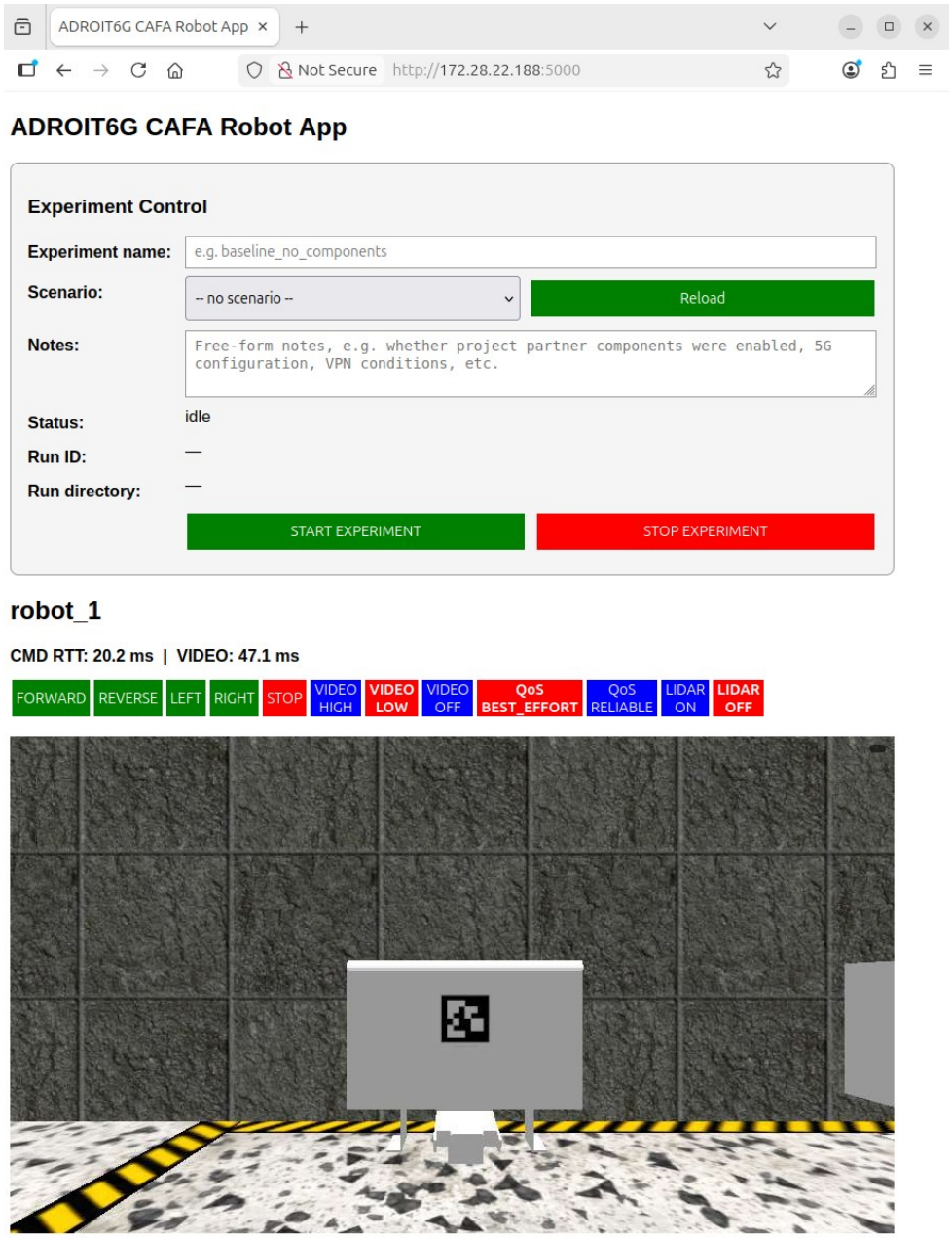

Figure 5-1: CAFA Robot app - the dashboard.

In the picture you can see a view from the camera of the virtual robot_1 (tugbot_1 model in the Gazebo). You can also see a control block for running experiment scenarios, and current values for command and video latency.

5.2.3 Baseline requirements for PoC3

The application requirements stem from the need for seamless interaction, precise control, and efficient data processing in such a complex setting.

- **Extreme Ultra-Reliable Low-Latency Communication (URLLC):** Essential for real-time interaction and control among humans, ground robots, and drones for autonomous and collaborative coordination.
- **Device-to-Device (D2D) Communication:** Primary method for information exchange between robots and drones, provided by 6G.
- **Cloud-Native Sensor Data Processing, Command and Control:** Cloud-native applications are needed to process the extensive sensor data flows and for managing command and control functions.
- **Environmental Sensing:** The 6G network should enable the ability to sense the environment, people, and objects, determining their type, shape, relative location, velocity, and potentially material properties.

5.2.4 Limitations and 5G challenges

- **Insufficient Data Rate/Throughput for Extreme Applications:** 5G's peak data rate is limited. This is explicitly stated as insufficient to support the extremely large volume of data required for future-looking applications.
- **Higher Application Latency:** actual networks typically offer application latencies of less than tens of milliseconds. For designed applications aim to achieve ultra-low latencies, targeting sub-millisecond levels (collaborative robots in PoC3) is needed, crucial for real-time human-machine and machine-machine interactions and coordinated tasks.
- **Lower Reliability (Higher Packet Error Rate):** reliability with a packet error rate that is not sufficient for extreme Ultra-Reliable Low-Latency Communication (URLLC) applications.

5.3 Experiments Description

This section outlines the experiments which goal is to drive the validation efforts planned for this PoC. These experiments have been finalized and agreed upon during structured PoC workshops led by the PoC3 owners, with the co-leading of the Testbed owners and the participation of all involved component owners. Each experiment includes a clear description of its objective, its operationalization into a concrete workflow, a description of testbed set up, and the whole flow of interaction between the different components that take part in the experiment.

Table 5-2: Summary of Experiments for PoC.3 during Testing Cycle 2

Experiment	Objective	Associated with User Story (described in D5.3)
#1 Autonomous Transport with Dynamic Re-Routing	Demonstrate that AGVs and a UAV can autonomously navigate and transport material, maintain localization accuracy, and be dynamically re-routed under high network load, using the full stack of 6G-ready components.	User Story #1 User Story #2 User Story #3
#2 Drone–AGV Collaborative	Showcase precise, low-latency D2D coordination between a UAV and an AGV for synchronized loading, with safe fallback in case of link	User Story #1 User Story #2

working with D2D	degradation, leveraging distributed AI and orchestration components.	

The figure below shows the experimental architecture for experiments of the PoC3.

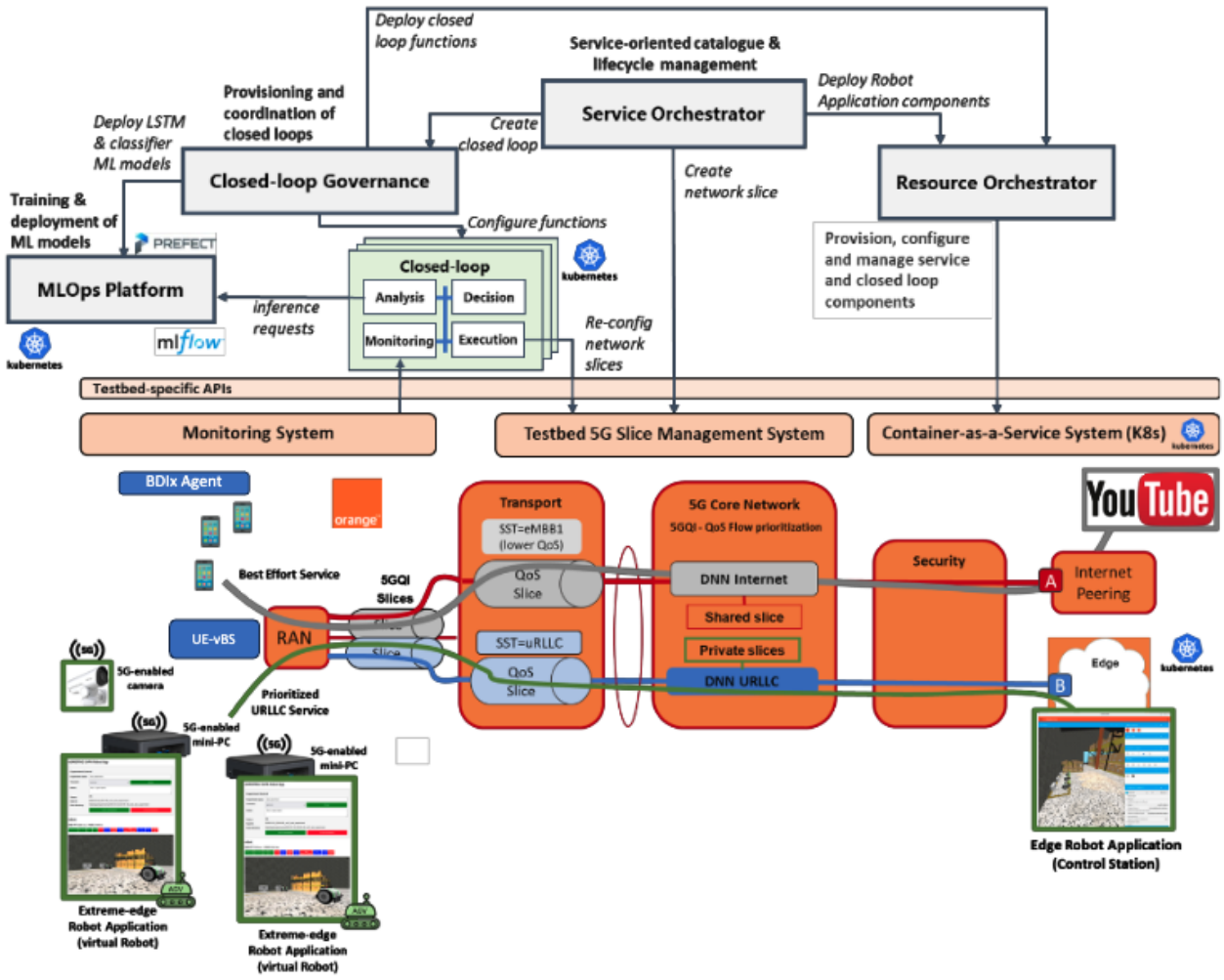

Figure 5-2: PoC3 experimental architecture

5.3.1 Experiment #1: Autonomous Transport with Dynamic Re-Routing Loading

In a modern construction site, materials need to be transported from the main storage depot to various building zones. The construction company deploys an Automated Ground Vehicle (AGV) for ground-level material transport. All devices perform the cargo delivery tasks on the ground while transmitting data to be processed to ensure best routes and avoid collisions in sites.

This experiment focuses on demonstrating autonomous material transport within a dynamic construction site environment, utilizing collaborative robots (drones, robots, and an Automated Guided Vehicle - AGV). The scenario involves intelligent task assignment and fleet selection, followed by the formation of a self-organizing of devices routes to ensure robust communication and efficient delivery. The core of the experiment validates the AGV's navigation to a loading point

and a task of routing efficiency, all while leveraging 6G capabilities for ultra-reliable, low-latency communication, precise positioning, and real-time data processing at the edge.

Objectives of this experiment:

- Establish and maintain ultra-reliable, dynamic communication for collaborative robotic fleets
- Enable real-time, ultra-low latency coordination and control for complex robotic tasks
- Validate precise positioning, time synchronization and edge-based environmental sensing and data processing.

5.3.1.1 Operationalization of experiment into a concrete Workflow

Dynamic Task Initiation, Devices Allocation and Route Planning:

The task scenario is initiated from the dashboard of the **CAFA Robot Application**. The **Service Orchestration & Resource Management** component receives this request and assure that all required resources for components are in place in the infrastructure and start deployment process.

Once all details are in place and resources set, these routes are pushed to the devices via the CAFA Robot Application. In parallel, the **CLG** verifies that network slices for control, positioning and sensor/video streams are active and correctly parameterized (latency, bandwidth, reliability) and, if needed, requests the orchestrator to adapt slice configurations to meet the experiment KPIs.

Autonomous Navigation, Data Streaming and Dynamic Re-Routing:

The robots start navigation along their assigned routes. Each device continuously uploads video and lidar sensor data via the configured network slices. The **AI/ML Pipeline Manager** processes these streams in real time and checks network parameters detecting new obstacles, congestion on the network and updated different parameter of profiles.

Whenever the AI/ML logic detects a significant change in the environment, it triggers a **dynamic configuration** decision. Updated network parameters to assure correct routes and collision-avoidance manoeuvres and communications of devices through the CAFA Robot Application. The **Service Orchestration & Resource Management** and **CLG** cooperate to adapt resource allocation and QoS policies on-the-fly (e.g., temporarily prioritizing control and positioning traffic during route changes) to preserve target latency and reliability.

ADROIT6G Components Involved: CAFA Robot Application, AI/ML Pipeline Manager, Service Orchestration & Resource Management, CLG.

5.3.1.2 Sequence Diagrams

The experiment is conducted in the following phases described below and showed in the diagram in Figure 5-3 below:

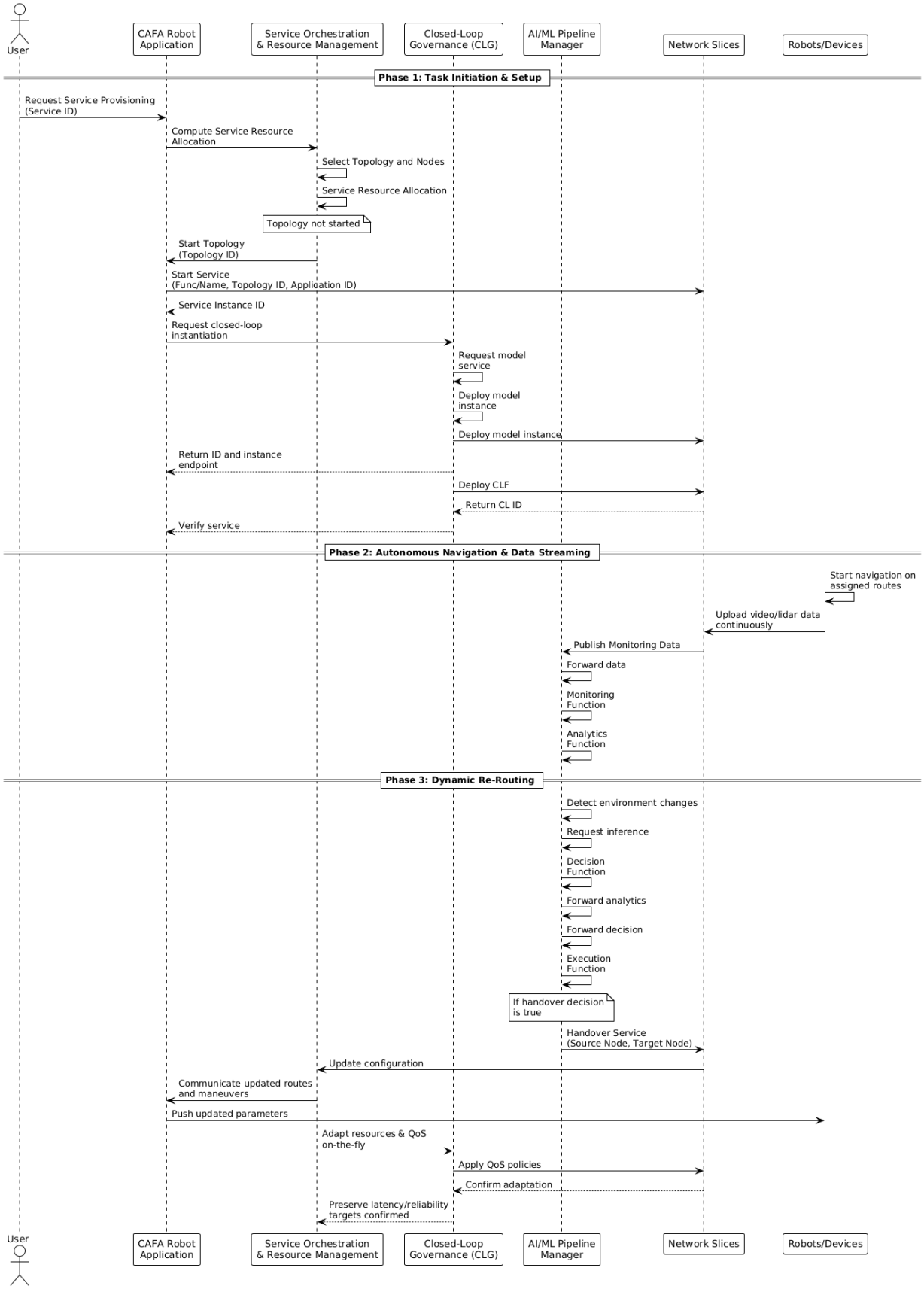

Figure 5-3: PoC3 Experiment #1 Sequence Diagram

1. **Task Initiation & Setup**
    - **Robot App** initiates task → **Service Orchestration** provisions infrastructure resources → starts deployment
    - Resources ready → routes pushed to devices via **CAFA Robot App**
    - **CLG** verifies network slices (control, positioning, sensor/video) → requests **Orchestrator** to adapt slice configurations if needed
2. **Autonomous Navigation & Data Streaming**
    - Robots start navigation on assigned routes → continuously upload video/lidar data via network slices
    - **AI/ML Pipeline Manager** processes streams in real-time → monitors network parameters, obstacles, congestion
3. **Dynamic Re-Routing**
    - **AI/ML Pipeline Manager** detects environment changes → triggers dynamic configuration decision
    - Updated network parameters & collision-avoidance maneuvers → communicated via **Robot App**
    - **Service Orchestration** + **CLG** adapt resources & QoS policies on-the-fly → preserve latency/reliability targets

5.3.1.3 Testbed setup/Configuration

The testbed for Experiment #1 consists of a 5G network infrastructure, multiple devices equipped with wireless interfaces and video cameras and sensors, and the ADROIT6G software components.

Hardware requirements:

Component	Specification	Purpose
5G Testbed / Base Station	5G NR gNodeB (FR1: Sub-6 GHz band); Support for network slicing; Edge computing capability; Open RAN compatible	Primary network infrastructure providing 5G/6G connectivity, network slicing for differentiated QoS, and edge computing for low-latency processing
robots/cobots	5G/LTE module (e.g., Quectel RM500Q or similar); WiFi Direct interface (802.11ac/ax); Onboard computing unit (e.g., NVIDIA Jetson or Raspberry Pi 4); GPS/GNSS receiver; Video camera and sensors	For autonomous transport and video processing/sensing.
Edge Computing Server	Multi-core server (Intel Xeon or AMD EPYC); Minimum 32GB RAM; GPU for	Hosts ADROIT6G components (CLG, Service Orchestration, AI/ML Pipeline Manager, Topology Module);

© ADROIT6G, 2023

	AI/ML inference (optional); Kubernetes/Docker support	Provides edge computing for low-latency decision making
Control Station	Workstation with monitoring displays; Network management interface; CAFA Robot Application dashboard	Operator interface for task assignment, monitoring, and data collection during experiment execution

ADROIT6G Software Components used in Experiment #1:

Component	Role in Experiment
CAFA Robot Application	Task-level control for robots; User interface for task assignment
Closed-Loop Governance (CLG)	KPI monitoring; Policy provisioning; Coordination with AI/ML framework; Ensures SLA compliance during operation
Service Orchestration & Resource Management	Drone deployment coordination; Network slice allocation; Resource pool management; Dynamic resource reallocation
AI/ML Pipeline Manager & Catalogue	ML model selection and deployment; Trajectory optimization; Network coverage planning; Position data processing

Setup Phases:

- **Phase A** - Infrastructure Preparation: Deploy and configure the testbed including gNodeB and edge computing server. Install and configure ADROIT6G software components (CLG, Service Orchestration, AI/ML Pipeline Manager, AI/ML Catalogue, Topology Module) on the edge server. Verify network connectivity and slice configuration.
- **Phase B** – Device Preparation: Configure each device and connect to the network. Install CAFA Robot Application on all devices. Install video camera and sensors on devices.
- **Phase C** – Network Registration: Power on all devices. Each device connects to the network.
- **Phase D** – Devices start autonomous transport and upload video and sensors data using networks. Data is processed and will adapt the network slice accordingly to assure best quality of services.
- **Phase E** – Experiment Readiness Verification. Confirm that all devices can communicate via network infrastructure. Validate that KPI monitoring is active. The system is now ready for experiment execution.

5.3.2 Experiment #2: Drone–AGV Collaborative working with D2D

In a modern construction site, materials need to be transported from the main storage depot to various building zones. The construction company deploys an Automated Ground Vehicle (AGV) for ground-level material transport and a fleet of Unmanned Aerial Vehicles (UAVs/drones). One drone performs the cargo delivery task while the remaining drones form a Device-to-Device mesh network using WiFi Direct to provide enhanced signal coverage at the loading zone. This dual-purpose

deployment addresses connectivity challenges caused by the dynamic physical environment and ensures ultra-low latency D2D communication for the synchronized loading operation.

The D2D mesh network formed by the drones extends the 6G coverage to areas where the base station signal is weak or blocked by construction obstacles. Each drone in the mesh acts as a UE-VBS node, relaying traffic between the task-executing drone, the AGV, and the infrastructure network. The WiFi Direct interfaces enable direct peer-to-peer communication with latency below 10ms, while the 5G interfaces maintain connectivity to the core network for coordination with the ADROIT6G management components.

This experiment showcases precise, low-latency D2D coordination between a UAV and an AGV for synchronized loading, leveraging distributed AI and orchestration components. A key innovation in this experiment is the deployment of devices and drones that form a Device-to-Device (D2D) mesh network to enhance signal coverage across the construction site. The drones act as mobile UE-VBS nodes, providing extended 6G/WiFi Direct connectivity in areas where traditional base station coverage is insufficient.

Objectives of this experiment:

- Demonstrate D2D network formation using devices to provide enhanced signal coverage at the construction site
- Showcase precise, low-latency D2D coordination between different types of devices (robots) or synchronized material loading
- Validate the integration of network interfaces for seamless device communication.

5.3.2.1 Operationalization of experiment into a concrete Workflow

Task Initiation, Fleet Allocation, and D2D Mesh Network Establishment:

The task scenario is initiated from the dashboard of the **CAFA Robot Application**. The **Service Orchestration & Resource Management** component receives this request and, consulting the **Topology Module**, dynamically selects a task drone for cargo delivery and coverage devices for D2D network formation based on task requirements and device availability. The selected devices receive their assignments.

The **Closed-Loop Governance (CLG)** component requests coverage verification from the **AI/ML Orchestrator**, retrieving models from the **AI/ML Catalogue**. The **CLG** sends test messages through the mesh, measures signal quality at the loading zone, and upon confirming adequate coverage (sufficient for <10ms latency), deploys operational policies (QoS parameters, relay priorities, fallback procedures) to each drone's **BDIx Agent**. All agents acknowledge policy receipt and report readiness.

ADROIT6G Components Involved: CAFA Robot Application, Service Orchestration & Resource Management, Topology Module, BDIx Agent, UE-VBS, AI/ML Pipeline Manager, CLG, AI/ML Orchestrator, AI/ML Catalogue.

© ADROIT6G, 2023

Real-Time Synchronized Devices Approach and Transfer Execution:

The task devices begin its approach to the loading location, with their trajectory managed by the **CAFA Robot Application**. The devices **BDIx Agent** continuously transmits real-time position, velocity vectors, and intended path data to the AGV via the D2D link. This critical traffic is prioritized by **UE-VBS** components through the mesh network.

As the task drone descends, the **CAFA Robot Application** coordinates the precise timing for cargo transfer, sending synchronization commands through the D2D link. The **BDIx Agents** exchange real-time status messages, including drone altitude, AGV platform status, and cargo release readiness. The synchronized cargo transfer is executed, with the drone releasing cargo at the precise moment the AGV confirms platform stability.

ADROIT6G Components Involved: CAFA Robot Application, BDIx Agent, AI/ML Pipeline Manager, UE-VBS, CLG.

Step	System Component(s)	Action	Input	Output / System Response
1	CLG; AI/ML Orchestrator; AI/ML Catalogue; BDIx Agent	Coverage verification at loading zone. Operational policies deployed to all BDIx Agents.	CLG: Coverage test messages via mesh. AI/ML Orchestrator: Retrieves models from Catalogue.	CLG: Coverage confirmed (<10ms latency achievable). BDIx Agents: Policy acknowledgment, readiness confirmed.
2	CAFA Robot Application; BDIx Agent; Service Orchestration & Resource Management	D2D link established between task devices. Network slice allocated with guaranteed QoS.	CAFA: Link request. BDIx Agents: Capability negotiation messages exchanged.	Service Orchestration: Slice allocation confirmation. BDIx Agents: Link established, <10ms latency confirmed.
3	CAFA Robot Application; BDIx Agent; Topology Module	AGV navigates to loading point using 6G positioning. Position updates relayed through D2D mesh to Topology Module.	CAFA: Navigation commands. BDIx Agent: Periodic position messages via D2D mesh.	Topology Module: Position updates received. AGV: Arrival confirmed, localization <3m verified.
4	CAFA Robot Application; BDIx Agent; CLG; AI/ML Pipeline Manager	Synchronized cargo transfer executed. Drone releases cargo to AGV with precise timing coordination.	CAFA: Sync commands via D2D. BDIx Agents: Status messages (altitude, platform, readiness).	CLG: KPIs logged AGV: Cargo receipt acknowledgment via D2D.
5	CAFA Robot Application; Service	Task completion signaled. D2D mesh network dissolved.	CAFA: Task completion signal. Service	CLG: KPI compliance report. Topology Module: Final

	Orchestration; BDIx Agent; Topology Module; CLG	Drones return to standby. KPIs evaluated against targets.	Orchestration: Mesh dissolution command.	positions, resource registry updated.

5.3.2.2 Sequence Diagrams

The experiment is conducted in the following phases described below and showed in the diagram in Figure 5-4 below:

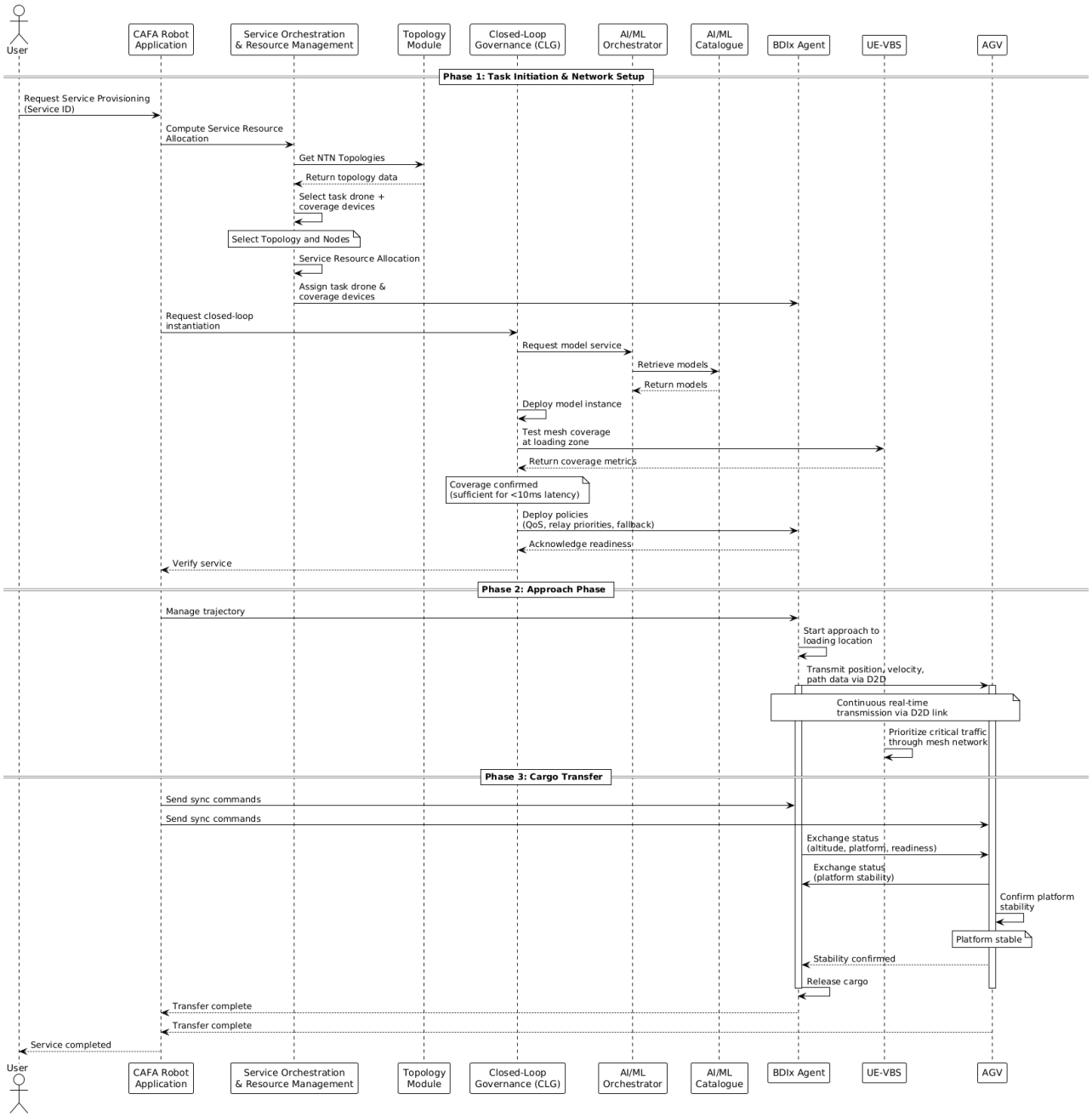

Figure 5-4: PoC3 Experiment #2 Sequence Diagram

1. **Task Initiation & Network Setup**
   - Robot App → Service Orchestration (via Topology Module) → selects & assigns task drone + coverage devices
   - CLG → AI/ML Orchestrator (retrieves models) → tests mesh coverage at loading zone
   - Coverage confirmed → CLG deploys policies → BDIx Agents acknowledge readiness
2. **Approach Phase**
   - **Robot App** manages trajectory → **BDIx Agent** transmits metrics → **AGV** via D2D
   - **UE-VBS** prioritizes traffic through mesh network
3. **Transfer**
   - **Robot App** sends sync commands → **BDIx Agents** exchange status (altitude, platform, readiness)
   - **AGV** confirms stability → **Device** releases cargo → transfer complete

5.3.2.3 Testbed setup/Configuration

The testbed for Experiment #2 consists of a 5G network infrastructure, multiple drones equipped with dual wireless interfaces (5G/LTE and WiFi Direct), an AGV with similar connectivity capabilities, and the ADROIT6G software components. The setup enables the formation of a D2D mesh network where drones serve as mobile relay nodes to extend coverage.

Hardware requirements:

Component	Specification	Purpose
5G Testbed / Base Station	5G NR gNodeB (FR1: Sub-6 GHz band); Support for network slicing; Edge computing capability; Open RAN compatible	Primary network infrastructure providing 5G/6G connectivity, network slicing for differentiated QoS, and edge computing for low-latency processing
UAV Drones/cobots	Commercial quadcopter platform; 5G/LTE module (e.g., Quectel RM500Q or similar); WiFi Direct interface (802.11ac/ax); Onboard computing unit (e.g., NVIDIA Jetson or Raspberry Pi 4); GPS/GNSS receiver; Flight time: minimum 20 minutes	Form D2D mesh network for coverage extension; Act as UE-VBS nodes; One drone performs cargo delivery task while others provide network relay
AGV	Wheeled robot platform; 5G/LTE module; WiFi Direct interface (802.11ac/ax); Onboard computing unit; Cargo receiving mechanism; Localization sensors (IMU, wheel encoders)	Ground material transport; Receives cargo from drone during synchronized loading operation; Participates in D2D communication

Edge Computing Server	Multi-core server (Intel Xeon or AMD EPYC); Minimum 32GB RAM; GPU for AI/ML inference (optional); Kubernetes/Docker support	Hosts ADROIT6G components (CLG, Service Orchestration, AI/ML Pipeline Manager, Topology Module); Provides edge computing for low-latency decision making
Control Station	Workstation with monitoring displays; Network management interface; CAFA Robot Application dashboard	Operator interface for task assignment, monitoring, and data collection during experiment execution

ADROIT6G Software Components used in Experiment #2:

Component	Role in Experiment
CAFA Robot Application	Task-level control for robots; User interface for task assignment
BDIx Agent	Autonomous decision-making on each device; D2D network formation logic; Communication with CLG and Topology Module; Policy execution
UE-VBS	Enables drones to act as mobile base stations; Extends network coverage via WiFi Direct mesh; Resource contribution to far-edge infrastructure
Closed-Loop Governance (CLG)	KPI monitoring; Policy provisioning; Coordination with AI/ML framework; Ensures SLA compliance during operation
Service Orchestration & Resource Management	Drone deployment coordination; Network slice allocation; Resource pool management; Dynamic resource reallocation
AI/ML Pipeline Manager & Catalogue	ML model selection and deployment; Trajectory optimization; Network coverage planning; Position data processing
Topology Module	Device registry; Position tracking; Network topology management; Resource availability tracking

Setup Phases:

- **Phase A** - Infrastructure Preparation: Deploy and configure the testbed including gNodeB and edge computing server. Install and configure ADROIT6G software components (CLG, Service Orchestration, AI/ML Pipeline Manager, AI/ML Catalogue, Topology Module) on the edge server. Verify network connectivity and slice configuration.
- **Phase B** - Device Preparation: Configure each device with dual wireless interfaces: 5G/LTE module connected to the testbed and WiFi Direct interface enabled for D2D mesh formation. Install BDIx Agent and UE-VBS component software on each device. Configure the AGV with identical wireless interfaces and BDIx Agent. Install CAFA Robot Application on all devices.

© ADROIT6G, 2023

- **Phase C** - Network Registration: Power on all devices. Each device connects to the network and the BDIx Agent initiates registration with the Topology Module. The CLG provisions device-specific policies and the AI/ML Catalogue deploys appropriate ML models to each BDIx Agent. UE-VBS components are configured based on device capabilities.
- **Phase D** - D2D Mesh Formation: devices designated for coverage extension establish WiFi Direct connections with each other, forming a mesh network. The BDIx Agents coordinate the mesh topology, with each drone acting as a relay node. The mesh network is registered in the Topology Module and resources are accounted in the Service Orchestration component.
- **Phase E** - Experiment Readiness Verification: Verify D2D mesh connectivity and signal coverage at the designated loading zone. Confirm that all devices can communicate via both networks' infrastructure. Validate that KPI monitoring is active in CLG. The system is now ready for experiment execution.

5.4 Application of Simulators and Emulators in Testing and Validation

The robots are simulated in the Gazebo robot simulator running in the Control Station container in the server of ORO testbed. Robots' data flow from movement command and sensors are relayed through Raspberry Pi Virtual Robots to create measurable and repeatable network traffic.

5.5 Associated KPIs and KVIs

This section presents the KPIs and KVIs associated with PoC3, which will be used to evaluate the performance of the PoC3 experiments. For consistency across the project, the selected KPIs/KVIs are aligned with those defined in D2.3 [1]. In addition, for each KPI and KVI, this section specifies the measurement approach, including the tools, data sources, and methods that will be used on the testbed to compute and assess the reported results.

Table 5-3: Service KPIs for PoC3

KPI	Description	Target Value	Measured in	Measurement Strategy
SR-Poc3-01	Latency	Communication delay <10 ms	all	Continuous monitoring through PoC dashboard & API calls
SR-Poc3-02	Reliability	SLA 99.95%	all	Continuous monitoring through PoC dashboard & API calls

© ADROIT6G, 2023

SR-Poc3-03	Energy Efficiency	Reduction of energy consumption by >30% compared to legacy	all	Specialized devices that measure energy consumption in DCs

Table 5-4: Network KPIs for PoC3

KPI	Description	Target Value	Measured in	Measurement Strategy
NR-Poc3-01	Jitter of Terrestrial Network	<1ms	Exp #1	Continuous monitoring through PoC dashboard & API calls
NR-Poc3-02	Localisation accuracy using 5G/ 6G network	<3 m	Exp #1	Using specific API calls from network components data & using localization data as GPS information
NR-Poc3-03	Upload and download speeds	≥10 Mbps per device	Exp #2	Continuous monitoring through PoC dashboard & API calls

Table 5-5: KVIs for PoC3

KVI	Description	Values/Scaling	Measurement
KVI-R-HT-I1	Safety	User-Centric perception Low / Medium / High	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation

© ADROIT6G, 2023

KVI-R-HT-I2	Security	User-Centric perception Low / Medium / High	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-I3	Regulation	# Standards used for definitions, requirements etc	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-I4	Responsibility	0 users manipulation incidences	Zero users reporting of manipulation incidences
KVI-R-HT-I5	Energy efficiency	Reduction of energy consumption by >30% compared to legacy	Energy e User-Centric perception Low / Medium / High efficiency KPI
KVI-R-HT-D1	Privacy	User-Centric perception Low / Medium / High	Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-D2	Fairness	User-Centric perception Low / Medium / High	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-D3	Digital Inclusion	99.9% communication service reliability	Relevant KPI
KVI-R-HT-D4	Trustworthiness	100% consistent coverage of the network	Relevant KPI
KVI-R-HT-E1	Sustainability	Reduction of energy consumption by >30% compared to legacy	Energy Efficiency indicator used in PoC

© ADROIT6G, 2023

KVI-R-HT-E2	Business Value	User-Centric perception Low / Medium / High	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-E3	Economic Growth	User-Centric perception Low / Medium / High	User-Centric perception Low / Medium / High Questionnaires, 3-point scale and measured by the % of users provide “high” evaluation
KVI-R-HT-E4	Open collaboration	Number of Devices connected	# of devices
KVI-R-HT-E5	New Value Chain	Number of Devices connected	# of devices

© ADROIT6G, 2023

6 Generated Datasets

This chapter presents the datasets generated during Testing Cycle 2 across the different PoCs. The sections that follow provide an overview of these datasets.

6.1 PoC #1 Datasets Generated

To facilitate the evaluation of holographic teaching applications within the test cases of PoC1, a multimodal dataset has been generated. This collection incorporates two distinct data modalities: raw volumetric capture and perspective-dependent holographic renders. The dataset provides the technical foundation for assessing spatial capture fidelity and the performance of view-dependent rendering systems.

6.1.1 Data Acquisition Parameters

Data acquisition was performed using three depth-based sensors (Azure Kinect) positioned at discrete angles to provide a multi-perspective volumetric representation of the instructor. The sensors were configured to the Narrow Field of View (NFOV) unbinned mode, characterised by the following technical specifications:

- **Depth Resolution:** 640 $\times$ 576 pixels.
- **Color Video Resolution:** 1920 $\times$ 1080 pixels.
- **Field of Illumination (FoI):** 75° (horizontal) $\times$ 65° (vertical).
- **Exposure Time:** 12.8 ms.
- **Temporal Resolution:** 30 frames per second (fps).

6.1.2 Dataset Components

The dataset comprises two primary components, each 10 seconds in duration (300 frames per sequence):

1. **Volumetric Point Cloud Data:** Raw spatial data recorded from three depth sensors. These data are stored in **Matroska Video (MKV)** files, which contain both the colour feed and the depth data stream. These files are compatible with the **Azure Kinect Viewer**, allowing for the playback and inspection of synchronised RGB-D frames. The point clouds represent the instructor's physical geometry and serve as the basis for volumetric reconstruction.
2. **Holographic Perspective Renders:** Synthetic renders of the instructor generated from 10 discrete viewing angles. These angles range from **-60° to 60°** relative to the teacher's position, corresponding to the viewing positions of students situated in an auditorium. These renders are stored as **MP4** files and facilitate the analysis of spatial consistency and visual quality in holographic displays across a wide viewing arc.

© ADROIT6G, 2023

6.1.3 Data Utility and Evaluation

The structured nature of this dataset allows for the benchmarking of several system components:

- **Spatial Capture Accuracy:** Evaluating the alignment of point cloud data from multiple sensors under specific illumination and exposure constraints.
- **View-Dependent Rendering Fidelity:** Assessing the quality of holographic renders across the 120-degree viewing arc to ensure visual consistency for distributed users.
- **Temporal Stability:** Analysing the consistency of 3D reconstructions at 30 fps to support real-time teaching applications.

6.1.4 Dataset Specifications

The technical organisation of the data files is summarised in the table below:

Attribute	Specification
Modality A	Raw Point Cloud (3 source angles)
Modality B	Holographic Renders (10 viewing perspectives)
File Formats	.mkv (Volumetric), .mp4 (Renders)
Sequence Duration	10 seconds
Temporal Resolution	30 fps
Depth Format	NFOV unbinned (640 $\times$ 576)
Color Format	RGB (1920 $\times$ 1080)
Angular Range	-60° to +60° (10 increments)

6.2 PoC #2b Datasets Generated

6.2.1 Satellite Connectivity Data for Dataset generation

To generate a comprehensive satellite connectivity map for a specific location, we developed an emulated experiment designed to simulate satellite communication. This experiment enabled us to systematically gather data on coverage over time for the designated ground location, thus supporting the generation of synthetic data used by the AI/ML solutions in PoC #2b. For Testing Cycle 2, a new and longer dataset was generated to provide a broader and more diverse collection of information. This is being used in the Experiment #1, as part of the analytics function of the closed-loop.

To do so, the implemented setup (as depicted in Figure 6-1) considers a ground device that monitors its connectivity towards available satellites over an emulated satellite network, thus allowing us to

generate an initial dataset that records the available satellite connections over time. This initial dataset provides the following information (directly or indirectly) for each moment in time:

1. Temporal distance (in terms of latency) between the ground entity and each satellite).
2. Number of satellites in line of sight with the device.
3. The time window that each satellite has line of sight to the device.

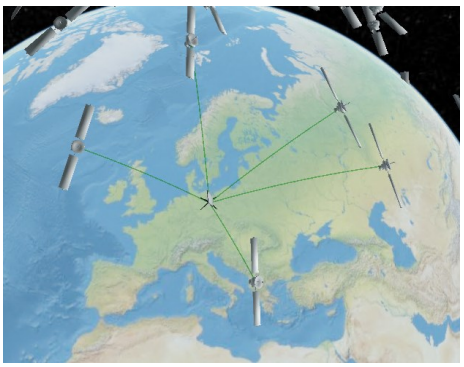

Figure 6-1: Emulated experiment of Satellite Connectivity

The ground device is emulating the maximum number of connections to all available satellites, which distance is measured in terms of latency. To do so, the ground device continuously pings the satellites it is connected to in one-second intervals.

The outcome is an initial dataset, provided as a CSV file, including the following fields:

- **epoch:** The date and time when the data was recorded
- **satellite_id:** Unique identifier of the satellite
- **distance_time:** Represents the distance between the ground device and the satellite, modelled as latency time in milliseconds (ms)

6.2.2 Application Performance Data with Background Traffic for Dataset generation

To gather information about the impact of background traffic on the performance of a given application, we developed an emulated experiment where two ground nodes are connected via (emulated) satellite communication and background traffic is injected into the communication paths. This experiment enabled us to systematically gather data on the application performance over time for the designated ground location, which is being used by the AI/ML solutions in PoC #2b. For testing cycle 2, a new and longer dataset was generated to provide a broader and more diverse collection of information. This is being used in the Experiment #3.

To do so, the implemented setup considers a ground device (client application) that is connected to another ground device (server application) via the available (emulated) satellites. Client application establishes two redundant flows with the server application, each going through a distinct satellite path. The client application exchanges state information at a rate of 3Hz with its counterpart server application.

The same experiment is run for different scenarios, where background traffic is injected into the communication paths:

1. No background traffic between application client and server.
2. Background traffic between application client and server only on a single flow path.
3. Background traffic between application client and server on both flow paths.

This initial dataset provides the following information (directly or indirectly) for each moment in time:

1. Impact of moving satellites in the E2E application.
2. Impact of background traffic in the E2E application.

The outcome is an initial dataset, provided as a CSV file, including the following fields:

- **point_id:** Unique identifier for the application entity.
- **sn:** Unique sequence number for the transmitted message for a given scenario.
- **delay_ms:** E2E latency time in milliseconds (ms).
- **delta_ms:** Time between redundant messages (at the receiver side) for the same SN in milliseconds (ms).
- **lost_sn:** Represents a SN that had all the redundant packets lost (true if 1, otherwise 0).
- **lost_flow:** Number of missed packets for a given SN between all the redundant flows.
- **net_bps:** Bytes per second at the given time slice.
- **scenario:** Identifies the measured scenario.

6.3 PoC2a and PoC #3 Datasets Generated

The validation process was methodically designed to assess the core functional blocks of the AI/ML framework, including model training, deployment, resource and service orchestrator. The test cases were executed in a controlled environment, ensuring that network configurations, device setups, and monitoring systems did not interfere with the results. Data collection was performed through monitoring systems that logged key metrics such as Bytes Received/Sent, throughput rates, and latency, across various traffic patterns and network slices configurations. Monitoring agents were deployed across all components from the testbed and data generated from the network/testbed part related to monitoring data from compute and network resources.

Data related to traffic (based on the slice), example below:

Timestamp	2024-10-07 13:32:15.000 UTC
BytesReceived_URLLC	125650180
BytesSent_URLLC	318333011
URLLC_BytesReceived,	375000
URLLC_BytesSent,	4132500
URLLC_Received_thrp_Mbps,	0.4293848
URLLC_Sent_thrp_Mbps	6.41398

Data related to monitoring compute resources, example below:

node_cpu_seconds_total{cpu="24",mode="nice"} 9.36
node_cpu_seconds_total{cpu="24",mode="softirq"} 2289.6
node_cpu_seconds_total{cpu="24",mode="steal"} 0
node_cpu_seconds_total{cpu="24",mode="system"} 49257.24
node_cpu_seconds_total{cpu="24",mode="user"} 511764.42
node_cpu_seconds_total{cpu="25",mode="idle"} 5.05240697e+06
node_cpu_seconds_total{cpu="25",mode="iowait"} 2800.12

6.3.1 Description of data logged by CAFA Robot app during the experiments

The dataset contains data of number of robots, their commands and events, and latency of commands and video frames. Each experiment run creates a new directory in the form:

YYYY-MM-DD_HH-MM-SS_<scenario>_<experimentname>

Example:

2026-04-02_12-40-42_07_poc2a_modeA_patrol_run

Each experiment directory contains following files:

- cmd_latency.csv
- video_latency.csv
- events.csv
- meta.json
- scenario.csv

6.3.2 Description of the files

cmd_latency.csv: Contains command round-trip latency measurements.

Example:

t_wall_iso,robot,video_mode,qos_mode,lidar_state,cmd_id,linear,angular,rtt_ms
2026-04-02T12:40:44.488,robot_1,LOW,BEST_EFFORT,ON,1810aa3286de425ba100d551a70784bf,-0.6,0.0,24.773
2026-04-02T12:40:44.488,robot_2,LOW,BEST_EFFORT,ON,ebc071050ccb4b9ba0f4f0a886998f5d,-0.6,0.0,18.224
2026-04-02T12:40:54.491,robot_1,LOW,BEST_EFFORT,ON,07e7a6575b234bf0b4df8b5bc0e332ed,0.0,0.5,28.314

video_latency.csv: Contains video frame latency measurements.

Example:

t_wall_iso,robot,video_mode,qos_mode,lidar_state,frame_stamp_sec,frame_stamp_nsec,latency_ms,bytes
2026-04-02T12:40:42.487,robot_1,LOW,BEST_EFFORT,ON,1775133642,411871737,53.589,60378

2026-04-02T12:40:42.488,robot_2,LOW,BEST_EFFORT,ON,1775133642,349025288,40.093,64192
2026-04-02T12:40:42.988,robot_1,LOW,BEST_EFFORT,ON,1775133642,806687804,67.363,69322

events.csv: Records system events during the experiment.

Example:

t_wall_iso,robot,event,value,extra_json
2026-04-02T12:40:42.100,ALL,RUN,START,"{""name"": """", ""scenario"":
""07_poc2a_modeA_patrol.csv""}"
2026-04-02T12:40:42.100,robot_1,SCENARIO,VIDEO,"{""cmd"": ""VIDEO LOW""}"
2026-04-02T12:40:42.100,robot_1,SCENARIO,QOS,"{""cmd"": ""QOS BEST_EFFORT""}"
2026-04-02T12:40:42.101,robot_1,SCENARIO,LIDAR,"{""cmd"": ""LIDAR ON""}"
2026-04-02T12:40:42.101,robot_2,SCENARIO,VIDEO,"{""cmd"": ""VIDEO LOW""}"

meta.json: Contains metadata about the experiment.

Example:

{
 "run_id": "2026-04-02_12-40-42_07_poc2a_modeA_patrol_run",
 "started_at": "2026-04-02_12-40-42",
 "name": "",
 "notes": "",
 "scenario": "07_poc2a_modeA_patrol",
 "robots": [
  "robot_1",
  "robot_2"
 ],
 "scenario_file": "/data/logs/experiments/2026-04-02_12-40-
42_07_poc2a_modeA_patrol_run/scenario.csv"
}

scenario.csv: The experiment scenario file.

Example (just a few lines from the start):

seconds_from_start, robot_id, command
0,robot_1,VIDEO LOW
0,robot_1,QOS BEST_EFFORT
0,robot_1,LIDAR ON
0,robot_2,VIDEO LOW
0,robot_2,QOS BEST_EFFORT
0,robot_2,LIDAR ON
2,robot_1,MOVE -0.6 0.0
2,robot_2,MOVE -0.6 0.0
12,robot_1,MOVE 0.0 0.5
12,robot_2,MOVE 0.0 -0.5

© ADROIT6G, 2023

7 Conclusions

This deliverable reported the work performed in Task 5.2 during Testing Cycle 2, which represents the final stage of the ADROIT6G technology validation activities in lab environments before the quantitative performance evaluation phase. In contrast to D5.3 [2], which focused on the validation of smaller functional blocks, the work presented in D5.4 advanced to the validation of integrated operational workflows across the ADROIT6G framework stack, aiming to verify that the different components developed in WP3 and WP4, and integrated under WP5, can operate together correctly, in a coordinated and interoperable manner, within the context of the project's end-to-end PoCs. In this respect, the deliverable captures the transition from partial functional validation to PoC-level integrated validation, providing the final technical confirmation that the ADROIT6G platform can function as a cohesive, stable, and integrated system at PoC level, thereby representing an important milestone in the overall technical maturity of the project.

More specifically, the deliverable focused on the validation of the four PoC streams executed during Testing Cycle 2, namely: i) PoC1 – Holographic Teaching; ii) PoC2a – Terrestrial 6G IIoT with extreme mMTC; iii) PoC2b – Rail Automation Support by Non-Terrestrial Networks; and iv) PoC3 – Collaborative Construction Robots. Across these PoCs, the work validated integrated experiment workflows that combine architectural elements from the AI-driven Management and Orchestration framework, the Distributed AI / CrowdSourcing AI framework, and the BDI- and AI-driven control framework, as defined in the ADROIT6G architecture and further developed in the corresponding WP3 and WP4 deliverables.

The validation activities reported in this deliverable confirm that the ADROIT6G components, when combined into concrete experiment workflows, are capable of supporting the intended PoC logic and of realising well-scoped portions of the target end-to-end scenarios. In this respect, D5.4 provided the final technical confirmation that the designed interfaces, APIs, coordination procedures, and closed-loop interactions among the involved frameworks are sufficiently interoperable and synchronised to support the execution of the PoCs in realistic lab-based settings. This is a key outcome of Testing Cycle 2, as it demonstrates the transition from isolated or partially integrated technical blocks towards cohesive PoC-level validation.

Overall, D5.4 marks the culmination of the integration and validation efforts of ADROIT6G under Task 5.2. The work presented here demonstrates the operational maturity of the developed technologies and confirms that the PoCs have progressed to a stage where the integrated workflows can be executed and assessed as coherent ADROIT6G demonstrations. It is important to stress, however, that D5.4 does not present the final quantitative performance evaluation results of the PoCs. Its purpose is to document the final technology validation in lab environments and to verify that the PoCs are correctly prepared, integrated, and operational for such assessment. The measurement, analysis, and critical assessment of the PoCs against the project's Key Performance Indicators (KPIs) and Key Value Indicators (KVIs), as defined in the ADROIT6G architecture and methodology, is carried out in Task 5.3 and reported in Deliverable D5.6.

In conclusion, the work reported in this deliverable confirms that ADROIT6G has successfully completed the final phase of technology validation in lab environments by demonstrating integrated PoC workflows across its targeted use cases. This provides a solid basis for the final

© ADROIT6G, 2023

performance evaluation phase and for the overall demonstration of the project's technological maturity, innovation potential, and contribution towards future AI-driven, open, programmable, and distributed 6G network architectures.

References

[1] ADROIT6G Consortium, "D2.3 Requirements, use cases and system architecture," 2024.
[2] ADROIT6G Consortium, “D5.3: ADROIT6G technology validation in lab environments – Initial” 2025
[3] ADROIT6G Consortium, “D5.2 - Reports on integration and testbeds deployment in lab settings - Final”, 2025