# D5.1: Preliminary experimental setup and data set collection

HEU 6G SNS JU Project

Grant No. 101096466

Co-funded by the European Union

# Document properties

| | |
|---|---|
| **Document number** | **D 5.1** |
| **Document title** | Preliminary experimental setup and data set collection |
| **Work Package** | 5 |
| **Editors** | Adam Zahir, Milan Groshev (UC3M) |
| **Authors** | Francesco Paolucci, Michelangelo Guaitolini (CNIT), Luis Velasco, Marc Ruiz, Jaume Comellas, Sima Barzegar (UPC), Milan Groshev, Adam Zahir, Carlos J. Bernardos, Pablo Picazo, Carlos Barroso, Alejandro Calvillo (UC3M), Chrysa Papagianni, Cyril Hsu (UVA), Simon Pryor, Revaz Berozashvili (ACC), Péter Vörös, Sándor Laki (ELTE), Chathuranga Weeraddana, Charbel Bou Chaaya (UOU), Roberto Gonzalez (NEC), Gergely Pongrácz, Dávid Kis (ERI-HU), Anastassios Nanos (NUBIS), Rafael López Da Silva, Marta Blanco Caamaño (TID), Andrea Sgambelluri, Alessandro Pacini (SSSA), Vincent Lefebvre, Mark Angoustures (TSS) |
| **Internal reviewers** | 1. Francesco Paolucci (CNIT) 2. Luis Velasco (UPC) |
| **External reviewers** | 1. Anestis Dalgkitsis (UVA) 2. Gergely Pongrácz (ERI-HU) |
| **Dissemination level** | PU (Public) |
| **Status of the document** | Final |
| **Version** | 1.0 |
| **File name** | DESIRE6G_D5_1_Preliminary_experimental_setup_dataset_collection |
| **Contractual delivery date** | January 31, 2024 |
| **Delivery date** | January 31, 2024 |

# Document history

| Revision | Date | Issued by | Description |
|---|---|---|---|
| V0.1 | 30/11/2023 | UC3M | First draft |
| V0.2 | 08/01/2024 | UPC/CNIT | Internal review version |
| V0.3 | 15/01/2024 | UVA/ERI-HU | External review version |
| V1.0 | 26/01/2024 | UC3M | Complete and clean version |

# Abstract

This document presents the initial plans to design the DESIRE6G testbed sites, encompassing the integration of developed technological elements and the formulation of demonstration scenarios relevant to the different DESIRE6G use cases. In addition, the document defines the relevant data sets that will be collected in the project. The document is organized as follows. Firstly, it outlines the experimentation sites to be used for integration, verification, and validation of the DESIRE6G solution. Secondly, it defines the demonstrations/Proofs-of-Concept that will be developed in the project. Thirdly, it addresses data collection approaches to enable AI/ML processing. Finally, it describes the next steps for executing on the integration and demonstration plans.

# Keywords

Testbed, demonstration, dataset, PoC, Digital Twin, robot control, AR/VR, Hardware, Software

# Disclaimer

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101096466.

DESIRE6G is supported by the Smart Networks and Services Joint Undertaking.

# Table of Contents

# List of Figures

# List of Tables

# List of Acronyms

| | |
|---|---|
| **xAPP** | Software Application for near-real time RIC automation and management |
| **3GPP** | 3rd Generation Partnership Project |
| **AAS** | Advanced Antenna Systems |
| **AD** | Administrative Domain |
| **AF** | Application Function |
| **AI** | Artificial Intelligence |
| **AP** | Access Point |
| **API** | Application Programming Interface |
| **AR** | Augmented Reality |
| **BGP** | Border Gateway Protocol |
| **BHT** | Binary Hardening Tool |
| **CN** | Core Network |
| **CP** | Control Plane |
| **CPU** | Central Processing Unit |
| **CU** | Central Unit |
| **DLT** | Distributed Ledger Technology |
| **DPDK** | Data Plane Development Kit |
| **DT** | Digital Twin |
| **DPDK** | Data Plane Development Kit |
| **DPU** | Data Processing Unit |
| **DU** | Distributed Unit |
| **eBPF** | extended Berkeley Packet Filter |
| **eMBB** | enhanced Mobile Broadband |
| **FL** | Federated Learning |
| **gNB** | gNodeB |
| **GPU** | Graphics Processing Unit |
| **GRE** | Generic Routing Encapsulation |
| **HW** | Hardware |
| **IETF** | Internet Engineering Task Force |
| **INT** | In-band Network Telemetry |

| | |
|---|---|
| **I/O** | Input/Output |
| **IPsec** | IP Security |
| **KPI** | Key Performance Indicator |
| **KVM** | Kernel-based Virtual Machine |
| **K8S** | Kubernetes |
| **LAN** | Local Area Network |
| **MAS** | Multi-Agent System |
| **ML** | Machine Learning |
| **MLFO** | Machine Learning Function Orchestrator |
| **MMTC** | Massive Machine-Type Communications |
| **NN** | Neural Network |
| **NF** | Network Function |
| **NF-CP** | Network Function Control Plane |
| **NF-DP** | Network Function Data Plane |
| **NFR** | Network Function Routing |
| **NFV** | Network Function Virtualization |
| **NIC** | Network Interface Card |
| **IML** | Infrastructure Management Layer |
| **OAI** | OpenAirInterface |
| **PC** | Personal Computer |
| **PoC** | Proof of Concept |
| **PoP** | Points of Presence |
| **P4** | Programming Protocol-Independent Packet Processors |
| **QoS** | Quality of Service |
| **RAM** | Random Access Memory |
| **RAN** | Radio Access Network |
| **REST** | Representational State Transfer |
| **RIC** | Radio Intelligent Controller |
| **RIS** | Reflecting Intelligent Surface |
| **ROS** | Robot Operating System |
| **RPC** | Remote Procedure Call |
| **RRU** | Remote Radio Unit |

| | |
|---|---|
| **RT** | Real Time |
| **RTK** | Real-Time Kinematics |
| **SA** | Stand Alone |
| **SDN** | Software-Defined Networking |
| **SDNc** | Software Defined Network controller |
| **SECaaS** | Security as a Service |
| **SLA** | Service Level Agreement |
| **SLAM** | Simultaneous Localization and Mapping |
| **SC** | Smart Contract |
| **SMO** | Service and Management Orchestrator |
| **SW** | Software |
| **TSN** | Time-Sensitive Networking |
| **UAV** | Unmanned Aerial Vehicle |
| **UE** | User Equipment |
| **UPS** | Uninterruptable Power Supply |
| **URLLC** | Ultra-Reliable Low-Latency Communications |
| **USRP** | Universal Software Radio Peripheral |
| **VIM** | Virtual Infrastructure Manager |
| **VR** | Virtual Reality |
| **VM** | Virtual Machine |
| **VNF** | Virtual Network Function |
| **VXLAN** | Virtual Extensible LAN |
| **V2X** | Vehicle-to-Everything |
| **WP** | Work Package |

# Executive Summary

One of the main goals of DESIRE6G is to integrate, validate and demonstrate the technological components developed in WP3 and WP4. This first deliverable of DESIRE6G Work Package 5 focuses on the testbed definitions, integration, demonstration plans and dataset collection. It first defines the DESIRE6G real-world testbeds and provides the initial plans for deploying and integrating components and Proofs-of-Concept (PoCs) into those real-world testbeds. Later this deliverable describes PoCs for two use cases defined in WP2, and provides the ways to implement PoCs into defined DESIRE6G testbeds.

# Key Contributions

The key achievements in this deliverable are highlighted below:

- Real-world integration and evaluation sites description, i.e., (5TONIC, ARNO), including available technologies and services that the partners of the DESIRE6G contribute to the project.
- Initial integration plan of the technologies and services into those testbeds.
- Detailed description of the main project PoCs derived from the DESIRE6G use-cases defined in WP2.
- A roadmap of each PoC throughout the timeline of the project.
- Detailed description of the demonstration scenarios including what DESIRE6G features the selected scenarios show and why exactly we need those features for a given scenario.
- An evaluation framework to measure the performance metrics of each PoCs and DESIRE6G component.
- Initial datasets that will be collected and/or used.

# 1. Introduction

One of the key objectives of DESIRE6G WP5 is to evaluate the benefits of a DESIRE6G solution through of proof-of-concepts (PoCs) featuring high-throughput and low-latency demanding applications using Smart Orchestration and deep programmability in real-world environments. The goal of WP5 is to integrate the technology components developed in WP3 [1] and WP4 [2]. This integrated platform will be used to experimentally validate all these components into real world multifunctional testbeds. Each of the components will be first developed and tested locally in local testbeds, that later in the project will be integrated in one of the two main global testbeds namely (1) 5TONIC in Madrid and (2) ARNO in Pisa.

This deliverable D5.1 presents a description of the global and local testbeds used in DESIRE6G, the generated and used datasets, and also defines the PoCs related to the use cases that will be deployed and tested in these testbeds during the project.

**Section 2** describes the testbeds in real-world environments. It highlights the local and global testbeds with the technologies and infrastructures to be integrated in each site and presents the corresponding integration plan for the Year 2 and Year 3 of the project. In addition, it describes the federated testbeds with their specific goal in developing the DESIRE6G components.

**Section 3** describes the initial planning of the PoCs relevant to the use cases defined in WP1. For each PoC we show the relation with project objectives, relation with use case, Software and Hardware configuration, ways to measure performance metrics and interim results. Required technological components developed in WP2 and WP3 are also highlighted in this section.

**Section 4** describes the demonstration scenario for the two PoC for the first half of the project. In this section, we present the demo scenario and the demo execution plan for the Digital Twin and the Intelligent and resilient VR/AR applications with perceived zero latency DESIRE6G PoCs.

**Section 5** describes the initially identified data sets that will be created or used in the project while developing the specific DESIRE6G components and PoCs.

Finally, **Section 6** concludes with the next steps for the integration and evaluation of the defined PoCs in the two testbeds.

# 2. Initial testbeds design

This section describes the different test sites that will be involved in the experimental evaluation of the project for Year 2 and Year 3. It highlights the technologies and infrastructures to be integrated in each site, distinguishing between **global** and **local** testbeds. Global testbeds are large-scale sites that will host both the Digital Twin (DT) and the Augmented Reality/Virtual Reality (AR/VR) main use cases of the project. Local testbeds, in contrast, are smaller-scale setups that will be used for DESIRE6G individual component development and validation.

Table 1 provides the identified DESIRE6G components that will be developed in the local testbeds and later integrated into the global testbeds and the representative PoCs. Please note that these integrations are preliminary and will be revised throughout the duration of the project. For any modifications we will provide information in D5.2.

| DESIRE6G component | Local testbed | Global testbed | PoC |
|---|---|---|---|
| SMO | Service and Management Orchestrator testbed | ARNO, 5TONIC | PoC#1, PoC#2 |
| IML | NPT-Budapest | ARNO, 5TONIC | PoC#1, PoC#2, PoC#3 |
| MAS | Multi-Agent System testbed | ARNO | PoC#1 |
| MLFO | Multi-Agent System testbed | ARNO | PoC#1 |
| ALTO | Technology & Automation LAB | None | None |
| BHT | TSS DLT-back mutual attestation and runtime monitoring testbed | None | None |
| RAN accelerated components | OPENLAB | None | PoC#1 |
| DLT federation | None | 5TONIC | None |
| SOL | None | ARNO, 5TONIC | PoC#1, PoC#2 |

TABLE 1. INTEGRATION OVERVIEW OF DESIRE6G COMPONENTS FROM LOCAL TO GLOBAL TESTBEDS

## 2.1. Global testbeds

DESIRE6G PoC development and experimental validations will be carried out in the two main global testbeds, namely: 1) 5TONIC testbed presented in Section 2.1.1 and 2) ARNO testbed presented in Section 2.1.2.

### 2.1.1. 5TONIC

#### 2.1.1.1. Description

5TONIC is an open co-creation laboratory focused on 5G technologies, founded by Telefónica and IMDEA Networks in 2015 and based in Madrid, Spain. 5TONIC aims to create an open global environment where industry and academic members can work together on research and innovation projects related to 5G technologies.

It provides facilities for conducting experiments and pilot deployments in realistically dense scenarios, encompassing both infrastructure and users. The testbed includes a full commercial 5G Stand Alone (SA) deployment provided by Ericsson, which supports the most common 5G services, including enhanced Mobile Broadband (eMBB), massive Machine Type Communication (mMTC) and Ultra-High Reliability & Low Latency (URLLC). Additionally, 5TONIC provides a portable 5G network, also sourced from Ericsson, for use in on-premises or event demonstrations. This portable network includes a RAN and the 5G Core Network (CN) User Plane, which are the elements positioned close to the users, and allows to connect with the central 5TONIC core for managing the control plane.

In the project, the main goals of the 5TONIC testbed are:

- To validate the DESIRE6G components via the DESIRE6G version of the Digital Twin (DT) use case, where the E2E DT system is cloud-native and distributed across the Device-Edge-Cloud continuum. This system will balance the computing, storage, and networking requirements of each virtual function, supporting *extreme low latency.*
- To develop, deploy, and validate the neural network (NN) acceleration component. This involves integrating a neural network compilation and acceleration virtual function, developed during the

project, into the 5TONIC testbed. The virtual function will host a version of the SOL[1] acceleration software, specifically optimized for the network setting.

- To develop and integrate Machine Learning (ML) algorithms specifically designed for federated learning at the DESIRE6G edge site. This involves development of federated learning algorithms with robustness with respect to dataset heterogeneity. The considered federated setting naturally guarantees the privacy of local datasets. The federated algorithms developed and tested locally within software are to be integrated into 5TONIC framework.

As shown in Figure 1, the 5TONIC testbed includes designated areas for both indoor and outdoor experimentation. Laboratory X3, represented in Figure 2, serves as the indoor testing zone and offers connectivity to the 5G SA network as well as various User Equipment (UEs) for interaction. Outdoor experiments will be conducted on the basketball courts, which are equipped with different Access Points (APs) that are also connected to the 5G SA network.

[1] https://sol.neclab.eu/


FIGURE 1. 5TONIC PREMISES AT IMDEA NETWORKS


FIGURE 2. 5TONIC INDOOR EXPERIMENTATION AREA (LABORATORY X3)

The last block of the 5TONIC testbed is SLICES[2], a designated area for developing mobile virtualized and programmable networks. This area is currently under development at UC3M premises, and it includes an open-source mobile network equipped with O-RAN and OPENAIR interfaces (OAI). The open-source nature allows for extensive modifications and adaptations, enabling researchers and developers to implement and test 6G RAN functionalities.

FIGURE 3. 5TONIC MOBILE VIRTUALIZATION AND PROGRAMMABILITY AREA (SLICES)

Finally, Figure 4 illustrates the 5TONIC data center, which consists of multiple server racks that offer substantial computing capabilities. The data center is integrated into the 5TONIC infrastructure and provides direct connectivity to the 5G network, ensuring high-speed data transfer and real-time processing.

[2] https://www.slices-ri.eu/

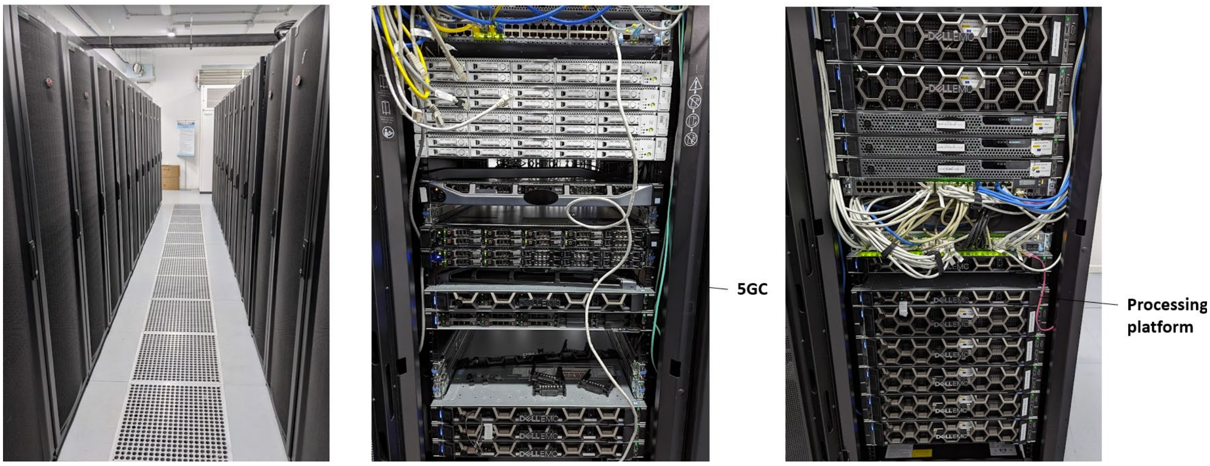


FIGURE 4. 5TONIC DATA CENTER

### 2.1.1.2. Overview of integrated components

The 5TONIC testbed will incorporate multiple technological components that will facilitate the execution of various experiments to verify and validate the DESIRE6G solution. Figure 5 presents a functional diagram of the testbed.

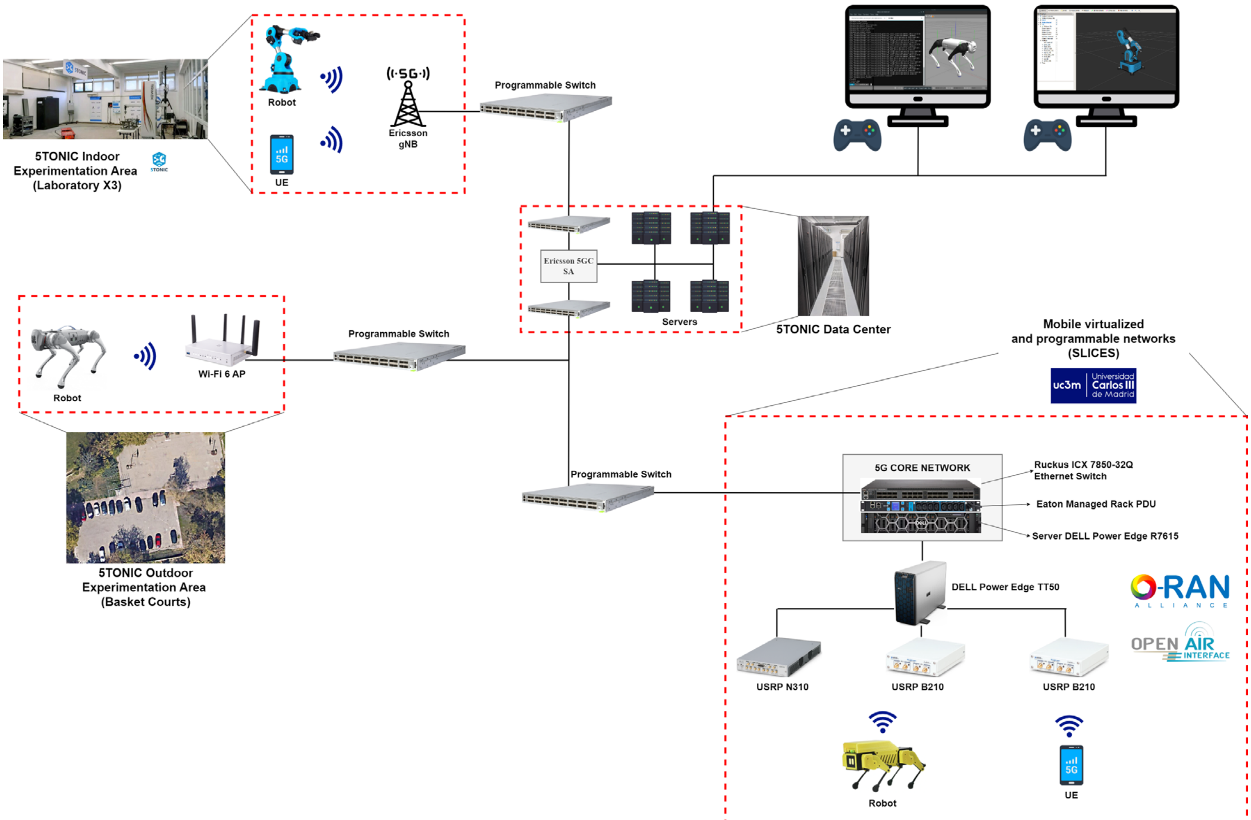


FIGURE 5. INTEGRATED TECHNOLOGIES IN THE 5TONIC TESTBED

We elaborate on each of the essential hardware and software components of the 5TONIC testbed below.

- **End devices:** Serving as the user end points of the network, the testbed includes:
  - **Robots**: Pre-built robots for conducting simulations and experimental activities related to the DT use case.
  - **User Equipment's**: Various types of UEs are integrated to test and validate user experience and network performance under different scenarios.
- **Radio Access Network (RAN):** The 5TONIC testbed incorporates a variety of RAN components designed to provide robust and high-speed wireless connectivity both indoors and outdoors. The RAN setup is composed of three main categories of components:
  - **5G SA commercial components:** This category consists of commercial-grade devices such as Advanced Antenna Systems (AAS), radio dots, remote radio units, and radio points. Figure 6 and Figure 7 show the locations of the deployed components in the testbed. For detailed specifications, please refer to Table 35 in the Annexes.

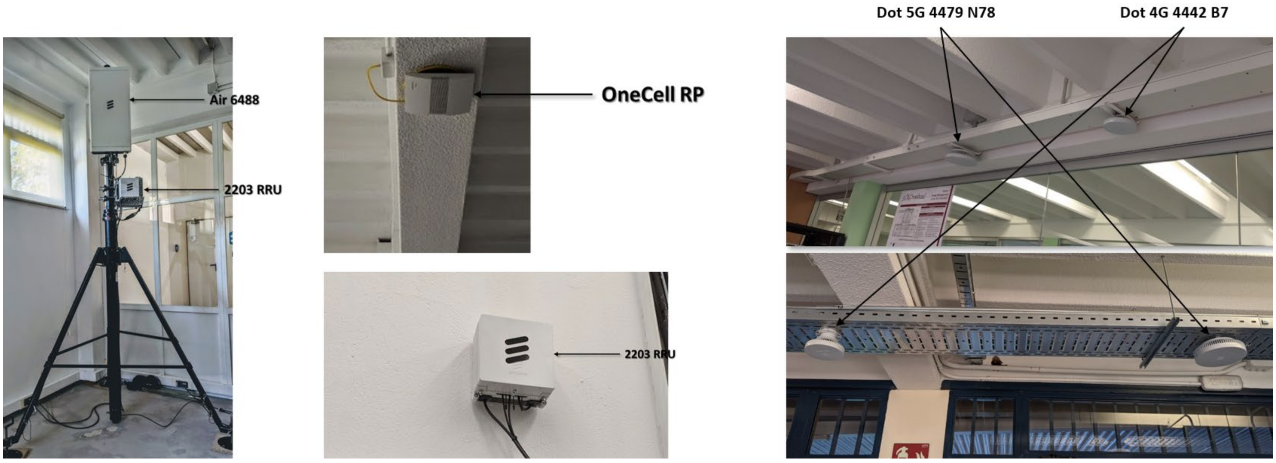


FIGURE 6. 5TONIC INDOOR COVERAGE


FIGURE 7. 5TONIC OUTDOOR COVERAGE

- **Wi-Fi 6 Access Points:** These will be primarily used for outdoor experimental activities.
- **SLICES components**: The SLICES segment features Universal Software Radio Peripheral (USRP) devices, including the *USRP Ettus B210* and *USRP Ettus N310* models. These devices are capable of implementing gNodeB (gNB) functionalities and are paired with a powerful computing Personal Computer (PC) for control and data processing. The architecture of this setup is depicted in Figure 8. Various UEs are supported for connecting to the 5G network through these USRPs. The 5G core network of this area is comprised of the following components:
    - **Ruckus ICX 7850-32Q ethernet switch:** Switch with 32 x 40/100 Gbps QSFP28 ports, allows to interconnect the different resources within the infrastructure.
    - **Dell Power Edge R7615 server:** Robust compute resource featuring 2x64 core EPYC 7742 processors, dual 100Gbps connectivity, and 128GB of RAM. It provides the necessary resources to deploy the open programmable mobile network.
    - **EATON EMAH28:** Power feed equipment which includes a manageable Power Distribution Unit (PDU).

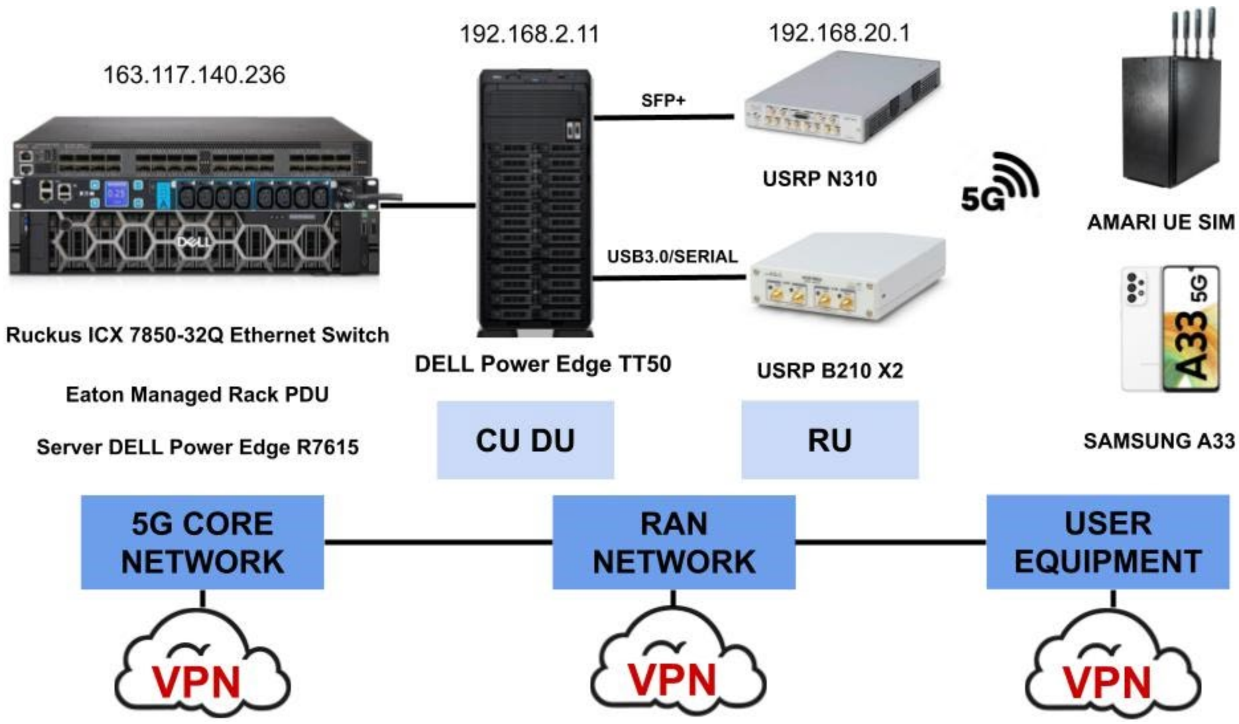


FIGURE 8. ARCHITECTURE OF SLICES

- **Network transport:** Programmable switches are the backbone of the data center, tasked with the intelligent management and routing of data traffic. Notable models include the *Switch SDN Wedge100BF-32QS* and *Switch SDN Wedge100BF-32X*, both using Barefoot Tofino technology. These models come with 32 x 100G QSFP28 ports, ensuring high-speed data transmission. They possess notable characteristics such as programmability, which permits personalized packet handling using a P4 pipeline, and real-time telemetry. Additionally, they offer efficient traffic distribution through Layer 4 server load balancing and improved security via network packet brokering.
- **Data center servers:** These servers are equipped with both CPU and GPU processing to optimize complex computational tasks. The GPUs, in particular, are specifically suited to meet the demands of machine learning (ML) training and robotic simulations from the DT application. For more information about the hardware specifications of servers in the 5TONIC data center, please refer to Table 36 in the Annexes.

### 2.1.1.3. Integration and validation timeline

Table 2 shows the integration and validation timeline of the 5TONIC testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q4 – 2023 | • Integration of hardware and software components |
| Q1 – 2024 | • Set up the basic network infrastructure.<br>• Deployment of the DT use case<br>• Conduct baseline experiments |
| Q2 – 2024 | • Deployment of DESIRE6G components inside 5TONIC resources<br>• Conduct experiments |
| Q3 – 2024 | • Integration of the DESIRE6G components in the DT use case<br>• Preliminary demonstration (Y2 demo) |
| Q2 – 2025 | • Validation of final DESIRE6G components within the 5TONIC test site |
| Q4 – 2025 | • Final demonstration of integrated DESIRE6G solution for the DT use case (Y3 demo) |

TABLE 2. 5TONIC TESTBED TIMELINE

### 2.1.2. ARNO

#### 2.1.2.1. Description

The Advanced Research on Networking (ARNO) testbed is located in the CNIT PNTLab and SSSA TeCIP Institute premises in Pisa, Italy. The testbed is jointly managed by CNIT and SSSA and features access, metro, and core networks interconnecting emulated data centers numerous participations to international research projects and academic and industrial collaborations.

The testbed has gained a solid reputation with both industrial partners and internationally recognized research groups. In fact, it has been developed and expanded thanks to research grants from leading companies like Ericsson, Rete Ferroviaria Italiana, Trenitalia, Schindler group, Telecom Italia and many others. In addition, it has served as an experimental platform, for more than twenty years, in major EU-sponsored projects on optical networking during FP6, FP7, and H2020. Several acquired national and regional projects complete the testbed funding landscape.

The ARNO testbed was born as a testbed dedicated to optical networking, including few optical nodes and IP routers as tributary overlay network, to realize an IP-over-(Dense) Wavelength Division Multiplexing (IP-over-(D)WDM) environment. Then, thanks to the participation in several projects involving other segments of the telecommunication networks, it was expanded to support other network scenarios, including metro, cloud/edge segments and next generation RANs... At the packet level, Software Defined Networking (SDN)-enabled devices such as OpenFlow switches and controllers were added, up to the recent evolution to SDN data plane programmability, opening the door to programmable devices (P4 switches, P4 Smart NICs, Data Processing Units -DPU). Recently, the access segment was covered with Wi-Fi 6, Long Term Evolution (LTE) and 5G Proof of Concept (PoC) infrastructures and with different type of UEs, such as drones, autonomous mobile robots, rovers. Finally, the network softwarization and the advent of cloud computing has pushed the testbed - to include IT infrastructures, spanning from computing servers dedicated to virtualization (virtual machines -VM)) up to small data center infrastructure emulation, including two racks of servers of different computing capabilities, including GPUs for running AI-based algorithms. Currently, the testbed is hosted in two adjacent lab rooms as shown in Figure 9.

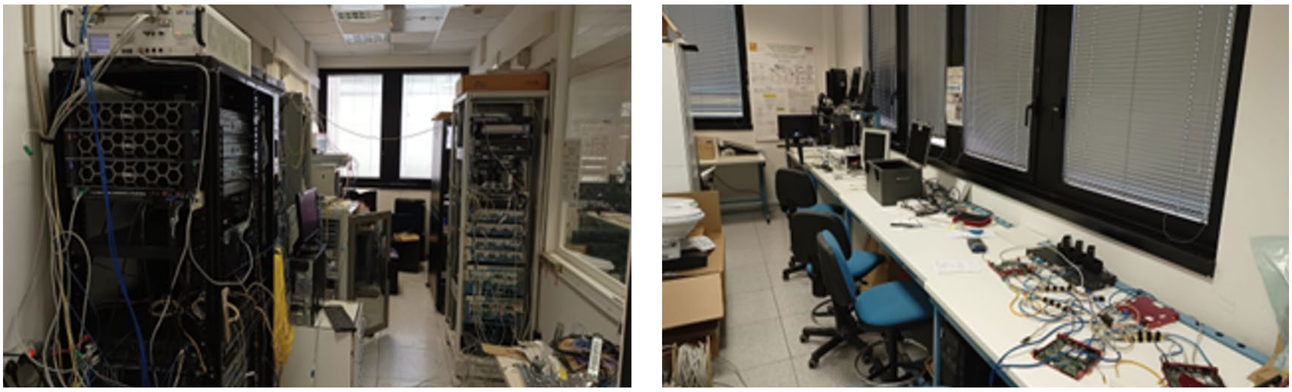

FIGURE 9. ARNO TESTBED: IT+TRANSPORT NETWORK SEGMENTS (LEFT), ACCESS AND MOBILITY SEGMENTS (RIGHT)

Currently, the testbed includes the following Telco+IT segments:

1. Long-haul transport optical networks
2. Metro flexgrid disaggregated optical networks
3. Aggregation (routers, OFswitches)
4. Access (PON, 5G RAN)
5. Cloud/edge nodes and interconnects
6. Wireless mobile terminals

The main goal of the testbed is the experimental validation of the concept, ideas and research topics that are studied and proposed by CNIT and SSSA researchers in the context of the different research projects, resorting to the state-of-the-art of the technology and the telecommunication/IT platforms. In addition, the historical mission of this testbed is to provide a wider interconnection to other distributed testbed to realize large-scale sandboxes. This goal was realized in many European projects, in which the ARNO testbed was interconnected with other testbeds by means of different connectivity, tunnelling and routing protocols, such as IP Security (IPsec), Generic Routing Encapsulation (GRE), OpenVPN and Border Gateway Protocol (BGP). This goal will be carried out also in the DESIRE6G development, integration, testing and validation activities. This integrated testbed will be open to partners in both two ways:

1. Remote access for integration, testing and validation of system components;
2. Peer-to-peer access for interconnectivity with other DESIRE6G testbeds, realizing federated testbeds.

In DESIRE6G, preliminary plans have considered to include a close participation to the testbed activities by specific partners. NVIDIA participates as a system integrator with special focus on the programmable switches and DPU resources available in the testbed. ACC participates as a system integrator of the 5G radio stack, complementing the resources already available in the testbed with their own software stack (RIC, CU and open source DUs). Finally, UvA participates as P4 data plane software integrator with special focus on in-band telemetry implementations.

#### 2.1.2.2. Overview of Integrated components

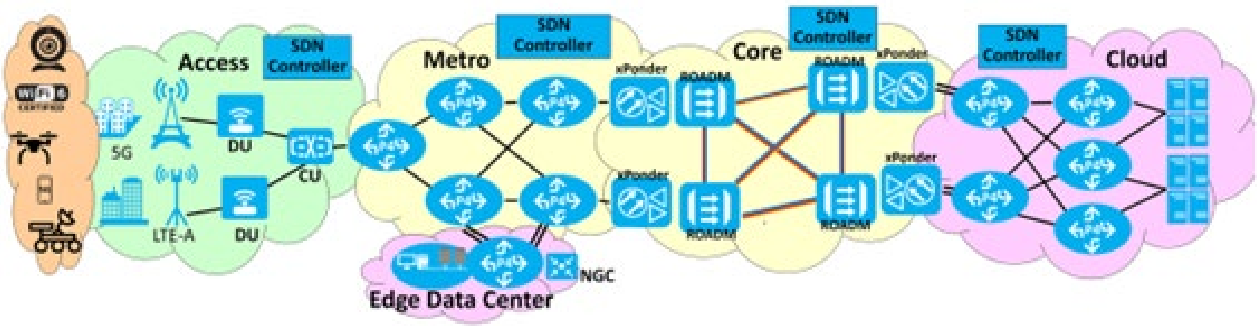


FIGURE 10. ARNO TESTBED SEGMENTS

Figure 10 shows the different segments covered by the ARNO testbed. Each segment is summarized below in terms of hardware and software facilities and components that are currently available. The detailed list and specifications of hardware equipment and software platforms are reported in Annexes.

- Long-Haul/metro Optical Transport: these two segments include the state-of-the-art of optical transport network technologies, including commercial ROADMs, disaggregated whiteboxes and optical line systems. In testbed configurations, the 5G mobile core network is typically hosted inside the metro segment.
- Cloud and edge data center: these segments reproduce cloud and edge computing facilities, including the related networking infrastructure. Cloud and edge include racks of servers of different flavours, spanning from high computing capacity (e.g., up to 128 cores) servers, AI-oriented machines (e.g., GPU and FPGA), network-oriented gateway machines (e.g., equipped with multiple and high-capacity network interfaces, or equipped with network accelerators). Far edge equipment (Jetson, ORIN) is included. The networking infrastructure comprises legacy switches and programmable switches (Tofino, Mellanox, bare metal servers) with different SDKs

(e.g., P4, eBPF) and SmartNICs (Alveo, Bluefield-1, Bluefield-2, Bluefield-2+GPU). Different topologies can be setup based on the desired testbed scenario, with data plane link capacity spanning from 1GbE up to 100GbE.

- Access and RAN: the segments reproduce the 4G/5G environment, including the radio hardware (antennas, RRU, dedicated FPGA, evaluation boards and terminals) and the radio stack software, including OpenAirInterface (OAI), FlexRIC and SRSran, currently under evaluation.
- Terminals and User Equipment's: small-factor UAV, droids, rovers, smart glasses, headsets for AR/VR (under purchasing), x-86 boards equipped with Wi-Fi and GPS, 5G dongles.
- Traffic generators and analyzers: hardware-based (Spirent, Viavi) with interfaces up to 400G, software-based (Cisco Trex).

**Testbed functional view**

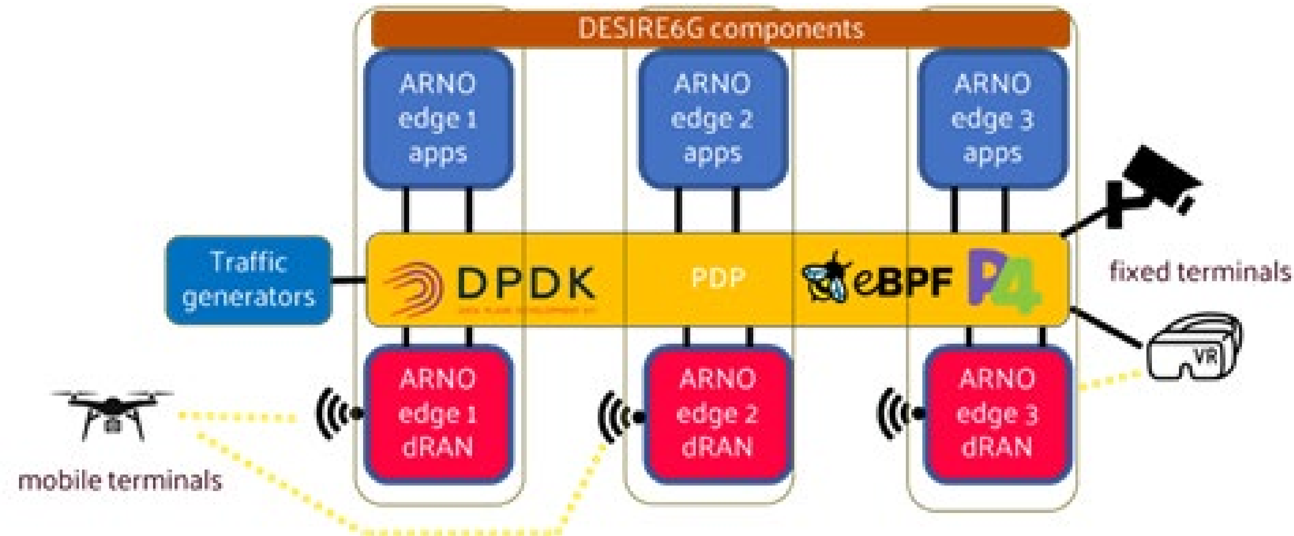


FIGURE 11. ARNO TESTBED: FUNCTIONAL VIEW

The functional view of the testbed for its usage in DESIRE6G is shown in Figure 11. The testbed has the capability to provide different edge nodes (e.g., 3 nodes emulating two edge and one cloud data center). The DESIRE6G components will be integrated inside each node on top of existing IT/network resources. Moreover, the vertical apps (e.g., the AR/VR chained application) will be hosted as service applications that may be instantiated by the orchestrator. The programmable data plane network, including P4, Data Plane Development Kit (DPDK) and extended Berkeley Packet Filter (eBPF) will be exploited for both intra-edge networking and inter-edge connectivity. Fixed terminals will be attested to given edge nodes, while mobile terminals will be connected using distributed RAN functions that may be either centralized in a single edge (e.g., Distributed Unit + Centralized Unit + Radio Access Network Intelligence Controller + Core Network, namely DU+CU+RIC+CN) or distributed across different edges, depending on the required 6G deployment scenario.

**Access to ARNO**

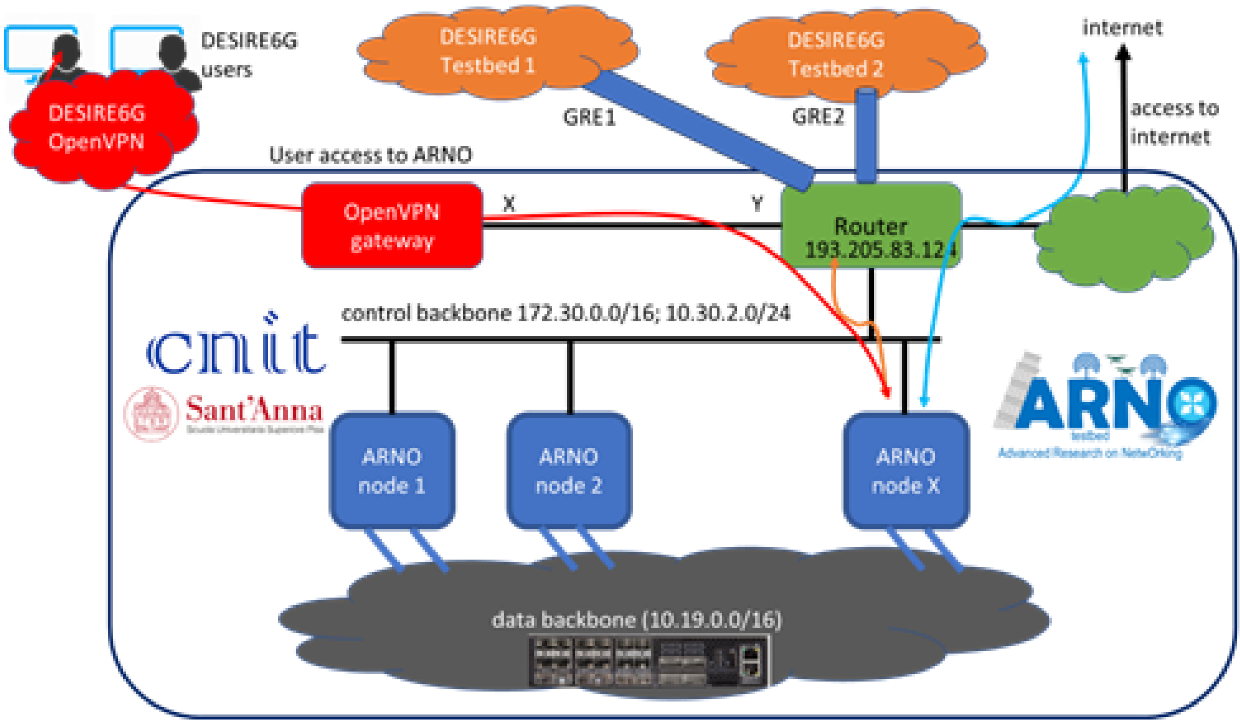


FIGURE 12. ACCESSING TO ARNO TESTBED RESOURCES

The ARNO testbed is available for different kinds of access by selected users and, upon agreements, by the DESIRE6G partners willing to utilize its resources for the project goals. Basically, two different types of access are available: user access and peer access. User access is given to third party users willing to access to specific portions of the testbed, for example a virtual machine, or a programmable switch. Peer access is given to remote software entities of third-party users to communicate directly with the ARNO resources, for example a remote Kubernetes orchestrator connected with the ARNO Kubernetes orchestrator or a dedicated ARNO cluster.

User access is permitted provided that the user is granted a personal profile with certificates, user and password, with the list of allowed resources. Typically, OpenVPN server is utilized at the ARNO testbed using the general configuration shown in Figure 12. The OpenVPN server is attested on a router, filtering the user access and the granted resources. Peer access is permitted through the configuration of a tunnelled connectivity to the resources of the remote testbed. Typically, the tunnelled connectivity is realized using a Generic Routing Encapsulation (GRE) [3], due to the easy deployment in gateways, routers and Linux machines. The tunnel is established between the local router and the remote gateway, enable to route two different L3 subnets between the two sites. This way, any instance, API, or protocol communication can be established between two entities belonging to the two routed subnets. In this case, the communication bypasses the OpenVPN gateway and is forwarded directly to the ARNO resource. Firewall rules are required to be exchanged, approved and configured by local IT departments (no specific rules at the ARNO side).

#### 2.1.2.3. Integration and validation timeline

The timeline of integration and validation activities in the ARNO testbed are reported in Table 3.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q4 – 2023 | • Identification of required hardware and software components |
| Q1 – 2024 | • Set up the basic network infrastructure<br>• Starting initial deployment of the AR/VR use case<br>• Conduct baseline experiments |
| Q2 – 2024 | • Starting integration of DESIRE6G components inside ARNO resources<br>• Conduct experiments and preliminary demonstrations (Y2 demo) |
| Q4 – 2024 | • Starting integration of final release components of DESIRE6G inside ARNO resources<br>• Starting final deployment of the AR/VR use case |
| Q3 – 2025 | • Validation of final components of DESIRE6G in ARNO |
| Q4 – 2025 | • Final Demo of integrated DESIRE6G solution for the AR/VR use case (Y3 demo) |

TABLE 3. ARNO TESTBED TIMELINE

## 2.2. Local testbeds

The DESIRE6G system components will be individually developed in the local testbeds and later in the project some of them will be integrated in the global testbeds as part of an end-to-end DESIRE 6G solution. Specifically, Section 2.2.1 presents the details of the OpenLab testbed where the user plane acceleration is developed. In Section 2.2.2 and Section 2.2.3 the details regarding the designed testbeds for developing the MAS and SMO are presented respectively. Section 2.2.4 presents the details of the Technology and Automation LAB where the ALTO component of the SMO will be developed. Section 2.2.5 presents the NPT-Budapest testbed used for developing the IML component of the DESIRE6G framework. Finally, in Section 2.2.6 the details of the local testbed that develops the DLT-based mutual attestation security mechanisms are presented.

### 2.2.1. OPENLAB

#### 2.2.1.1. Description

The OPENLAB setup in Antwerp involves a single server that hosts the Service and Management Orchestrator (SMO), the non-RT RIC, near RT RIC, CU-CP and the DU. It also consists of Benetel 550/650 Radio units and a Smart NIC (Netronome Agilio CX 2X25 GbE) that will host CU-UP functionality for accelerated user plane performance. The main goal of this test bed is to demonstrate user plane acceleration using the SmartNIC's multi-threaded flow processing cores. The cloud native deployment management system used in OPENLAB setup is Kubernetes. All programs, except CU-UP, are deployed as Kubernetes pods using Helm charts.

#### 2.2.1.2. Overview of integrated components

The test setup will consist of Ryzen or Dell servers with at least 3.5GHz processor, 16 core CPUs, 64 GB DDR4 RAM and 512 GB SSD Harddisk. The DUs and CUs will be installed in a Kubernetes cluster created on this server. The Benetel radios will be connected directly to the DU using a Fiber optic cable and SFP port.

Figure 13 shows the OPENLAB Blueprint at Accelleran premises including the srsRAN DU, Accelleran CU-CP, CU-UP, and the dRAX. It also shows near RT RIC and non-RT RIC with their relevant interfaces. 5GCore is deployed as a separate Virtual machine. The test setup consists of Benetel 650 RUs or B210 directly connected to the DU. A 5G NR mobile phone is connected to the test setup for test purpose.

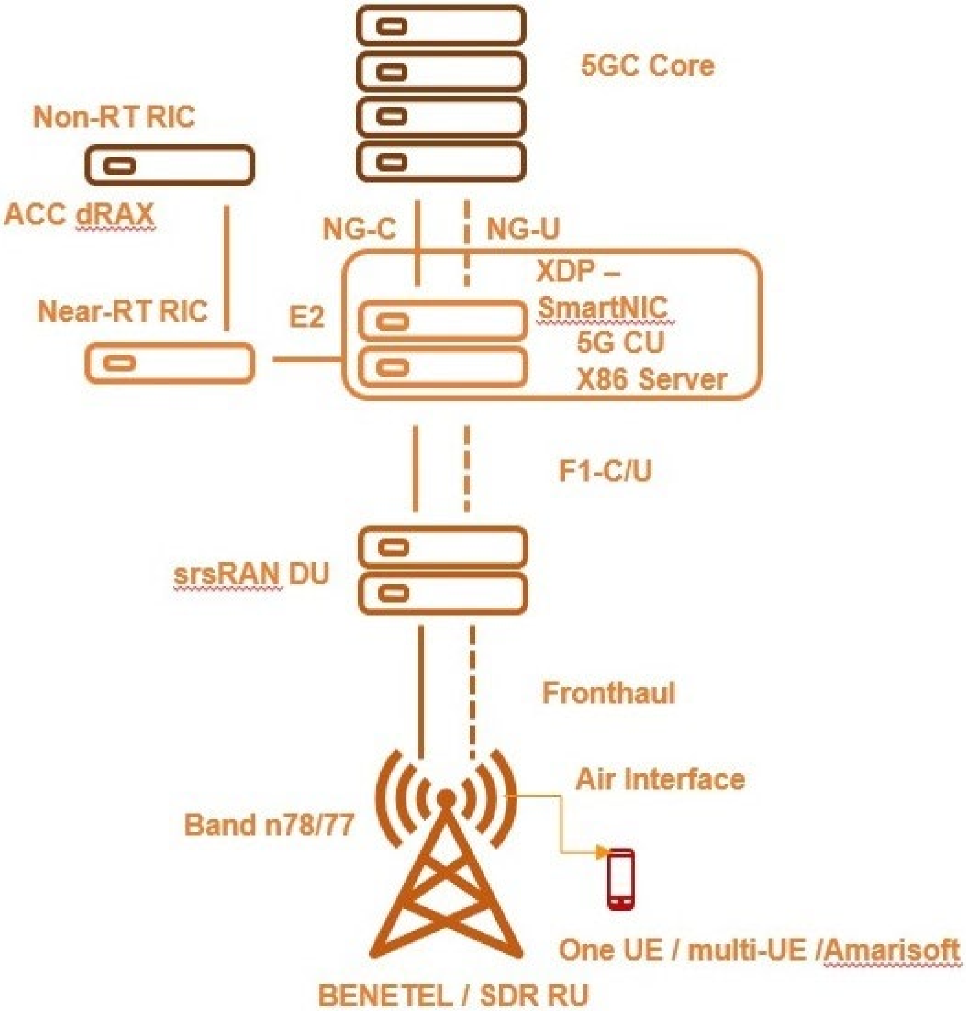


FIGURE 13. OPENLAB SETUP @ACCELLERAN PREMISES

#### 2.2.1.3. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q4 – 2023 | • Identification of required hardware and software components |
| Q1 – 2024 | • Set up the network infrastructure |
| Q2/Q3 – 2024 | • Start integration and testing of accelerated RAN components |

TABLE 4: OPENLAB TESTBED TIMELINE

### 2.2.2. Multi-Agent System (MAS) testbed

#### 2.2.2.1. Description

The Multi-Agent System (MAS) testbed is composed of a collection of compute, storage and network resources used by the GCO research group in different projects. It is based in the UPC Campus Nord in Barcelona, Spain. OpenStack is used as cloud resource manager providing a shared virtualized environment where workloads can be deployed on-demand. Additionally, a number of bare-metal machines are available for specific cases where virtualization cannot be used.

Researchers can deploy and run their experiments inside VMs deployed in the OpenStack cluster. The infrastructure includes an Uninterruptable Power Supply (UPS) protecting the devices from voltage spikes or power supply failure.

In this project, the testbed will be used for deploying the architecture supporting the deployment of agents in Multi-agent Systems (MAS) for controlling services. The ML function orchestrator (MLFO) and the MAS agents will be later integrated in global testbeds.

#### 2.2.2.2. Overview of Integrated Components

Figure 14 show shows the overall architecture deployed and the integration of the different components in GCO-UPC premises used to demonstrate the secure and optimized deployment of agents. Note that the deployed architecture is not the final one being defined in the project and it is only intended for integration and demonstration purposes, in principle for secure MAS deployment.

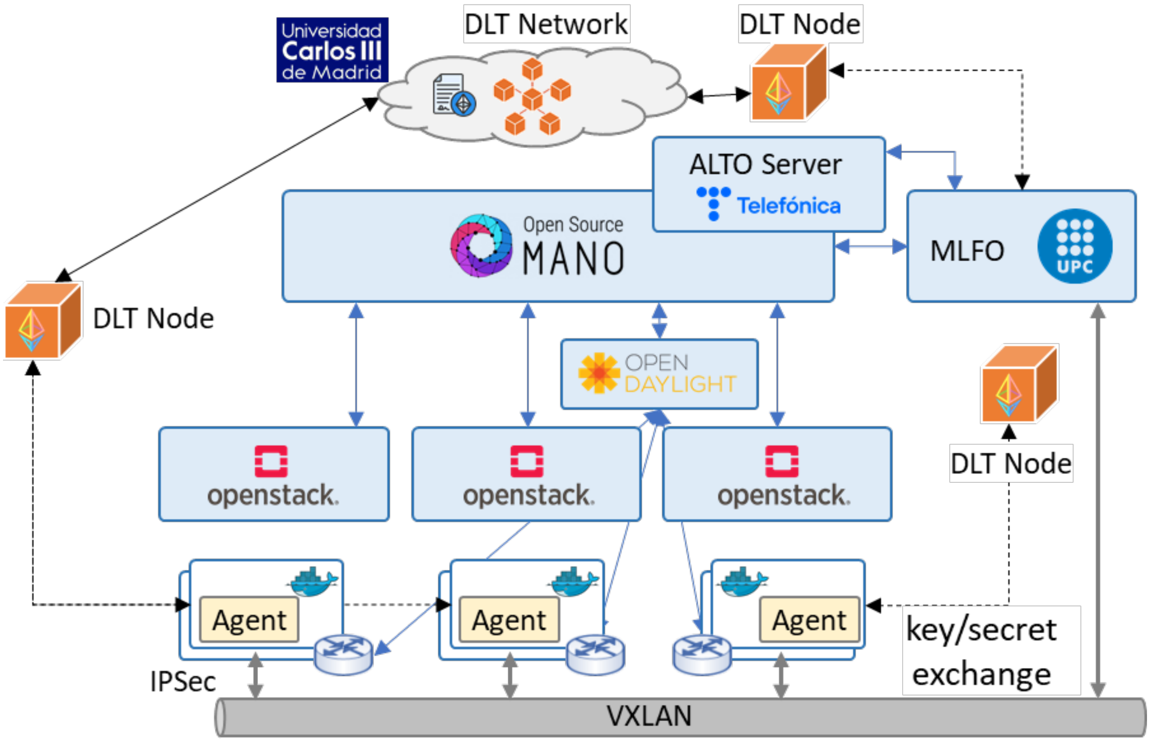


FIGURE 14. MAS AND MLFO COMPONENTS DEVELOPED AND INTEGRATED ON GCO-UPC PREMISES

The setup considers three different locations are considered, where each location includes computing resources, so a local virtualized infrastructure manager (OpenStack) is in charge of automating the deployment of VNFs. OpenDaylight (ODL) SDN controller is on top of the packet network and used to create the connectivity for the ML pipeline. Opensource MANO (OSM) is the orchestration system in charge of the deployment of NSs. A MLFO decides the locations where agents need to be deployed, how they need to be connected and coordinates their activity. Finally, a DLT infrastructure is also setup.

Figure 15 shows the components being developed and integrated in GCO-UPC premises.

The distributed system consists of several interconnected software components. Specifically, the telemetry agent, which is part of the distributed telemetry architecture, includes: i) a telemetry collector that periodically receives telemetry data from the P4 collector; and ii) a per-flow telemetry processor. The role of these telemetry components is to compute the required measurements and statistics that characterize the current QoS of the traffic flow.

The MAS agents and components, along with their interfaces, are implemented in Python 3.10.4 and run inside Docker containers on two separate VMs using Ubuntu Server 22.04 LTS.

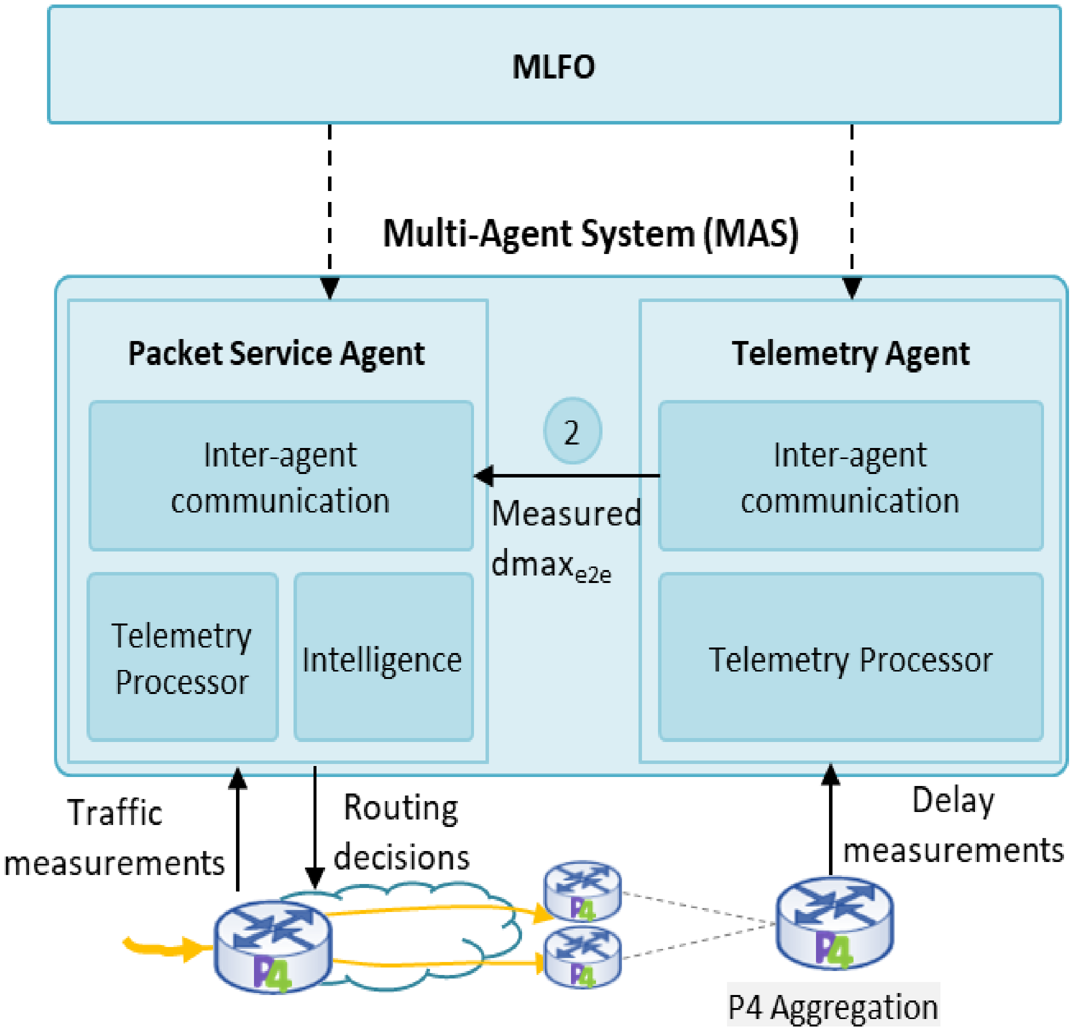


FIGURE 15. MAS AND MLFO COMPONENTS DEVELOPED AND INTEGRATED ON GCO-UPC PREMISES

Figure 16 shows the GCO-UPC data center, which consists of multiple servers in several racks and two programmable switches allowing the interconnection between the virtualized instances and the external network.

FIGURE 16. GCO-UPC DATA CENTER

- **GCO-UPC data center servers:** These servers offer CPU, memory and storage resources that are shared all along the OpenStack cluster. See server specifications in Table 37.
- **Programmable switches:** These switches are the backbone of the data center, tasked with the intelligent management and VLAN setup. In particular, two Cisco SG220-26P 26-Port Gigabit PoE Smart Plus Switches are used.
- **Router/Gateway:** Self-managed PfSense based router tasked with the internal routing of the data center and external traffic filtering by implementing a firewall. External access to the data center resources can be given on demand by using an OpenVPN server. Interconnection between testbeds could be also achieved by using tunnelling protocols such as GRE.

#### 2.2.2.3. Integration and validation timeline

Table 5 shows the integration and validation timeline of the Multi-Agent System testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | • Integration of DLT, MLFO, and MAS for MAS deployment and experimental analysis (Demo 3) |
| Q1 – 2024 | • MAS to control of the routing of the traffic of a service between two sites. Experimental analysis (Demo 4) |
| Q4 – 2024 | • MAS for load balancing of intra-site resources |
| Q3 – 2025 | • MAS and service reconfiguration |

TABLE 5. MULTI-AGENT SYSTEM TESTBED TIMELINE

### 2.2.3. Service and Management Orchestrator (SMO) testbed

The Service and Management Orchestrator (SMO) testbed is composed of a collection of compute, storage and network resources used by the NUBIS research & engineering staff in various commercial and research projects. It is based in our central office in Halkida, Greece. It is comprised of a variety of bare-metal servers, development boards and edge devices, all interconnected using 10Gbps and 1Gbps managed and unmanaged switches. Researchers can deploy and run their experiments in various setups, depending on the requirements of the experiment, with a particular focus on distributed workload management.

#### 2.2.3.1. Description

There are several distinct multi-arch (x86_64 and aarch64) k8s clusters, along with bare-metal machines that act as VM containers (plain KVM/libvirt). VMs deployed in these nodes are used as flexible and disposable cluster nodes when needed. The infrastructure includes an Uninterruptable Power Supply (UPS) protecting the devices from voltage spikes or power supply failure, along with a standby generator (genset). We have already provisioned for 2 new servers that will act as VM containers to be able to deploy distributed workloads as microservices. In this project, the NUBIS testbed will be used for developing, deploying and validating the SMO architecture. Additionally, it will ensure proper API support for external components to the SMO (e.g., Infrastructure Management Layer (IML), Software Defined Network controller (SDNc), etc.).

Figure 17 and Figure 18 show the testbed capacity, network topology and architecture.

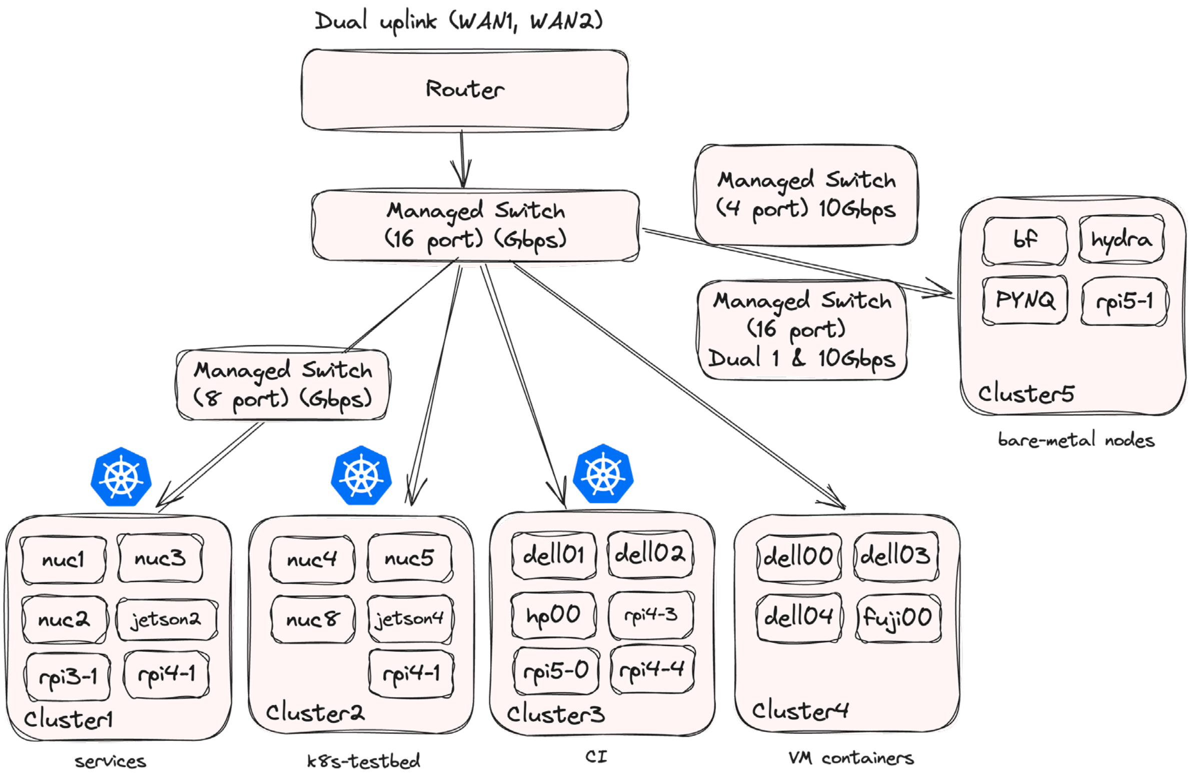


FIGURE 17. OVERVIEW OF SERVICE AND MANAGEMENT ORCHESTRATOR TESTBED

FIGURE 18. NUBIS DATA ROOM

### 2.2.3.2. Overview of integrated components

Figure 19 shows how the SMO components will be deployed on this testbed, as distinct micro-services.

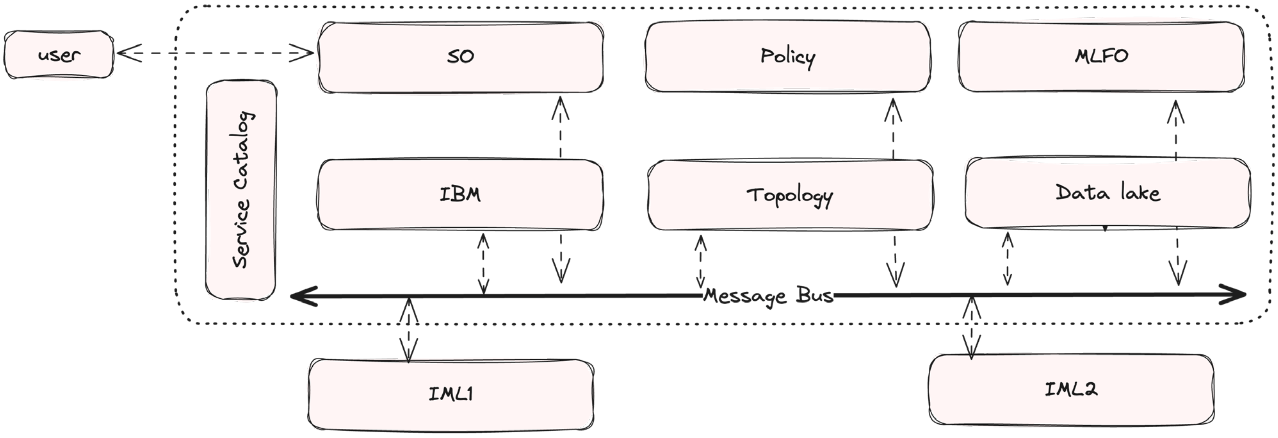


FIGURE 19. DESIRE6G SMO COMPONENTS DEPLOYED AS SEPARATE MICROSERVICES

#### 2.2.3.3. Integration and validation timeline

Table 6 presents the schedule of integration and validation tasks within the Service and Management Orchestrator testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | • Initial implementation of SMO components (Topology, Data lake & SO) – placeholders for individual service logic (policy, IBM) |
| Q1 – 2024 | • Integration of all components to a simple working example of service instantiation |
| Q3 – 2024 | • Local and Edge Site Telemetry integration with the SMO & Data lake |
| Q4 – 2024 | • Enhancement with advanced functionality for SMO components. Addition of the MLFO and MAS instantiation on IML sites |
| Q3 – 2025 | • End-to-end SMO functionality - user to IML site service and application instantiation |

TABLE 6. SERVICE AND MANAGEMENT ORCHESTRATOR TESTBED TIMELINE

### 2.2.4. Technology & Automation LAB

#### 2.2.4.1. Description

Telefónica has its own laboratory located in Madrid, named the Technology & Automation LAB. In this facility, TID defines, tests, experiments, and evolves cutting-edge technologies for the development of next-generation networks, featuring autonomous architectures supported by artificial intelligence. TID have an end-to-end service infrastructure and complete equipment to manually and automatically test TID products and prototypes across various devices. The TID laboratory serves as an open workspace to test solutions by simulating Telefónica environments, with the aim of collaboratively constructing new products and services alongside TID partners.

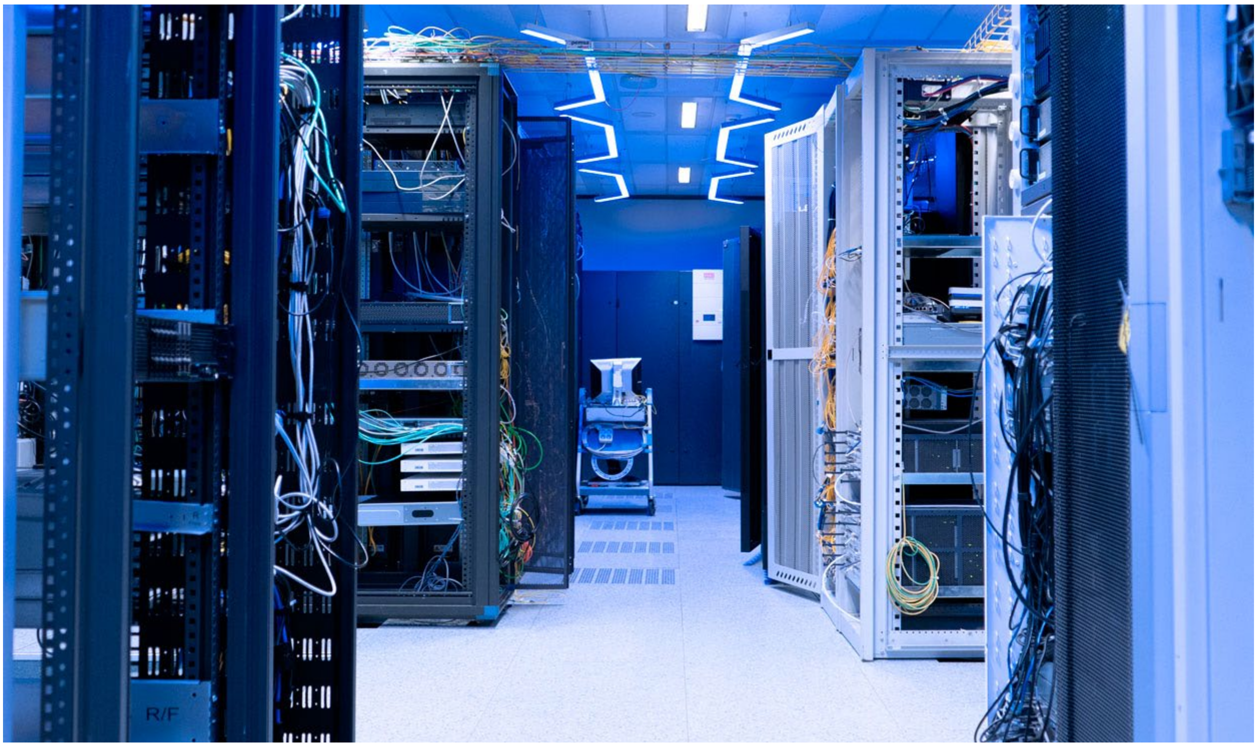

FIGURE 20. TECHNOLOGY & AUTOMATION LAB

In the project, the main goals of the Technology & Automation LAB testbed are:

- Functional validation and integration of IETF DetNet capabilities between DESIRE6G sites.
- Functional validation and integration of translator of Intents from SMO towards the DESIRE6G Transport Network.
- Functional validation and integration of translator of ALTO based Topology Exposure Server.

#### 2.2.4.2. Overview of integrated components

The Technology & Automation LAB boasts a diverse array of resources, including servers, traffic generators (such as the Spirent N12), switches, antennas, etc. In this section, we showcase the most notable resources for DESIRE6G.

On one hand, we have **TSN switches** that will serve us in validating and integrating IETF DetNet capabilities between DESIRE6G sites. For detailed specifications, please refer to Table 38 in the Annexes.

On the other hand, the remaining resources we are going to present belong to the programmable data plane. We categorize these resources into three parts: HW switches, SW switches, and other resources.

HW switches

We have 2x CSP-7550 server-switch for P4 (TNA) connected to a PowerEdge R730 server, from which the various P4 programs will be compiled and executed.

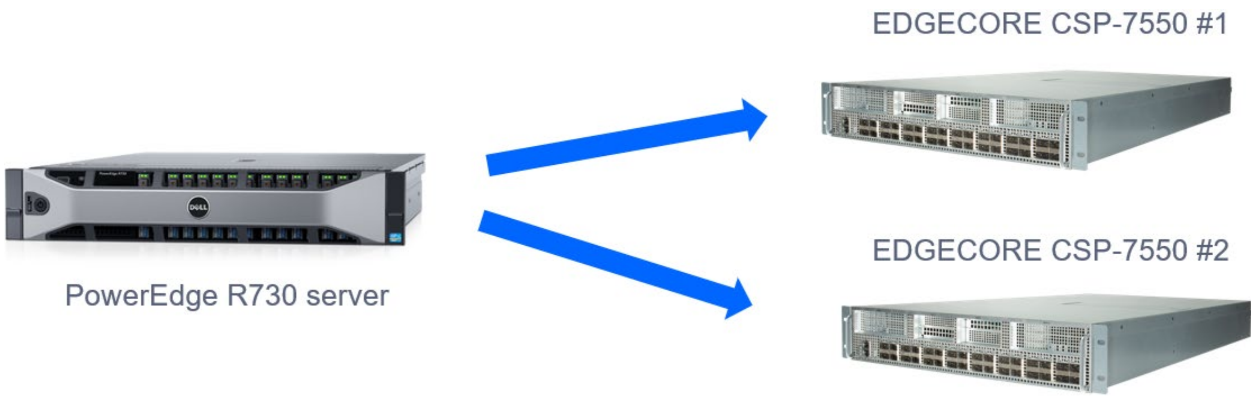


FIGURE 21. TECHNOLOGY & AUTOMATION LAB: HARDWARE SWITCHES

For detailed specifications of **PowerEdge R730** server features and **Edgecore CSP-7550 switches**, please refer to Table 38 in the Annexes.

SW switches

Additionally, from virtual machines where we can work with **BMv2 switches**, we have **P4+DPDK** solutions deployed in various Docker containers. In this way, we are able to use a traffic generator connected through this solution to perform various measurements.

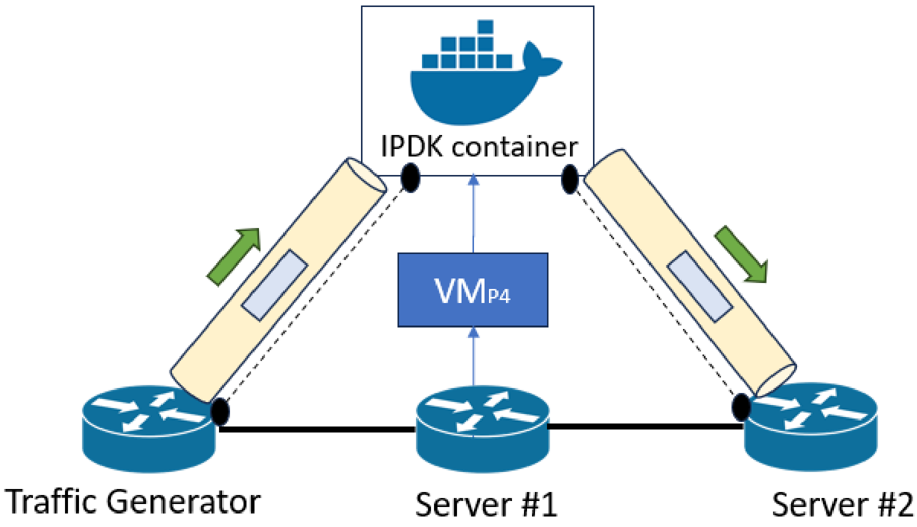


FIGURE 22. TECHNOLOGY & AUTOMATION LAB: P4+DPDK SOLUTION

Other resources

We have recently acquired 2x **NetFPGA Sume** (Xilinx Virtex-7 690T) and 2x **SmartNICs Netronome Agilio CX 2x10 GbE**, which will be available for use in the project.



FIGURE 23. TECHNOLOGY & AUTOMATION LAB: OTHER RESOURCES

### 2.2.4.3. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q2 – 2024 | • Integration of hardware and software components |
| Q3 – 2024 | • Set up the basic network infrastructure |
| Q4 – 2025 | • Integration and validation of the components<br>• Test in Technology & Automation LAB |

TABLE 7. TECHNOLOGY & AUTOMATION LAB TESTBED TIMELINE

## 2.2.5. NPT-Budapest: Network Programmability testbed

### 2.2.5.1. Description

Two very similar testbed infrastructures were built at ERI-HU and ELTE to support the evaluation and validation of data plane algorithms and methods. Both testbeds are in Budapest few hundred meters from each other (but connected to different external networks) and rely on several servers with different hardware accelerators, like smartNIC, DPU, FPGA and Tofino ASIC. We started building them around five years ago with the purpose of enabling the performance evaluation of our 5G-related packet processing and QoS solutions. During the years, the testbeds were extended with different hardware accelerators and low-cost programmable router-boards. ELTE and ERI-HU have used these testbeds in a number of recent publications in the past few years.

The testbed will be used for the development and validation of components (e.g., IML, infraNFs) needed for the operation of a single DESIRE6G site. After validation these infrastructure components will be moved to the global testbeds (5TONIC and ARNO) as they are needed for the implementation of predefined demos.

### 2.2.5.2. Overview of integrated components

The network setup of the two testbeds is depicted in Figure 24 . The servers in both testbeds can be reached via SSH. The access is currently limited to collaborating partners, based on agreed workplans.

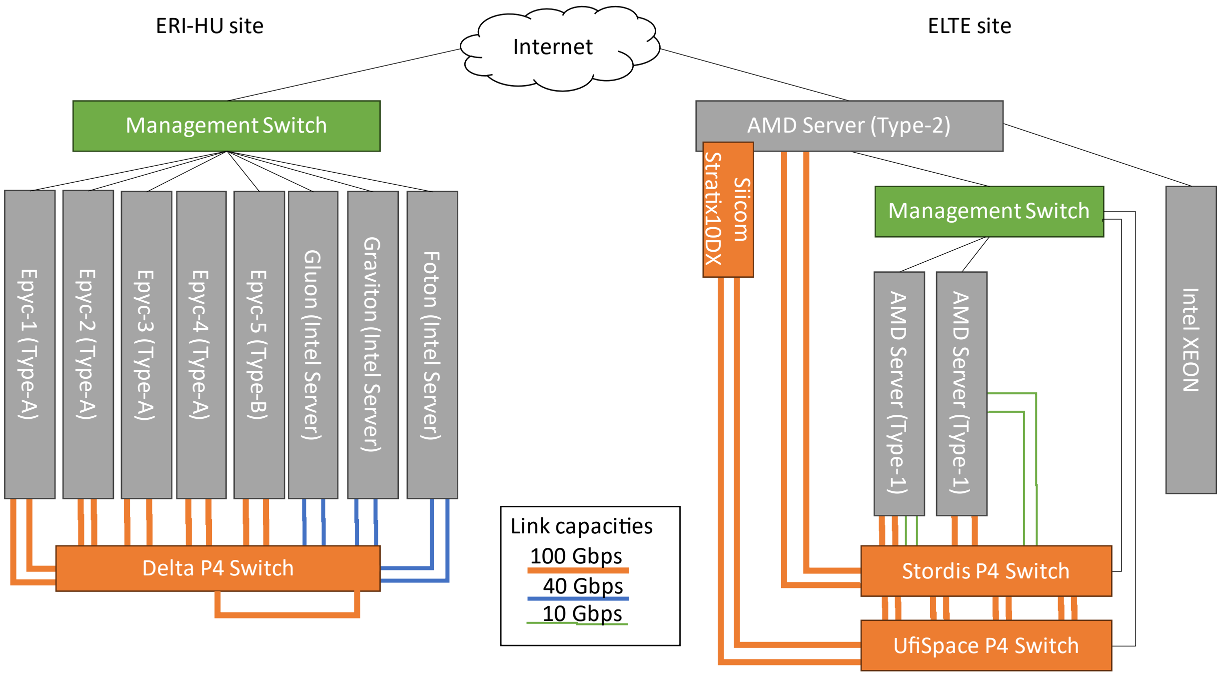


FIGURE 24. THE NETWORK SETUP OF THE TWO TESTBED INFRASTRUCTURES WITH DIFFERENT LINK CAPACITIES[3]

The detailed descriptions of different hardware components operated by ERI-HU and ELTE are described in Table 39 in the Annexes.

### 2.2.5.3. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | • Local K8S deployment with IML |
| Q1 – 2024 | • IML can deploy network services along SW and HW data planes, NF Routing operates, initial integration with SMO, initial integration of infraNFs |
| Q3 – 2024 | • Integration of the IML traffic management feature in the Digital Twin PoC |
| Q4 – 2024 | • IML can optimize load distribution in run-time, initial integration with MAS |
| Q3 – 2025 | • Multi-site IML deployment – e.g., between the two sites of Budapest testbeds or with UvA's OpenLab |

TABLE 8. BUDAPEST TESTBEDS FOR NETWORK PROGRAMMABILITY TIMELINE

[3] The management network uses 1 GbE links.

### 2.2.6. TSS DLT-back mutual attestation and runtime monitoring testbed

#### 2.2.6.1. Description

TSS DLT-back mutual attestation and runtime monitoring testbed is a software development platform, made of personal machines and a centralized code storage repository at an external cloud vendor.

TSS is currently engaged into two developments: the DLT-back mutual attestation method (as part of WP3) and network function hardening and deep performance monitoring (as part of WP4)

TSS's DLT backed mutual attestation is a novel concept leveraging the strength of DLT and its flexibility by encoding a smart contract which orchestrates the proving and verification functions over several agents. The goal of the testbed is the development of a representative implementation (e.g., DLT nodes, the smart contract, proving and verification primitives appended on agents) to progress the on-going developments, leading to a PoC and fine tuning of the roles, actions of the stakeholders, inter-stakeholder interactions.

TSS's deep performance monitoring is a novel concept of extracting telemetry related to a software intrinsic performance ratio deeply rooted into the software control flow, producing time series. This is exemplified with a network function and Intel's DPDK offering a second and source of telemetry, useful for the calibration of our method. The objective of the testbed is to implement a realistic workload benchmark on a performance constrained network function to calibrate and fine tune our control flow derived metadata.

The DLT-back mutual attestation modifies the MAS agent through TSS's Security as a Service (SECaaS) server which injects the required code primitives for measuring and verifying the agent signatures as well as storing the associated signatures.

#### 2.2.6.2. Overview of integrated components

Below, we provide detailed descriptions of each software component that comprises the TSS DLT-back mutual attestation and runtime monitoring testbed.

- **TSS proving and verifying primitives:** Remote attestation acts as a digital trust handshake, crucial for ensuring the integrity and authenticity of network devices or agents. The process is centred around two key functions:
    - **Proving Primitive:** This component is responsible for demonstrating that the current state of an agent is secure and unaltered. It achieves this by generating a cryptographic hash of its software and configuration, which is then signed with a private key. This approach ensures that the proof is accurate and resistant to tampering. Any changes in the system would result in a different hash, signalling a possible security breach.
    - **Verifying Primitive:** The verifying primitive's role is to authenticate the proof provided by the proving primitive. It does this by comparing the received hash or signature against a known, trusted value or set of values. This comparison is vital in determining whether the state of the proving primitive is intact and trustworthy.
- **DLT nodes:** Blockchain technology, in the form of Distributed Ledger Technology (DLT), particularly Ethereum, which stands as the most widely recognized blockchain platform. Ethereum offers a versatile platform capable of supporting various applications via the execution of smart contracts, which operate directly on the blockchain. We employed Go Ethereum[4] (Geth) to establish a private Blockchain network and record state of agents during the remote attestation workflow. Additionally, we have leveraged the Web3.py[5] library for seamless interaction with the Blockchain.
- **Smart contracts:** The implementation of smart contracts is interesting for ensuring that the states of all agents remain secure and trustworthy. This process maintains a continuous chain of trust, as every agent state is immutably recorded and can be reliably verified when a remote attestation is triggered. To compile and deploy smart contracts, we relied on node.js-Truffle, which simplified the development process and allowed us to efficiently integrate these contracts into our network.

[4] https://geth.ethereum.org/

[5] https://web3py.readthedocs.io/en/stable/

- **Docker:** Virtualization technology utilized for providing flexibility and efficiency in deploying and managing agents.
- **UPC's MAS agent:** We have further enhanced the UPC agents by introducing a python FastAPI, enabling seamless communication and interaction between agents. This API allows agents triggering the verifying and proving functions within our network.
- **PostgreSQL:** Relational database used by the SECaaS component to store measures and public key of agents provided by the measurer and the key generator.
- **L2fwd network function:** We have integrated probes into Ericsson's L2fwd network function to instrument the control flow, allowing us to closely monitor its operation and promptly detect any deviations or anomalies in its behaviour. This enhanced monitoring capability provides valuable insights into the performance and reliability of the network function, facilitating proactive measures to ensure its optimal functioning. We also have set up a (DPDK) environment on local virtual machines to enable monitoring that closely simulates a production environment.

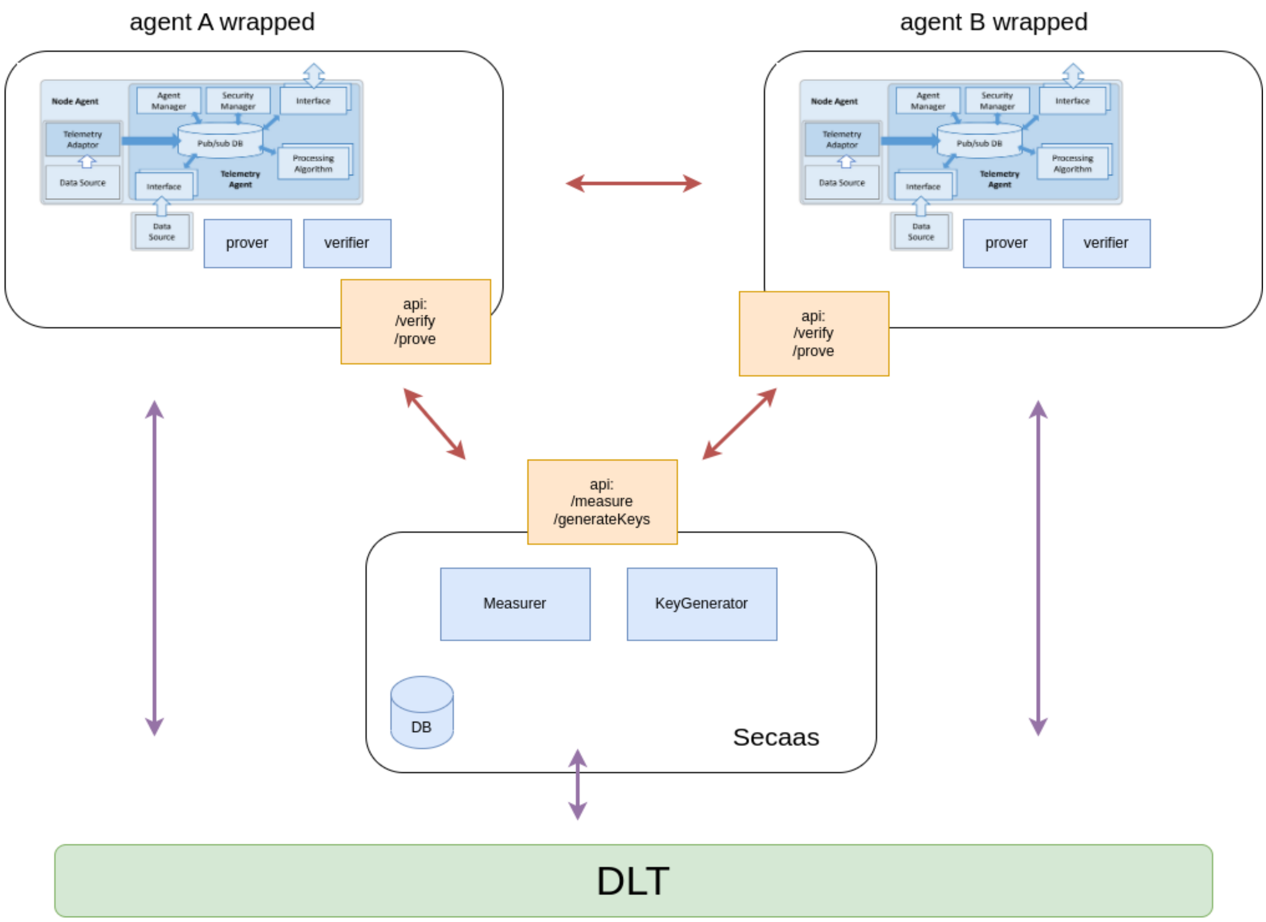


FIGURE 25. DLT-BACKED MUTUAL ATTESTATION TESTBED COMPONENTS

### 2.2.6.3. Integration and validation timeline

Table 9 shows the different integration steps to be taken at TSS side before code migration. The integration of the code will be done in the DLT MAS Security federated testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q3 – 2023 | • Elaboration of specifications and selection of DLT, smart contract |
| Q4 – 2023 | • Preliminary implementation of smart contract and DLT nodes<br>• Preliminary implementation of Intel DPDK for Preliminary implementation of trampoline code on a sample function.<br>• Preliminary L2fwd hardening (over several security profiles) delivered and tested at Ericsson Hungary |
| Q2 – 2024 | • UPC agent protection insertion of prove and verify functions<br>• Trampoline code insertion inside L2fwd<br>• Insertion of trampoline code into TSS's SECaaS automatic binary rewriting tool |
| Q4 – 2024 | • Code push into DLT MAS Security federated testbed and at Ericsson |
| Q4 – 2024 | • Integration and validation of the components and tests with UPC, UC3M and Ericsson |

TABLE 9. DLT-BACKED MUTUAL ATTESTATION TESTBED TIMELINE

## 2.3. Federated testbeds

The main goal of the federated testbeds is to connect diverse sites among DESIRE6G project partners, allowing resource sharing through remote sessions. Figure 26 provides a visual overview of the interconnected testbeds and their relationships. These testbeds can be classified into two main types based on their involvement. The first type, known as "Federation", enables remote interconnection of components for collaborative activities, while the second type primarily focuses on granting partners remote access to the testbed resources.

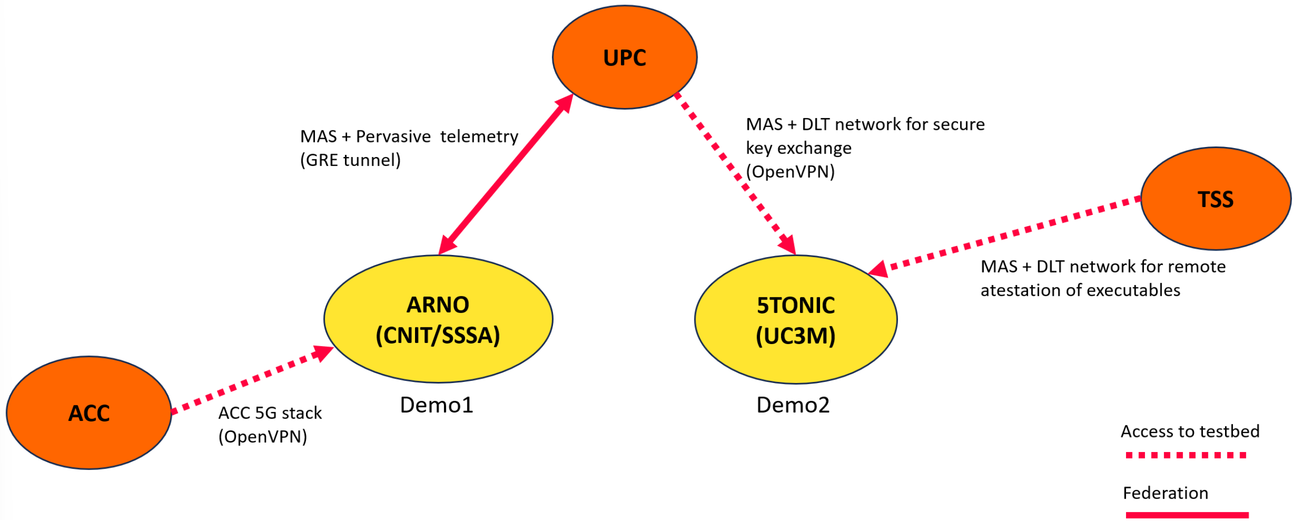


FIGURE 26. INTERCONNECTED TESTBEDS AND THEIR RELATIONSHIPS

## 2.3.1. Telemetry and Service Control

### 2.3.1.1. Description

The ARNO testbed and the UPC local testbed have been utilized jointly in past project to complement the resources and the modules developed for several different projects to exploit decentralized demonstrations. In DESIRE6G, the two premises will form a federated testbed sharing specific modules of the DESIRE6G architecture and provide preliminary demonstrations towards the final demonstration that will be entirely integrated in the ARNO testbed during the second part of the project. This federated testbed is a key asset because it allows to deploy the preliminary releases of the pervasive telemetry solutions. In fact, ARNO exploits In-band telemetry – equipped data plane network segments and telemetry collectors, while the UPC testbed hosts the different modules of the MAS. In this way, a preliminary test case and demonstration can be studied and deployed before the starting of the integration tasks, providing first results and a gap analysis for the integrated deployments of the second part of the project.

### 2.3.1.2. Overview of the federated testbeds

The experimental setup for preliminary demonstrations involves deploying the federated testbed as shown in Figure 27. At the CNIT/SSSA ARNO testbed, a packet-optical data plane network is deployed, while the distributed MAS finds its deployment at UPC premises. The connectivity between the two sites is assured by the configuration of a dedicated GRE tunnel between the two testbed gateways, utilizing the L3 GARR/GEANT internet routing between university network sites. To create the packet network, five native in-kernel P4-programmable software switches (P4-NIKSS) are deployed and interconnected using 10G interfaces. These P4 switches operate on a separate server each, leveraging an Intel(R) Xeon(R) Gold 6238R CPU @2.20GHz with 256GB RAM to maximize performance. Additionally, an extra P4-based software process serves as a telemetry data collector, receiving INT data from the P4 switches.

To replicate network congestion and selectively introduce delays in selected traffic flow paths, a Juniper M10i router equipped with 1G interfaces connects to switch R1. Specific flows are directed through the Juniper router before route switching, creating varied congestion scenarios. Concerning the optical layer, an optical bypass between R1 and R5 is established using a pre-programmed packet-optical whitebox that utilizes 10G and 100G optical pluggables. These pluggables connect to an Optical Line System (OLS) via an arrayed waveguide gratings multiplexer. The OLS comprises three 80km fibre spans and line optical amplifiers. The local enforcement and pre-establishment of the optical connection between the pluggables occur through OpenConfig-enabled SDN agents co-located with the SONiC node operating system. Lastly, for packet generation, a Spirent SPT N4U equipped with 10G ports serves as the packet generator.

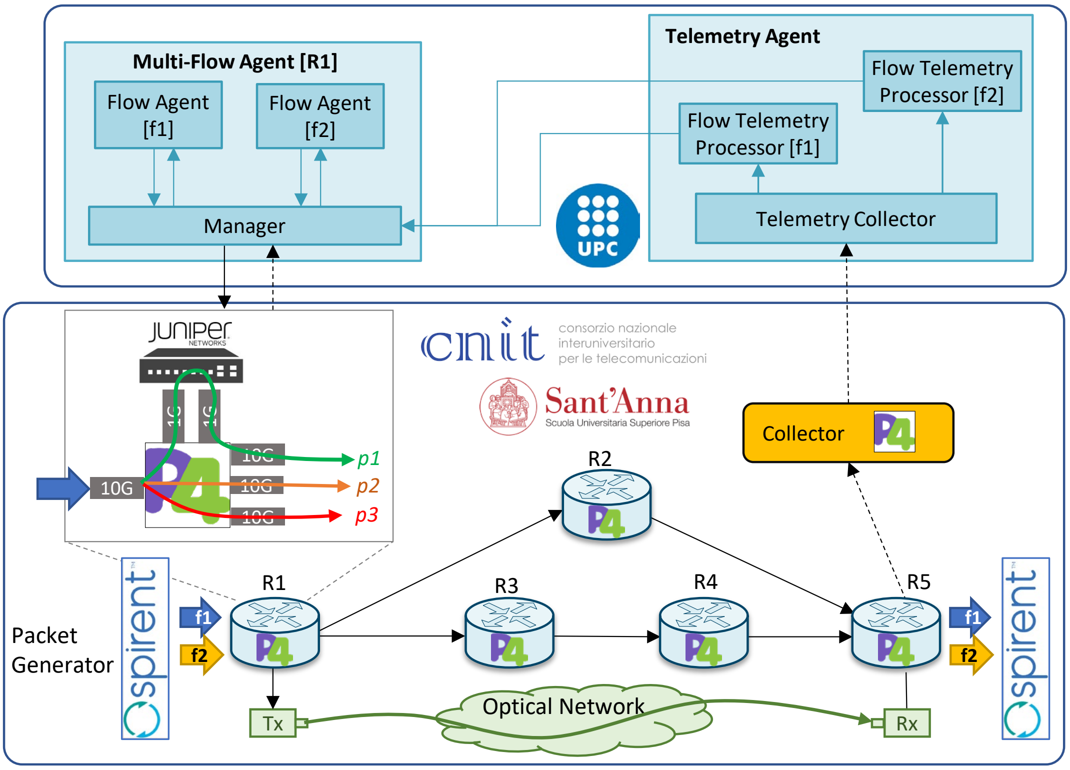


FIGURE 27. TELEMETRY AND SERVICE CONTROL FEDERATED TESTBED

The distributed system comprises interconnected software components, notably the telemetry agent within the distributed telemetry architecture. It consists of a telemetry collector receiving data from the P4 collector and a per-flow telemetry processor. These components analyse INT telemetry data to measure and characterize the Quality of Service (QoS) of traffic flows. The system relies on REST API interfaces between agents and P4 systems. The MAS agents and components, along with their interfaces, are implemented in Python 3.10.4 and run inside Docker containers on two separate VMs using Ubuntu Server 22.04 LTS.

#### 2.3.1.3. Integration and validation timeline

Table 10 shows the integration and validation timeline of the Telemetry and Service Control federated testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | • Telemetry data collection and processing |
| Q1 – 2024 | • Preliminary control of the routing of the traffic of a service between two sites<br>• Demo 4 |
| Q4 – 2024 | • Load balancing of intra-site resources |
| Q3 – 2025 | • Service reconfiguration |

TABLE 10. TELEMETRY AND SERVICE CONTROL FEDERATED TESTBED TIMELINE

### 2.3.2. Open disaggregated RAN

#### 2.3.2.1. Description

This testbed, which is federated into the DESIRE6G testbeds, provides an Open 5G SNPN which consists of a 3GPP conformant Standalone Private 5G network (extensible towards 6G), with the O-RAN Alliance aligned extensions of Open RAN (such as the RIC RAN intelligent Controller, additional interfaces like E2/O1/… and low-Level Split RU/DU Open fronthaul), providing a cloud-native, disaggregated and software-defined Open RAN based solution, which integrates with an open source 5GC and SMO, together with both commercial 5G-NR UE equipment and UE emulation test equipment.

This testbed also supports hardware acceleration of some of the Open RAN network functions, to validate and assess the programmable hardware targets such as SmartNICs.

#### 2.3.2.2. Overview of the federated testbeds

The Accelleran testbed shown below integrates the Accelleran dRAX commercial product line (consisting of the RIC, CU-CP, CU-UP and SMO dashboard) with multi-vendor DU support (both open-source srsRAN and commercial 3rd part L1/L2 solutions) and flexible RU selection (including B210/N310 SDR, and Benetel indoor 550 and outdoor 650 units and many others). Components are integrated together using standard 3GPP & O-RAN interfaces. Once the test bed is installed, users can interact with testbed using RIC dashboard GUI or northbound configuration API. While most RAN network functions are supplied as cloud-native software based, running on COTS x86/ARM server, the CU-UP on this test bed supports hardware acceleration via programmable smartNIC to test the hardware accelerated aspects of the DESIRE6G architecture.

Figure 28 shows the design of the federated testbed with the representative HW and SW components and the network in between.

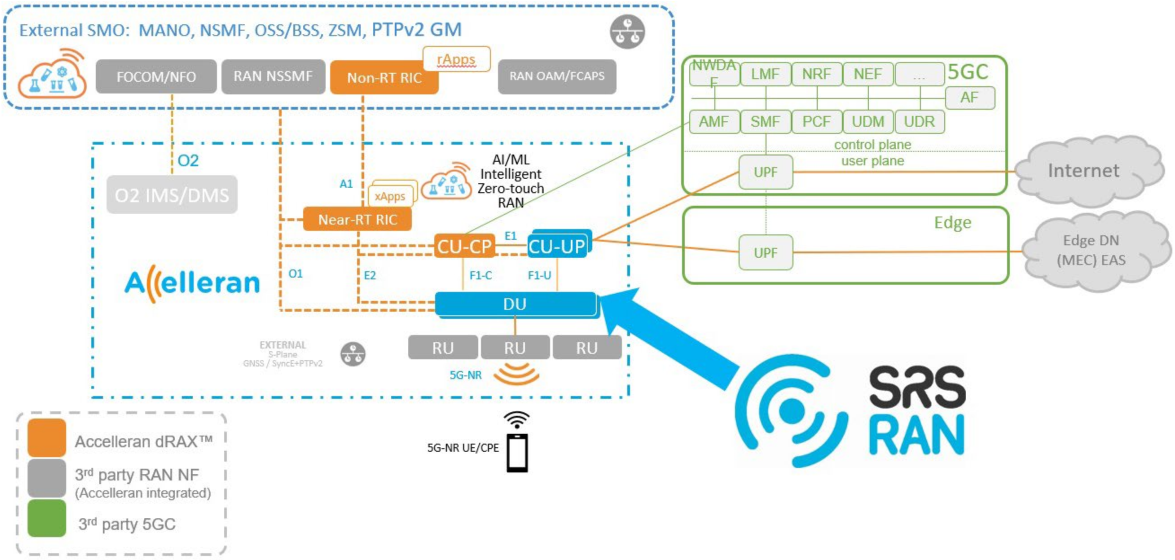


FIGURE 28. ARCHITECTURAL OVERVIEW OF THE OPEN 5G SNPN

The integrated & validated Open 5G SNPN is developed, integrated, tested and validated in the Accelleran labs, aligned with their commercial solution offering and product roadmap. Deployment instances of this testbed are available for replication in CNIT and UvA testbeds, to support their development activities as a DESIRE6G project instance in Accelleran, to develop and test the project results.

#### 2.3.2.3. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q3 – 2023 | • Accelleran system release based on 3GPP Release-15 & O-RAN extensions (such as RIC & E2SM) |
| Q1 – 2024 | • Deployment of mirrored instance to CNIT/SSSA<br>• Ongoing development of ACC product line to support new 3GPP Release-16 COTS UEs, fuller srsRAN integration and DESIRE6G project specific functionality |
| Q2 – 2024 | • Deployment of mirrored instance to UvA, supporting CU-UP smartNIC hardware acceleration initial research |
| Q4 – 2024 | • System release snapshot supporting D6G research extended capabilities |
| Q2 – 2025 | • Finalization of D6G functionality for integration in ACC and upgrades in CNIT/UvA |
| Q3 – 2025 | • Support for final D6G demonstrators |

TABLE 11. OPEN DISAGGREGATED RAN FEDERATED TESTBED TIMELINE

### 2.3.3. DLT MAS Security

#### 2.3.3.1. Description

The DLT MAS Security testbed is a collaborative effort between UC3M, UPC and TSS that combines hardware and software components to enhance the security of the MAS in DESIRE6G architecture. The main goal is to leverage a Distributed Ledger Technology (DLT) infrastructure to enable:

- Secure inter-agent communications through the exchange of VXLAN secret keys.
- Remote attestation of agents using Binary Hardening Tool (BHT).

### 2.3.3.2. Overview of the federated testbeds

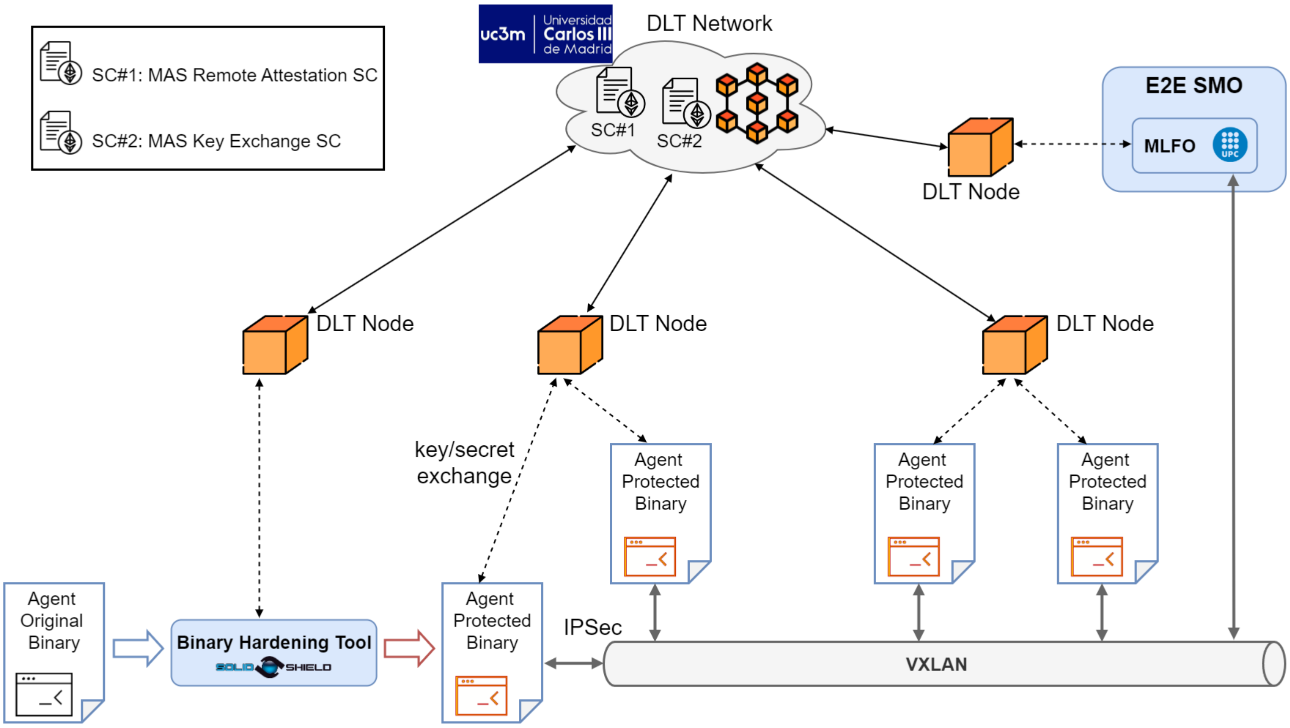


FIGURE 29. DLT MAS SECURITY FEDERATED TESTBED

As depicted in Figure 29, a DLT network is configured with two different Smart Contracts (SCs): **MAS Key Exchange SC** and **MAS Remote Attestation SC**. The first contract focuses on facilitating a secure exchange of VXLAN keys/secrets for inter-agent communications, while the second contract is specifically designed to handle remote attestation procedures for agents.

The process begins with the MLFO component of the SMO, which is responsible for determining agent deployment locations and their interconnections. Subsequently, it creates DLT accounts for these agents and registers their addresses with credentials to enable interaction with the **MAS Key Exchange SC**. Agents gain access to this smart contract via their local DLT node. The MLFO then generates a random key which is used to initialize IPsec for secure communication over the configured VXLAN and pushes the key to the smart contract.

The setup runs on different virtual machines (VMs) located at the UPC premises in Barcelona, Spain. The agents and MLFO are implemented in Python and run inside Docker containers. Finally, a DLT network, consisting of multiple container-based nodes, is established on the UC3M testbed.

The integration of TSS components into the federated testbed is currently in progress, with the aim of establishing the remote attestation mechanism before the VXLAN key exchange process begins. This integration will enhance the overall security of the MAS by ensuring that only trusted and verified agents can communicate with each other. Further details will be provided in D5.2.

#### 2.3.3.3. Integration and validation timeline

Table 12 illustrates the timeline for integrating and validating the federated testbed.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | • Integration of DLT, MLFO, and MAS for MAS deployment (Demo 3) |
| Q4 – 2024 | • Integration of BHT for mutual remote attestation of agents |
| Q1 – 2025 | • Set up the global DLT infrastructure<br>• API development for Blockchain/Smart Contract interactions |
| Q3 – 2025 | • Testing of APIs functionalities<br>• Verification tests |

TABLE 12. DLT MAS SECURITY FEDERATED TESTBED TIMELINE

# 3. Proof-of-concepts

This section outlines the preliminary plans for the PoCs associated with the use cases defined in WP1. An overview of each PoC is provided, including its alignment with project objectives, its relation with use cases, Software and Hardware configuration, ways to measure performance metrics, interim results, as well as the required technological components developed in WP2 [4] and WP3 [1]. Additionally, it describes the demonstration scenario and execution plan for the first half of the project.

## 3.1. PoC#1: VR/AR with perceived zero latency

### 3.1.1. Description

This PoC describes an Augmented reality application offering a perceived zero latency immersive experience to a human user, equipped with a proper headset for remote surveillance. This application will be designed and developed in parallel with the DESIRE6G project.

The AR app is composed of several functional modules that are placed at different locations, namely, the source node, edge computing nodes, and the headset. The source node is represented by a drone able to bring a camera (or cameras) and granting a superior point of view on the area of interest. The source node provides images streaming to more powerful computing resources, located into the Edge Node.

The Edge Node includes SmartNIC-powered stations, providing computing capabilities able to offload the other nodes of the network, to offload time-sensitive processing and grant perceived zero-latency to the application. The Edge Node is responsible for image processing and enhancement before sending processed information to the User Node.

The User Node is provided with an VR headset worn by the operator: the operator is able to communicate with the camera system of the Source Node through a forward/backward channel. The headset reproduces the user perspective augmented with revealed target objects that are hidden from the user's view (e.g., a car behind a truck, or a human body in a forest).

### 3.1.2. Relation with project objectives

Table 13 shows the relation of the VR/AR PoC with the project objectives.

| Project objectives | How will this PoC address the project objectives |
|---|---|
| **Obj. 1: Design a functional architecture for 6G mobile networks to support the next generation of URLLC use cases** | |
| R&D Topic 1: real-time and data processing must be granted to allow perceived zero latency for the VR/AR PoC. | The VR/AR PoC will validate the functional architecture designed for 6G mobile networks by demonstrating its efficiency in granting perceived zero latency and real-time view control. |
| Verification: validate DESIRE6G components through performances evaluation in testbeds from WP5. | Application performances will be verified in the ARNO testbed. |
| **Obj. 2: Employ a cloud-native approach to vertical service and mobile network deployments over heterogenous and dynamic resources that span across multiple administrative domains** | |
| R&D Topic 1: development of custom interfaces to facilitate the deployment of DESIRE6G functions and application for the final user. | SMO will be designed for an efficient deployment of the application in an end-to-end fashion, possibly including the user equipment at the drone. |
| Verification: validate DESIRE6G components through performances evaluation in testbeds from WP5. | Application performances will be verified in the ARNO testbed. |
| **Obj. 7: To integrate components and to build PoC demonstrators validating the whole architecture** | |
| R&D Topic 1: integrate and validate DESIRE6G components in a large-scale testbed such as ARNO. | The VR/AR PoC will be validated in the ARNO testbed. |
| R&D Topic 2: test and validate perceived real-time applications, both in terms of data | Perceived zero-latency VR/AR application will be demonstrated in the real-world ARNO testbed. |

| | |
|---|---|
| recording, object recognition and in view control from the user. | |

TABLE 13. POC#1 RELATION WITH PROJECT OBJECTIVES

### 3.1.3. Relation with project use case

This PoC is based on the AR/VR use case described in D2.1 and aims to fully develop it, from proper network design to the hardware selection and image processing algorithms. The PoC will handle the selection of all the hardware components and the design of the communication network between each node, as well as the setup of any programmable portion of the network. The final structure will also be tested for demonstration and verification.

### 3.1.4. Hardware and software configuration

The three nodes composing the VR/AR application will be the Source Node, the Edge Node and the User Node: nodes serve to group hardware modules by their function for the final application and each node may include more than one module. Modules of the same node may be placed at different locations: for example, the Source Node will include a fixed camera as well as cameras mounted on the drone.

The Source Node will consist in a drone able to bring a camera (or cameras) and granting a superior point of view on the area of interest: controlling the drone and the cameras will be performed through different channel for safer and faster communication. The employed drone needs to be a flying device able to bring the cameras around the area of interest. Up to four Ultra-low-light cameras from NVIDIA Jetson AGX Orin will be employed, granting lightweight, programmable, source of images from multiple angles. The source node will provide images streaming to more powerful computing resources, located into the Edge Node. The communication will be assured by onboarding a 5G dongle in the drone.

The Edge Node will rely on wired connectivity to the User Node, allowing for fast transmission of elaborated date, while it will rely on P4 switches and RAN connectivity to communicate with the Source Node. The Edge Node will be responsible for image processing and enhancement before sending processed information to the User Node. Thus, the Edge Node will work as a serverless framework, providing resources for communication between users and drones and for data analysis. The Edge Node will exploit one or more application functions relying on image processing libraries for object detection (e.g., YOLO v6) and for AR/VR reconstruction (e.g., OpenCV). The network will consist of radio controllers (e.g., non-RT and near-RTRIC), SDN controllers (P4 programs and entries) and elements necessary to provide enough computational power to grant perceived zero-latency in communication and data analysis.

The User Node will be provided with an AR/VR headset (namely, an Oculus 2 or Oculus 3 device) worn by the operator: the operator will be able to communicate with the camera system of the Source Node through a forward/backward channel. The forward chain will allow to get processed images directly on the headset, enhancing the view of a remote area under survey with proper markers and labels on any possible element of interest. The backward chain will allow the operator to take actions on cameras while surveying. The user will not be responsible for controlling the drone motion, which will be handled by another communication line, to not have to share its traffic with other data. This will help in having faster communication.

Finally, the application will include an RTK device to provide calibrated position assessment between the drone and the User Node: this is necessary to properly locate identified objects and allow for different camera systems to communicate between each other coherently. An RTK antenna will provide the reference frame origin, while an RTK node will be included with the headset (the User Node) and the drone (Source Node).

Figure 30 below, shows the application structure and its nodes and shortly explains their role within the overall system.

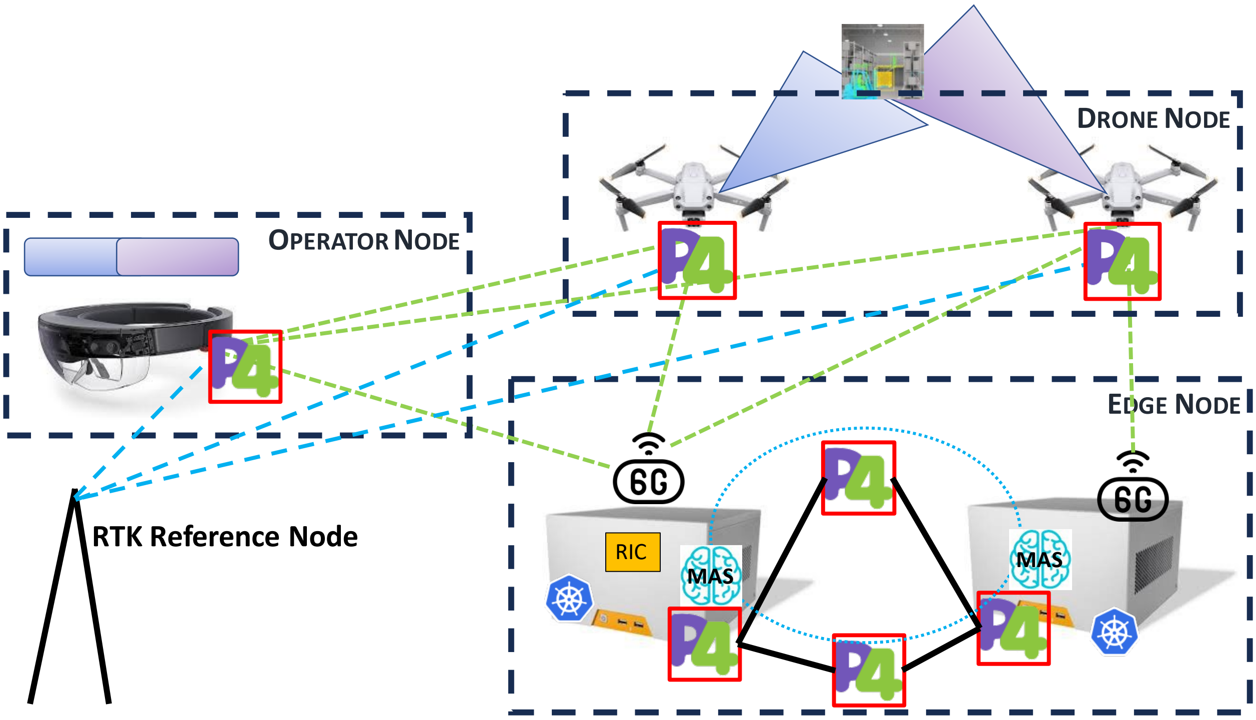


FIGURE 30. APPLICATION OVERALL STRUCTURE

For what concerns software configuration, the Edge Node will be equipped with proper AI-based algorithms to perform objects detection and labelling and able to provide real-time indications on the user's headset. Open-source material will be taken into account for this purpose, while the application will leverage on NVIDIA-specific software components for setting communication between all the nodes. RTK calibration and navigation system and camera streaming will be integrated in the system to correctly exchange information between all the components involved in the application.

### 3.1.5. Required DESIRE6G components

| Sub-System | Component name | Description |
|---|---|---|
| SMO | Service orchestrator | SMO component to efficient deploy the RAN network functions and the AR/VR application in an end-to-end fashion, possibly including the user equipment at the drone. |
| IML | Infrastructure Management Layer | IML component to abstract the physical resources to deploy NFs and AR/VR application. |
| SOL + vAccel | | DESIRE6G components integrated with SMO and IML required to optimize and speed up AI execution especially at end points and at the edge. |
| MAS | | MAS functionality in order to monitor the latency of the different segments at the flow level and suggest/trigger the most suitable segment re-optimization. |
| Monitoring | Pervasive Monitoring service | Pervasive end-to-end monitoring including the RAN, the P4 domain (INT) and the edge/cloud instances. |

TABLE 14. POC#1 REQUIRED DESIRE6G COMPONENTS

### 3.1.6. Initial performance metrics

Table 15 provides an initial list of performance metrics which will be measured through the experiments using this PoC.

| Performance metric | Description | Way of measurement |
|---|---|---|
| Forward latency | One way delay between sending data from the Source Node and their reception as processed data on the User Node. | Time difference between sending an image from the Source Node and receiving the processed image on the User Node. |
| Backward latency | One-way delay between issuing a command from the User Node and its execution on the cameras on the Source Node. | Time difference between execution of a command and receiving the acknowledgment from the Source Node. |
| Tracking accuracy | Difference between objects true position and their position recorded by the RTK, especially during motion. | Root mean square error between the measured RTK positions and true positions of some sample elements. |
| Detection accuracy | Ability of the object detection model to correctly identify and localize objects within an image. It is a measure of both object recognition and localization. | The detection accuracy could be evaluated as the F1 score of the detection algorithm employed. This, will be the harmonic mean of precision and recall of the model. |
| Detection latency | One-way delay between receiving an image on the Edge Node and their reception as processed data on the User Node. | Time difference between receiving an image on the Edge Node and receiving the processed image on the User Node. |

TABLE 15. POC#1 PERFORMANCE MEASURES/METRICS

### 3.1.7. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q4 – 2023 | • Purchase use case equipment (dongles, RTK, ORIN, cameras, headset) |
| Q1 – 2024 | • Starting preliminary app deployment |
| Q2 – 2024 | • Preliminary demonstration (Y2 demo) |
| Q3 – 2024 | • Starting final app deployment |
| Q2 – 2025 | • Integration of the final app with final DESIRE6G components (containerized chain) |
| Q3 – 2025 | • Validation results |
| Q4 – 2025 | • Final demo |

TABLE 16. POC#1 TIMELINE

### 3.1.8. Preliminary demo execution plan: AR with end-to-end pervasive monitoring (Demo 1)

This Y2 preliminary demo has the main goal of demonstrating a preliminary end-to-end pervasive monitoring operation on different network segments and active service re-optimization at runtime exploiting the MAS concept. The demo is applied on the AR use case PoC described in Section 3.1.

Three preliminary components of the DESIRE6G solution will be considered in the demo: 1) the SMO that will provide the service deployment; 2) the MAS as decentralized element aiming at monitoring the real-time performance of the service in the different network segments, while suggesting flavours of network and/or service re-optimization actions in the case of SLA violations; and 3) the pervasive monitoring components deployed in the PDP network segment and in the RAN segment.

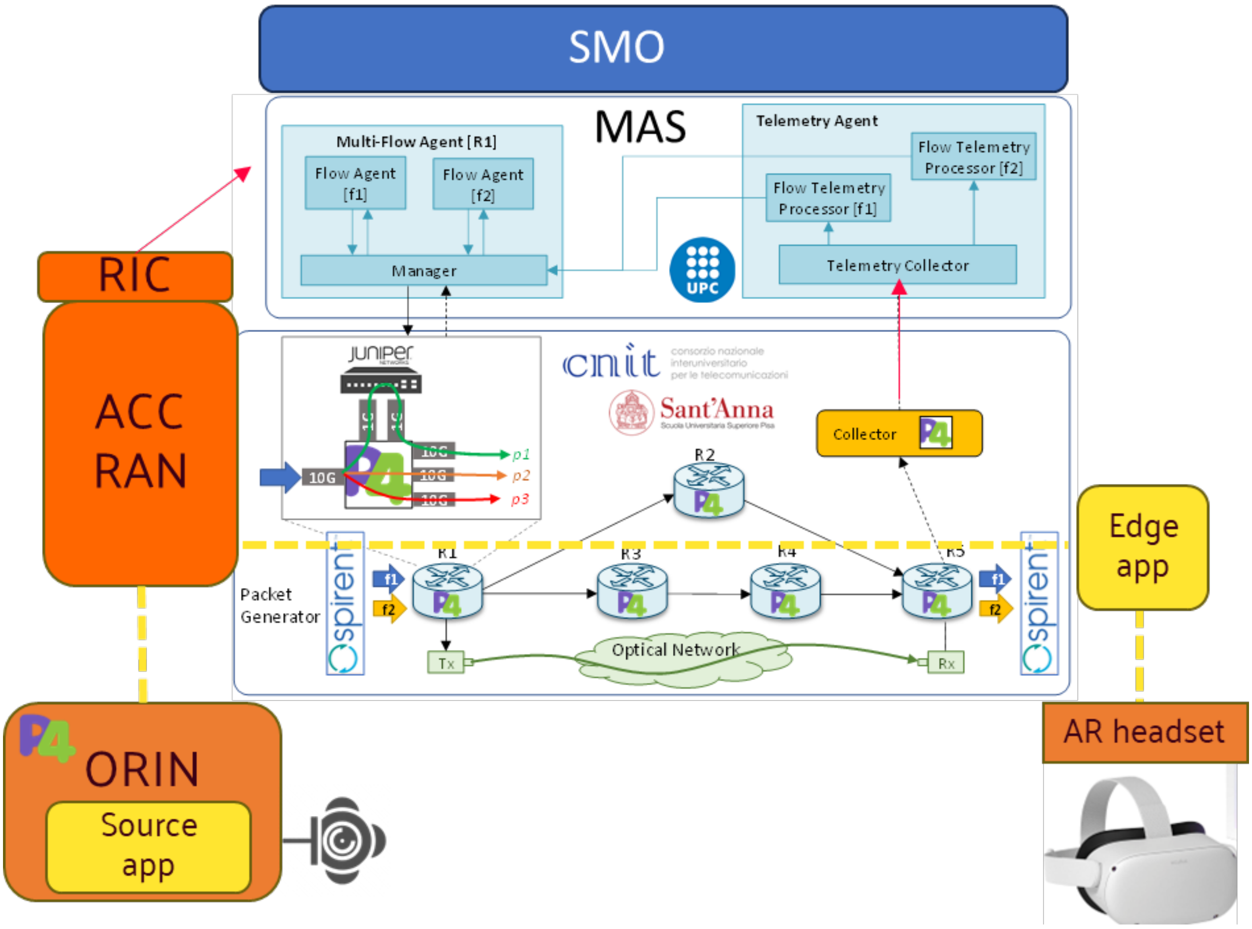


FIGURE 31. PRELIMINARY AR DEMO SCENARIO (Y2 DEMO)

The AR application is deployed in a preliminary version and includes a remote camera video streaming locally augmented with object recognition, further processed with a baseline AR library in the edge node. Moreover, the processed data is sent to an operator end-user, equipped with the AR Headset (i.e., Oculus) to catch the real time augmented video. The application traffic is end-to-end monitored thanks to different telemetry sources: 1) the user equipment telemetry, co-located with the camera source, 2) the RAN monitoring features, provided by the RIC through xApp (e.g., radio channel quality of service parameters); 3) the P4 network INT. Different MAS agents are responsible for processing the telemetry information generated by each segment related to the deployed service. A number of degradations will be considered, either at the radio, the P4 and the application levels in different network conditions and load. For some of these scenarios, being a single network segment unable to provide an effective re-optimization able to restore the desired QoS (i.e., the end-to-end latency requirement), the MAS agents will exploit East/West communication to exchange telemetry performance and compute the most suitable reconfiguration, to be suggested for enforcement to the involved segment(s).

Figure 31 shows the demo scenario. The AR application will be distributed between two components (source node process and edge node process), running at the endpoint user equipment and inside the edge node co-located with the RAN site. An NVIDIA ORIN board equipped with local cameras will be exploited to process the video and apply preliminary telemetry P4 NF to augment streaming packets. The ORIN board will be connected to the RAN site by means of a 5G dongle.

At the P4 network domain, a P4 telemetry collector will be adopted to receive INT from all the nodes, compute statistics and send them to the MAS agent.

The demo will be hosted in the ARNO testbed, featuring P4 domains and edge nodes, integrating the RAN components provided by ACC, the preliminary SMO by NUBIS, the MAS agents by UPC and the pervasive monitoring features by CNIT/SSSA, NVIDIA and UVA.

Table 17 shows the Demo 1 execution plans timeline.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | Start the baseline PoC implementation:<br>• Start the preliminary AR application development,<br>• Consolidate P4 domain deployment,<br>• Integrate the ACC RAN<br>• Preliminary integration with the MAS agents |
| Q2 – 2024 | Preliminary Integration of DESIRE6G components:<br>• SMO<br>• MAS (multi-segment re-optimization feature)<br>• Per-segment Pervasive Monitoring |
| Q3 – 2024 | Finalize the DESIRE6G AR demo |

TABLE 17. DEMO 1 TIMELINE

## 3.2. PoC#2: Digital Twin

### 3.2.1. Description

Digital Twins in the field of robotics serve as digital replicas of real-world robots, offering a real-time, consistent, and accurate virtual representation of their physical counterparts. While the concept is not novel, its broader application has been limited due to a segmented approach, focusing only on the virtual representation and not the entire encompassing system.

This PoC explores an all-encompassing DT system that is spread across the Device-Edge-Cloud. The goal is to balance the computational, storage, and networking demands of the DT application. In this setup, robots are either controlled remotely in real-time or operate autonomously using algorithms based in the Edge. They continuously transmit sensor data upstream (e.g., lidar, camera, odometry, joint states) to update their virtual models, while simultaneously receiving real-time navigation instructions downstream. The challenge lies in minimizing the delay between collecting sensor data and acting upon it. The in-network acceleration and optimization, along with the E2E data plane programmability offered by DESIRE6G, aim to reduce this delay and meet the strict KPIs of operational digital twins, mainly related to reliability and low latency. In addition, the PoC can benefit from DESIRE6G E2E service orchestration for efficient management throughout its life-cycle, optimizing resource usage and energy efficiency.

### 3.2.2. Relation with project objectives

Table 18 highlights the relation of the DT PoC with the project objectives.

| Project objectives | How will this PoC address the project objectives |
|---|---|
| **Obj. 1: Design a functional architecture for 6G mobile networks to support the next generation of URLLC use cases** | |
| R & D Topic 1: real-time synchronization and data processing for URLLC in DT systems. | The DT PoC will validate the functional architecture designed for 6G mobile networks by demonstrating the real-time synchronization and data processing capabilities required for URLLC use cases. |
| Verification: validate selected DESIRE6G components through a system verification in the integration testbeds from WP5. | Components will be integrated and verified in the 5TONIC testbed. |
| **Obj. 2: Employ a cloud-native approach to vertical service and mobile network deployments over heterogenous and dynamic resources that span across multiple administrative domains** | |
| R & D Topic 1: develop interfaces to automate the deployment of DESIRE6G functions and applications. | SMO allows automated deployed of Digital Twin application. |
| Verification: validate selected DESIRE6G components through a system verification in the integration testbeds from WP5. | Components will be integrated and verified in the 5TONIC testbed. |
| **Obj. 7: Integrate components and build PoC demonstrators validating the whole architecture.** | |
| R & D Topic 1: integrate and validate DESIRE6G components in large-scale testbeds, such as 5TONIC and ARNO. | The DT PoC will be demonstrated in the 5TONIC testbed. |
| R & D Topic 2: demonstrate low latency applications, such as augmented reality and digital twin, in real-world scenarios involving real users. | Low latency application “Digital Twin” will be demonstrated in the real-world 5TONIC testbed. |

TABLE 18. POC#2 RELATION WITH PROJECT OBJECTIVES

### 3.2.3. Relation with project use case

The DT use case defined in D2.1 envisions the implementation of a DT system for predictive maintenance and optimization in real industrial settings. The DT PoC aims to integrate different DT features and validate their functionalities, but there are some limitations to consider.

For instance, the PoC uses commercially available robots instead of industry-specific ones and relies on open-source software instead of proprietary solutions. Additionally, the robot virtual functions of the DT application operate within Docker containers rather than directly on industrial controllers.

However, the PoC effectively demonstrates critical features such as seamless physical-virtual integration, data collection-sharing, and digital model simulations. It offers a flexible and cost-effective approach to validate and demonstrate the DESIRE6G capabilities.

### 3.2.4. Hardware and software configuration

This PoC requires the following hardware components to run different software elements:

- **Robots –** The robots would be pre-built robots (e.g., Unitree Go1, Niryo One) equipped with various sensors (e.g., cameras, lidars). In addition, they will have heterogeneous constrained devices (e.g., Raspberry Pi, Jetson Nano) with software component running on top the Robot Operating System (ROS). In Figure 32, the Unitree Go1 robotic dog can be seen on the left, while the Niryo One robotic arm is displayed on the right.


FIGURE 32. UNITREE GO1 ROBOT (LEFT) AND NIRYO ONE ROBOT (RIGHT)

- **Radio Access Technologies –** Low-latency technologies for connecting the robots such as 5G or Wi-Fi 6. This enables real-time monitoring and control of DT functionalities, resulting in improved and seamless interaction between robots, sensors, and the DT system.
- **Cloud and Edge computing platforms –** These platforms enable the orchestration and management of the Digital Twin virtual functions. Virtual functions are containerized, utilizing Docker for virtualization, to facilitate deployment and scalability.
- **In-network devices –** This includes switches, routers, and commodity Cloud/Edge servers equipped with GPUs and CPUs. GPUs are especially important for enabling virtual replicas to operate effectively.

Figure 33 depicts the integration of these components in the PoC.

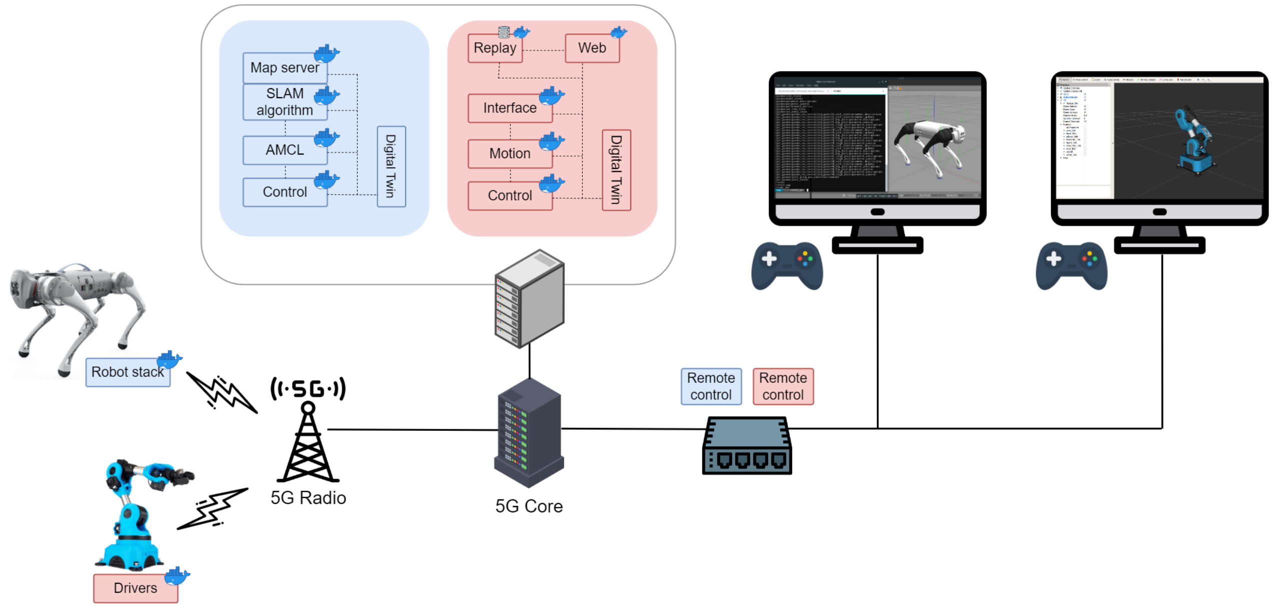


FIGURE 33. POC#2 INTERCONNECTED SYSTEM COMPONENTS

### 3.2.5. Required DESIRE6G components

Table 19 provides the list of the DESIRE6G components that are needed to create a working DT PoC.

| Sub-System | Component name | Description |
|---|---|---|
| SMO | Service orchestrator | SMO component to efficiently manage the DT application throughout its life-cycle. |
| IML | Infrastructure Management Layer | IML component to abstract the physical resources to deploy NFs and DT application, and manage the network traffic. |
| SOL | SOL NN acceleration server | DESIRE6G component integrated with SMO and IML that contributes to meet stringent KPIs. |
| Monitoring | Pervasive Monitoring service | Pervasive end-to-end monitoring for reporting statistics information from the RAN, the P4 domain (INT) and the computing infrastructure. |

TABLE 19. POC#2 REQUIRED DESIRE6G COMPONENTS

### 3.2.6. Initial performance metrics

Table 20 presents a preliminary set of performance metrics that will be measured through the experiments using this PoC. Additional metrics are being considered, and early results from ongoing experiments will be detailed in the next deliverable.

| Performance metric | Description | Way of measurement |
|---|---|---|
| Latency | Round trip time between issuing a command from a remote controller and execution from a robot. | Time difference between execution of a command and receiving the acknowledgement from the robot. |
| Jitter | Variability in latency measurements over time, which can affect the predictability of real-time robot control. | Calculate the standard deviation or variance of latency over a series of command executions. |
| Response time | The time taken for the DT system to respond to a query or command. | Time from the moment a command is issued until the first response is received. |
| Resource utilization | Computational and storage resources used while the system is running. | Percentage of CPU, memory, and storage used during the PoC operation. |
| Reliability | The ability of the DT system to consistently and reliably perform its intended functions under specified conditions. | Evaluation of system uptime, error rates, and successful task completions during operational periods. |

TABLE 20. POC#2 PERFORMANCE MEASURES/METRICS

### 3.2.7. Integration and validation timeline

Table 21 depicts the timeline of the integration and validation process for the DT PoC.

| Planned deadline | Planned integration & validation task |
| --- | --- |
| Q4 – 2023 | • Research and planning of the PoC hardware and software components<br>• Develop SW building blocks |
| Q2 – 2024 | • Building functional demo (Y2 demo) |
| Q3 – 2024 | • Installation and testing activities inside the 5TONIC testbed |
| Q1 – 2025 | • Final app integration with final DESIRE6G components |
| Q3 – 2025 | • Validation results |
| Q4 – 2025 | • Final demo |

TABLE 21. POC#2 TIMELINE

### 3.2.8. Preliminary demo execution plan: Real-time Digital Twins (Demo 2)

The main goal of the second-year demo is to showcase the automated deployment and traffic management of an end-to-end DT system on top of the compute continuum. This demonstration will highlight and validate two fundamental components of the DESIRE6G ecosystem: 1) the SMO that will provide the efficient automation of the life-cycle management of the end-to-end DT system and 2) the IML that will provide runtime optimizations optimizing the traffic load in the data plane taking the advantages of end-to-end network programmability.

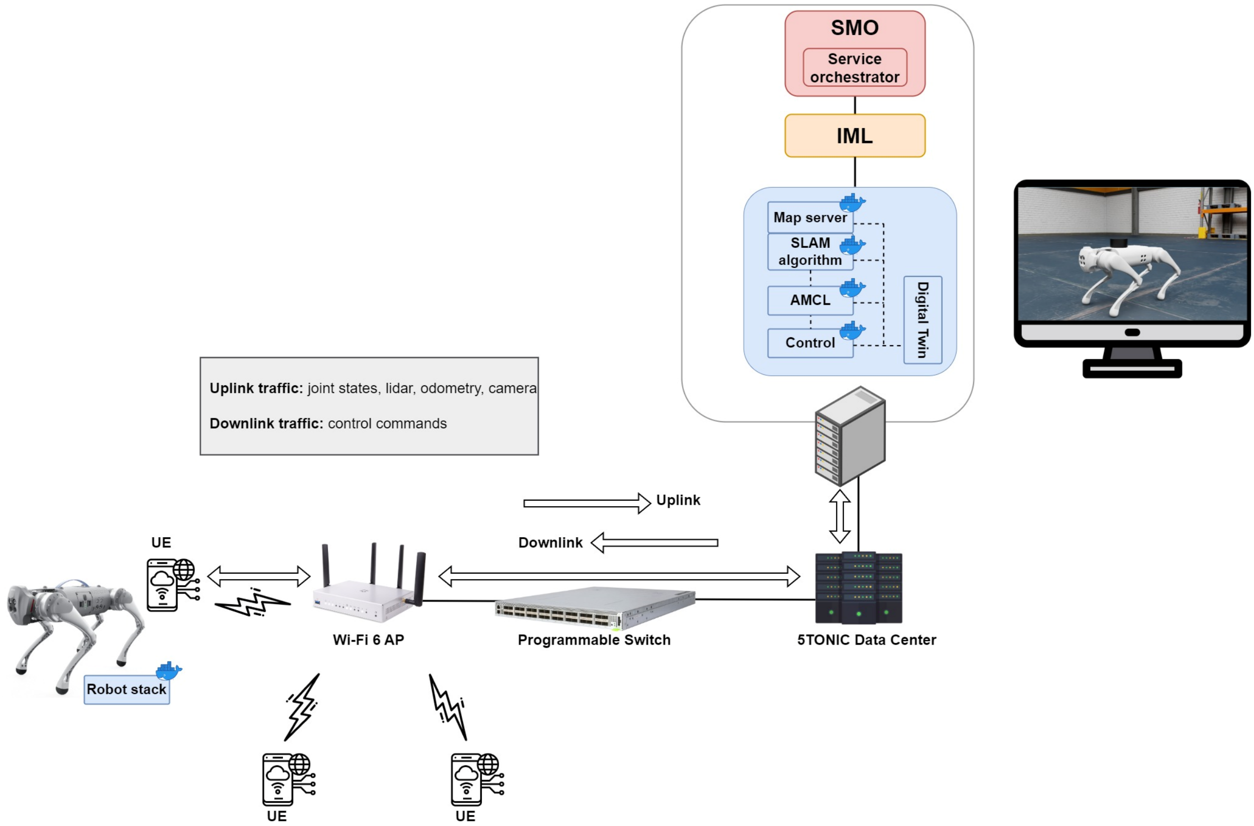


FIGURE 34. PRELIMINARY DT DEMO SCENARIO (Y2 DEMO)

The demo scenario is depicted in Figure 34 The infrastructure comprises a quadruped robot equipped with UE and connected to a Wi-Fi 6 AP. This AP is linked to programmable switches, forming a network backbone that extends to the 5TONIC datacentre. Other devices equipped with UEs are also connected to the AP, generating diverse industrial traffic flows within the network. Robot control commands are transmitted downstream while robot state and data from its sensors (lidar, camera and joint states) are transmitted upstream to keep the virtual replica synchronized. The robotic software is implemented using ROS, while the DT application utilizes the open-source robot simulator IsaacSim[6].

The demo will start with showcasing the end-to-end digital twin service instantiation future of the SMO. During the deployment phase, the SMO will instruct the IML to deploy the Application Functions (AFs) as well as Network Functions (NFs) that compose the DT system. This includes deploying the DT

[6] https://developer.nvidia.com/isaac-sim

application and connecting the robot to the network. Virtual remote controller (manual or autonomous) that will reside in the edge of the network will control the robot to perform a given task. In the same time, information from Lidar, Camera and Odometry sensors will be used to update the virtual model of the DT application in real-time. Next, the emulated industrial traffic will be used to create an industrial scenario with high traffic load. The increased network traffic load will directly lead to violations of the key KPIs of the real-time Digital Twin application (latency, jitter, and packet loss). As a result, the DT application will suffer performance degradations that can be observed. The crucial role of the IML in this context will be to manage traffic programmatically, giving priority to essential and time sensitive robot traffic flows (e.g., lidar, odometry, control commands) ensuring that critical robot operations (e.g., SLAM, autonomous navigation) can be performed even under network congestion.

Table 22 outlines the schedule for the execution plans of Demo 2.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | Finalize the baseline PoC implementation:<br>• Finish the DT application development<br>• Virtualize the AFs<br>• Integrate the applications |
| Q2 – 2024 | Integration of DESIRE6G components:<br>• SMO<br>• IML |
| Q3 – 2024 | Finalize the DESIRE6G DT demo |

TABLE 22. DEMO 2 TIMELINE

## 3.3. PoC#3: Intelligent Image Monitoring

### 3.3.1. Description

In this PoC, we focus on an industrial site where different IP camera streams are forwarded to an industrial controller application running in the (edge-)cloud via a critical link (e.g., a radio channel). Robots in the manufacturing area are managed by an industrial controller running in the cloud. The remote controller relies on the streams of IP cameras deployed in the area. We assume that not all the video streams are important at any time, as the state of the robot determines which camera streams are needed with high quality. The robots also send status messages to inform the controller about their activity. Both the status messages and the video traffic flow pass through a programmable aggregation point (e.g., P4 switch) at the industrial site before reaching the critical link. The network offers a quality control API through which high-level rules (e.g., a set of bounding boxes in which high quality video is needed for a given camera) can be defined for each camera stream and robot. These high-level rules are then mapped to low-level rules stored in match-action tables or compiled directly into the data plane. Based on the rules provided by the operator and the status of the robots, the aggregation point can carefully drop packets from the video streams without compromising the required functionality and ensuring the desired quality level. Note that the proposed method drops packets even when bandwidth is available, reducing the load on the critical link without affecting the behaviour of the control application.

In the DESIRE6G infrastructure, we will show that if a programmable switch is located at the UE side (considering the UE side as a limited DESIRE6G site), the quality control data plane function can easily be deployed before the critical link and thus can reduce the load to be handled by the radio, pointing to the benefits of end-to-end network programmability.

### 3.3.2. Relation with project objectives

| Project objectives | How will this PoC address the project objectives |
|---|---|
| **Obj. 1: Design a functional architecture for 6G mobile networks to support the next generation of URLLC use cases** | Transmitting the aggregated traffic of high-resolution cameras over a single critical link (e.g., radio) may generate significant load. As not all camera streams are needed with high quality all the time, selectively reducing the quality of less important video streams can lead to significant bandwidth saving. Less load can allow to satisfy the functional requirements of real time control. |
| **Obj. 2: Employ a cloud-native approach to vertical service and mobile network deployments over heterogenous and dynamic resources that span across multiple administrative domains** | This PoC will demonstrate that programmable data plane resources located at the industrial premises can also be integrated into the DESIRE6G infrastructure and NFs can be deployed there with the same framework. This may also lead to performance improvements in the application. |
| **Obj. 4: Unified management and control of heterogenous programable data planes and hardware accelerators, while enabling increased controllability for the tenant and service.** | This PoC demonstrates how end-to-end network programmability can be managed by the unified programmable data plane layer through IML UE side can also act as a limited D6G site and can be integrated into the management and orchestration framework of DESIRE6G. |

TABLE 23. POC#3 RELATION WITH PROJECT OBJECTIVES

### 3.3.3. Relation with project use case

The Intelligent Image Monitoring use case defined in WP2 envisions a network with the ability to control the quality of selected IP camera streams. The network also provides a quality control API through which the quality controlling rules can be defined for different states of the industrial environment. This PoC aims to implement the use case with emulated camera streams. The aggregation point is a P4 programmable switch while the critical wireless link is emulated by a rate limited link connected to a DESIRE6G Edge site. A control plane entity for this PoC hosted by the edge site provides a quality control API through which the video stream filtering rules can be added, removed and modified.

### 3.3.4. Hardware and software configuration

The PoC will be deployed in the local testbed of ELTE or ERI-HU. A dedicated bare-metal server will be responsible for emulating the IP camera streams in the industrial area. The aggregation point will be implemented by a P4 programmable switch (e.g., a low cost APU, or a Tofino). This node will run a local IML instance for managing the switch resource (or other resources if needed). The edge DESIRE6G site will consists of one or two P4-programmable (Tofino) switch (one of them will be the gateway) and two servers forming a K8S cluster. The server application terminating the video streams will run in the K8S cluster. The edge site will have its own IML instance responsible for managing the site resources. The wireless link will be emulated by a rate limited wired link between the aggregation point and the gateway switch in the emulated DESIRE6G edge site.

### 3.3.5. Required DESIRE6G components

| Sub-System | Component name | Description |
|---|---|---|
| SMO | Service Management Orchestrator | High level service will be defined end-to-end. The SMO will be responsible for the deployment of the service according to deployment constraints (e.g., IP-Cam Quality Control DP needs to run close to the camera streams before the radio link). |
| IML | Infrastructure Management Layer | Two IML instances will manage both the UE-side aggregation point and the D6G edge site running the demo server application terminating the IP camera streams. |
| IP-Cam Quality Control DP | Data plane of IP camera stream quality control (P4 program) | Data plane program that processing sensor streams and apply the decision logic for controlling the quality of IP camera streams. The program selectively drops P-frames of H265 video stream if quality degradation is allowed. |

| | | |
|---|---|---|
| IP-Cam Quality Control CP | Control plane of IP camera stream quality control with quality control API | This module provides a REST API through which the tables and registers of the quality control data P4 program can be filled and modified in run-time. The application can run in either the edge cloud or UE side. |
| Industrial Environment Emulator | Industrial Environment Emulator | Emulating the sensors and actuators in an industrial environment and generating the corresponding sensor traffic that can be used to determine the state of the industrial environment for quality control. |

TABLE 24. POC#3 REQUIRED DESIRE6G COMPONENTS

### 3.3.6. Initial performance metrics

Table 25 provides an initial list of performance metrics which will be measured through the experiments using this PoC.

| Performance metric | Description | Way of measurement |
|---|---|---|
| Latency | Round trip time between issuing a command from a remote controller and execution from a robot | Time difference between execution of a command and receiving the acknowledgement from the robot. |
| Bandwidth | Comparison of the bandwidth usage on the critical link for different camera streams with this PoC enabled and disabled | Send video streams over the critical link |
| Packet loss | Quantify the packet loss rate for video streams and status messages with and without the proposed method | Count packet loss |
| Video quality | Measure the video quality for important and not important video streams | Capture the video on the receiver side, and compare it to the original video |
| Quality change latency | Measure how much time is required for a given video stream to reach the given quality | Mark a stream from important to unimportant and back |

TABLE 25. POC#3 PERFORMANCE MEASURES/METRICS

### 3.3.7. Integration and validation timeline

| Planned deadline | Planned integration & validation task |
|---|---|
| Q4 – 2023 | • Develop SW building blocks, functional testing of components |
| Q1 – 2024 | • Integration with SMO and IML, development of quality control API |
| Q2 – 2024 | • Final proof-of-concept demo |

TABLE 26. POC#3 TIMELINE

# 4. Preliminary demo execution plan of individual DESIRE6G components

This section describes the demonstration scenario and execution plan for the individual DESIRE6G components for the first half of the project.

### 4.1.1. Deployment of Secure Machine Learning Pipelines for Near-Real-Time Control of 6G Network Services (Demo 3)

Near-real-time autonomous network operation is required to deal with the expected large traffic dynamicity and provide the stringent performance required by beyond 5G and 6G network services (NS). Solutions for autonomous operation running in a centralized controller have the potential to greatly reduce costs, but they might lead to inefficient resource utilization because of their long response times. To minimize response time, control algorithms (agents) might be executed as close as possible to data plane devices]. Nonetheless, such distributed control might bring security concerns as their exposure to software attacks is higher than that of the centralized control. In this regard, DLT have recently attracted attention, as they create a shared, immutable, and decentralized record of transactions. Applications of DLT in the context of 5G services have been proposed and showed that the consensus mechanism is key for the overall performance. However, even with the simplest consensus mechanism, data exchange can be slow for near real-time applications. For this very reason, DLT can be combined with Virtual Extensible LANs (VXLAN) to bring any added delay to a minimum.

In this demonstration, we will showcase the deployment of a secure ML pipeline consisting of: i) a set of intelligent agents deployed in distant locations that coordinate among them for the near-real-time control of a NS; and ii) the required communication infrastructure. We target the control of a 6G NS. Specifically, the demonstration will show the integration of: i) a ML Function Orchestrator (MLFO) coordinating the deployment of agents and their configuration with the help of a service management and orchestration system (SMO) and a topology server based on ALTO, fed with topological, IT and networking information; ii) a DLT with a smart contract used for key/secret exchange among the agents participating in the control of the NS and the MLFO; and iii) a distributed telemetry processing and data exchange among the agents. The systems in the demonstration are deployed in the UPC premises in Barcelona (Spain).

The secure deployment of a ML pipeline requires careful design as entities in distant locations are exchanging information of the underlying network topology and the configuration of optical devices over a public infrastructure, which can be used to craft specific attacks in case of eavesdropping. On the one hand, the use of DLT introduces additional delay that might impact on the near-real-time operation of the NS. On the other hand, VXLAN presents some security concerns, such as the possibility for rogue devices to join one or more multicast groups and inject fake traffic. Encryption protocols, such as IPsec, encrypt both the payload and the inner headers, which reduces rogue risk, as compared to other real-time encryption protocols at the application level. In addition, to reduce the delay added for encryption and to allow other agents to verify and trace communications, in the demonstration we will use pre-shared keys. Then, this solution requires an authentication infrastructure so that authorized agents can obtain and distribute these keys. Our demonstration will rely on the use of smart contracts on a DLT infrastructure to facilitate the coordination and collaboration of the agents and the MLFO. However, DLT exchange is kept offline while VXLANs are used for near real-time communication among the agents. In particular, the security intrinsic features of DLTs can be used to securely manage VXLANs as communication channels among the agents. Note that the impact of the used consensus mechanism is limited to the set-up phase of the dynamic associations between agents and it does not have an impact on the actual exchanges between them once the associations are established.

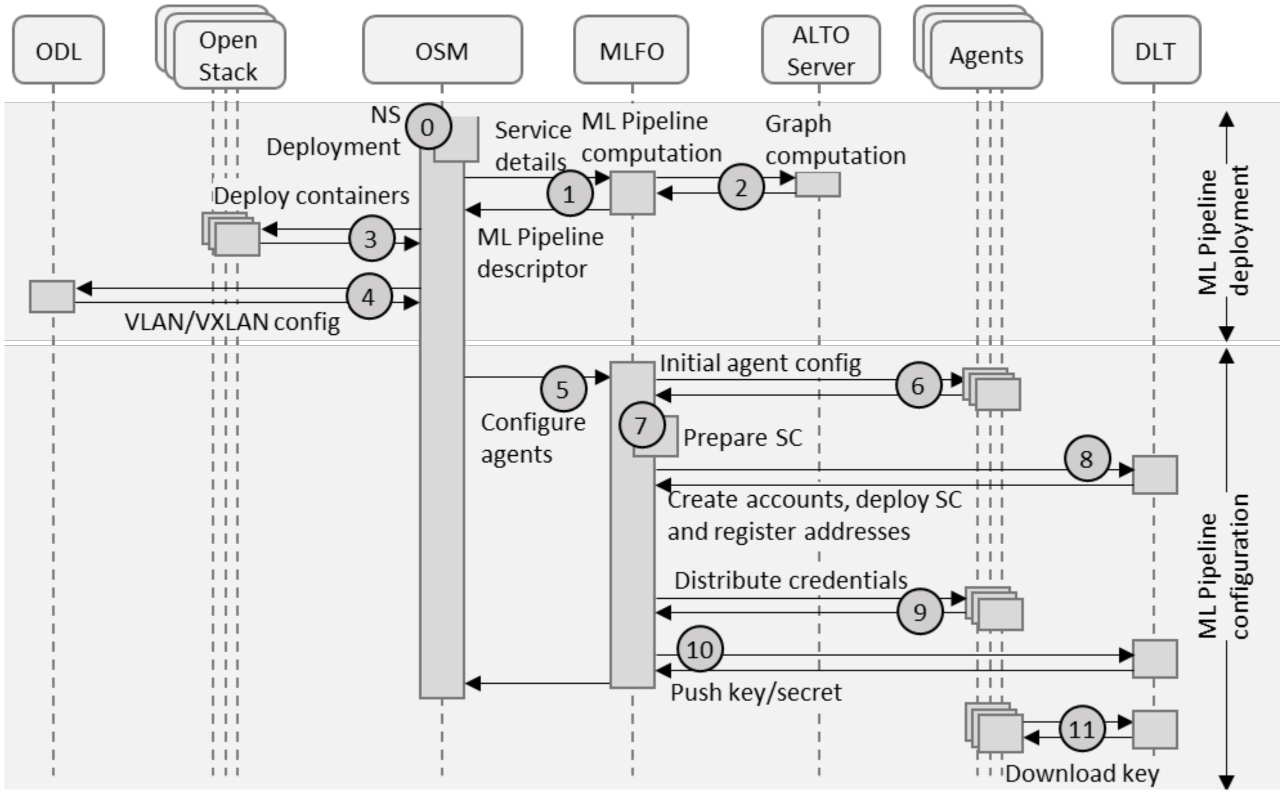


FIGURE 35. ML PIPELINE DEPLOYMENT WORKFLOW TO BE DEMONSTRATED (DEMO 3)

The architecture defined in Figure 14 will be used for this demonstration. The proposed workflow is sketched in Figure 35, which is carried out as a part of the NS deployment. We assume that the MLFO has been previously configured with the needed credentials in the DLT network. The workflow consists of two phases: i) ML pipeline deployment; and ii) agents' configuration and key exchange. The SMO initiates the workflow after the NS is deployed (0 in Figure 35). The SMO starts with the ML pipeline deployment and requests the definition of a ML pipeline for the NS and provides its details, including the location of the VNFs (1). Based on the NS details and the requirements of the ML pipeline in terms of delay and throughput among the agents, as well as the required IT resources of the agents, the MLFO requests the ALTO server to compute a graph with the resources in the network infrastructure that meet the requirements and can be used to support the ML pipeline (2). With such data, the MLFO computes the optimal ML pipeline design and sends back a descriptor containing the location where the agents need to be deployed, the VM image to be installed and the connectivity to be created. A list of iterations is generated that includes the communication of OSM with the OpenStack managers for the deployment of the agents encapsulated into virtual machines (VM) (3), and with the SDN controller for managing the connectivity (4). Once the agents are running and the connectivity is available, the ML pipeline is deployed and the configuration phase starts (5). When the MLFO receives the request to configure the agents, it sends the initial configuration that includes the addresses of the VNFs and that of the other agents, as well as the algorithms that every agent runs (6). After that, the MLFO compiles the smart contract that will be used to store the key for that ML pipeline (7). Next, the MLFO creates the DLT accounts for the agents, deploys the smart contract and registers the addresses with credentials to interact with the smart contract (8). In this way, the MLFO can control the access to the smart contract and add or revoke permissions if needed in case of ML pipeline reconfiguration. Once the addresses are registered, the MLFO distributes the credentials to interact with the smart contract among the agents involved in the ML pipeline (9). The agents connect to the smart contract through the local DLT node. The MLFO generates a random key that uses to initialize IPSec for secure communication through the established VXLAN and pushes the key to the smart contract (10). At this time, the deployment of the NS ends from the viewpoint of OSM. Once the transaction is validated by the DLT network, agents receive a notification, download the key and use for secure communications with other entities in the ML pipeline (11) for near-real-time control of the NS.

The workflow of the demonstration will be as follows: i) a NS with three VNFs is deployed, which triggers the deployment of a secure ML pipeline for its near-real-time control; ii) two ML pipelines will be demonstrated, one with one single centralized agent collecting telemetry and making decisions, and another distributed. The graph computed by the ALTO server will determine the ML pipeline that is deployed at every request; and iii) the MLFO configures the DLT for key exchange and the agents download the key and join the VXLAN.

A Web interface will facilitate iteration of the attendees with the system, so they can modify the configuration of the different components of the architecture.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | First release of the demo:<br>• DLT<br>• ALTO<br>• MAS |
| Q3 – 2024 | Integration of the Attestation mechanism |
| Q4 – 2024 | Finalize the DESIRE6G demo |

TABLE 27. DEMO 3 TIMELINE

### 4.1.2. MAS and Telemetry for service operation demo (Demo 4)

Near-real-time autonomous operation, i.e., when actions are performed in the seconds scale, is a special type of network automation that requires the availability of new technologies and architectures to be developed in the control and data planes. One of the required technologies is that of pervasive in-band network telemetry (INT) to get accurate measurements of relevant metrics, including Quality of Service (QoS), e.g., delay and/or jitter, of the services supported by the network.

Dynamic flow routing is a use case of near-real-time network operation that requires the interaction between telemetry agents and flow agents in charge of making routing decisions, which eventually leads to the deployment of a distributed MAS. Among different routing policies, multi-path routing introduces flexibility in the design and operation of the network by allowing operators to split traffic demands into multiple streams that are routed independently of each other to the destination. A possible routing policy is to evenly split each traffic flow among all available routes. However, such a strategy might fail under dynamic network conditions that can generate congestion in some routes and consequently, lead to flows QoS degradation. Moreover, routes might have different utilization costs, and hence, the percentage of traffic sent through each route is a complex decision that needs to be dynamically tuned in order to meet robust QoS performance with overall minimum cost.

In this demonstration, we will showcase the near-real-time operation of an optical packet network controlled by a distributed MAS. The MAS combines: i) pervasive INT agents supported on P4-based components [5]; and ii) multi-flow routing agents that are used to dynamically adjust multi-path flow routing policies in the packet nodes with the objective to guarantee the target QoS performance. Hence, flow routing operation is controlled by a set of heterogeneous agents that are fed with telemetry data collected from P4 switches. The systems in the demonstration will be deployed in the distributed federated testbed including the CNIT/SSSA (ARNO testbed) in Pisa (Italy) and the UPC testbed in Barcelona (Spain).

For this demo, we rely on the federated testbed depicted in Figure 27, which reproduces a 5-node network scenario. We assume that traffic flows are splitable, i.e., they consist of a large number of sub-flows that can be routed independently. The objective is to find the flow routing policy that balances the incoming traffic of the flows among the available paths, so to ensure per-flow QoS (specifically e2e delay). Such routing policy varies as a function of the incoming traffic of the flow and the network conditions, i.e., the traffic of the rest of the traffic flows in the network. Therefore, the routing policy decision making process is continuously carried out based on the incoming traffic and the e2e delay measurements that allow evaluating the quality of the decision making. Note that the state of the network is known, and it is indirectly represented by e2e delay measurements for the traffic flow. In this demonstration, we rely on the INT functionality provided by the P4 switches to measure packet delay. Specifically, a P4 collector that collects, aggregates, and provides statistics of the delay measured by the switches supporting the traffic flows will be showcased. Once the P4 collector pre-processes the QoS measurements, they are sent to a telemetry agent, which is in charge of producing flow telemetry statistics that are sent to the flow agent deployed at the source node, where flow routing policy decisions are made. In our case, flows are identified by source and destination IP.

In the demonstration, R1 will measure per-flow traffic entering the switch, before being forwarded through the different paths. In particular, three alternative paths are considered: p1 (R1-R2-R5), p2 (R1-R3-R4-R5), and p3 (R1-R5, using the optical bypass). The flow agents are grouped in a single module per location named multi-flow agent. In this case, the multi-flow agent at R1 includes flow agents for both f1 and f2. Note that one single flow agent makes routing decisions for each traffic flow. The manager running inside multi-flow agents has the role of collecting and distributing flow telemetry data and local input traffic data to the flow agents, as well as to push the flow routing policies computed by flow agents to the P4 switch.

The workflow to be demonstrated for traffic flows f1 and f2 is outlined in Figure 36. Let us assume that flow routing policies are configured at switch R1 (0 in Figure 36) to enable multi-path routing. Input flow traffic measurements are collected and sent periodically to the local multi-flow agent for every traffic flow entering in the network (1); the messages contain both packet and bit count for every traffic flow. Packet delay measured along the path is reported though INT messages to the P4 collector (2), which performs aggregations and computes statistics, e.g., min, average, and max of delay and jitter.

Periodically, the P4 collector sends the computed delay statistics to the collector in the local telemetry agent (3). In addition, delay statistics are reported to a centralized telemetry system that stores them in a time series database (not shown in the workflow in Figure 36). The aggregated QoS telemetry message includes per-path statistics for every flow. The received statistics are then processed by each of the flow processors (4) that computes per-flow QoS telemetry measurements and send them to the corresponding flow agent (5). When flow QoS statistics are received in the multi-flow agent, the manager triggers its analysis by pushing both traffic and QoS measurements to the corresponding flow agent (6). Note that a pre-trained reinforcement learning model made routing decisions, with the operational objective of guaranteeing that the e2e delay does not exceed a configured threshold (QoS requirement), while minimizing the use of the optical bypass. However, to be able to interact with the system, in this demonstration, we will use a parameterized deterministic algorithm (7), so different routing policies can be manually forced by changing the values of some parameters, e.g., to force using the optical bypass. Routing decisions are gathered by the manager that eventually sends them to the P4 switch (8). Message 8 includes a routing policy defined as the percentages of input traffic to be routed through each of the routes. Internally, the P4 switch translates this policy into flow rules to efficiently perform packet forwarding according to defined percentages.

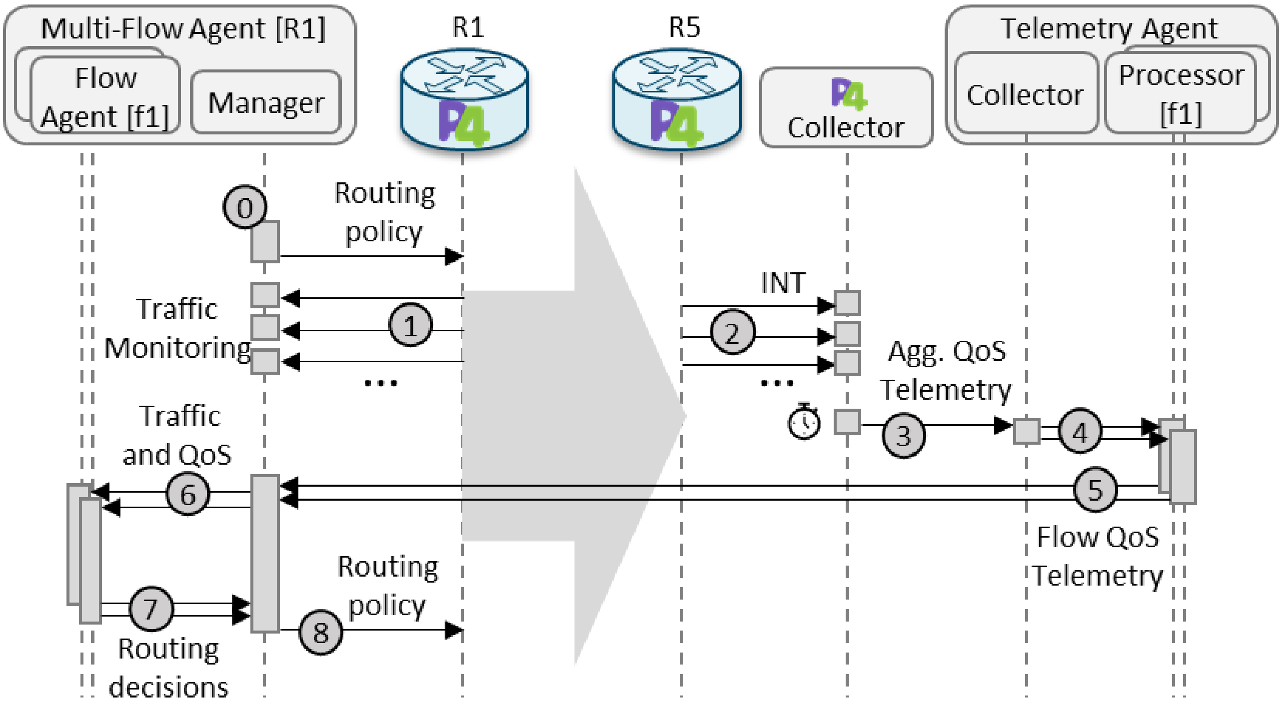


FIGURE 36. NEAR-REAL-TIME OPERATION WORKFLOW TO BE DEMONSTRATED (DEMO 4)

The illustrated workflow will be used to demonstrate different traffic scenarios, including: i) sharp traffic increase in one of the flows, leading to temporary QoS degradation (peak of delay) in both flows; and ii) gradual traffic increase in both flows. In the demonstration, autonomous and fixed routing policies will be demonstrated to show how algorithms can anticipate decision making and avoid congestion, leading to robust high QoS achievement. A Web interface will facilitate the interaction of the attendees with the system, so they can modify the type and configuration of routing policies. Moreover, a Grafana dashboard will be provided to visualize the resulting QoS performance of the flows and observe the impact of routing on flows QoS. In addition, traffic collected by the Spirent sink will be visualized for demo validation purposes.

| Planned deadline | Planned integration & validation task |
|---|---|
| Q1 – 2024 | Integration of the components:<br>• MAS<br>• P4 switches<br>• Telemetry collectors |
| Q2– 2024 | Finalize the DESIRE6G demo |

TABLE 28. DEMO 4 TIMELINE

# 5. Initial data set collection

The final section highlights the initially identified datasets that will be generated or employed in the development of the specific DESIRE6G components and PoCs.

## 5.1. ML-based orchestration and lifecycle management of network services

Instantiated network services may span multiple operators and heterogenous infrastructure, making lifecycle management in these large networks computationally complex. As in many fields, ML-based solutions can be used to quickly produce a robust action set. As described in Deliverable 3.1 [6] in Desire 6G we develop ML-based solutions for service management and orchestration, specifically targeted to: 1) the SLA decomposition into actionable objectives that can be used for resource allocation (section 5.4 in D3.1), and 2) scaling edge computing resources (Section 5.5 in D3.1). To evaluate the proposed approaches and support the reproducibility of the experimentation process, distinct synthetic datasets have been created as described in the following.

### 5.1.1. ML-based assisted SLA decomposition

To counteract the inherent complexity of managing and orchestrating services, the E2E SLA of the network service is decomposed to partial Service Level Objectives (SLOs). These SLOs are then assigned to distinct network domains (i.e., RAN, transport, core) and used for lower-level resource allocation. In DESIRE6G, we develop an SMO layer that is able to perform SLA decomposition based on historical feedback provided by each network segment on admission control of network services [7].

Therefore, the generated dataset [8] involves the partial SLOs provided to the DESIRE6G network domains (e.g., delay and throughput for RAN/transport/core) and whether each corresponding network service segment was successfully instantiated (admission control information).

#### 5.1.1.1. Attributes, variable type and a brief description

| Attributes | Variable type | Brief description |
|---|---|---|
| Requested SLOs | Float32 delay<br>Float32 throughput | Partial service level objectives per network domain e.g., (DESIRE6G site (edge/cloud), transport network) |
| Admission control decision | Boolean | Domain response on network service instantiation |

TABLE 29. ADMISSION CONTROL DATASET ATTRIBUTES

### 5.1.2. Elastic scaling

To guarantee performance and resource availability during operation, scaling mechanisms need to be deployed. Elasticity can be defined as the ability to increase and shrink resources in an autonomous manner, to adapt to workload changes. In DESIRE6G, we investigated the problem of scaling edge computing resources in the context of a Cellular Vehicular-to-Network (C-V2N) service [6].

#### 5.1.2.1. General dataset based on scaling decisions

The synthetic dataset [8] is constructed upon a real-world vehicular trace dataset. The source dataset spans from January to October 2020 and captures vehicle traffic in Turin, Italy [9]. It includes data from over 100 measuring locations distributed throughout Turin (See Figure 37). These measuring locations record the number of vehicles passing through each point at 5-minute intervals. At the end of each 5-minute interval, the dataset provides the traffic intensity - A(t) vehicles per hour - representing the number of vehicles arriving in that interval. The time-varying A(t) values serve as the arrival rates for P independent Poisson processes, where P represents the number of PoPs, as a subset of the measuring stations. These processes model vehicle arrivals at the designated PoPs during specific time intervals.

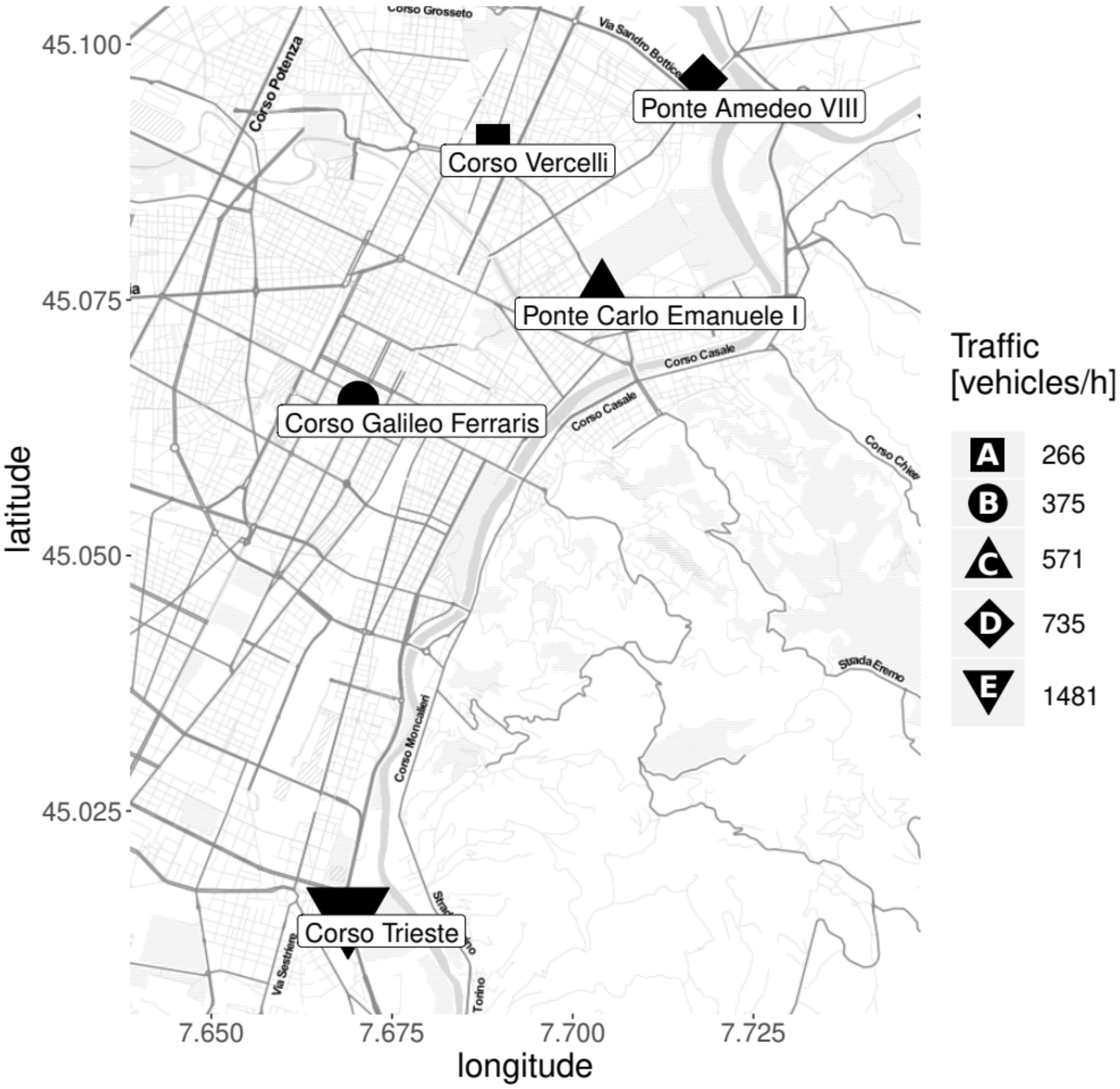


FIGURE 37. 5 POPS IN TURIN, ITALY

The synthetic dataset was generated by iterating the time-varying arrival rates, over 5 selected PoPs, following the form of $\{(t_1, p_1), (t_2, p_2), \ldots\}$, where $t_v$ represents the time of vehicle arrival at PoP $p_v$.

### 5.1.2.2. Attributes, variable type and a brief description

| Attributes | Variable type | Brief description |
|---|---|---|
| Time-series car arrivals | Int64 Arriving Time (UNIX Time)<br>String Arriving PoP | containing samples in the form of $\{(t_1, p_1), (t_2, p_2), \ldots\}$, where $t_v$ represents the time of vehicle arrival at PoP $p_v$. |

TABLE 30. ELASTIC SCALING DATASET ATTRIBUTE

## 5.2. Packet-based time series

### 5.2.1. General dataset description

This dataset focuses on the in-band telemetry collector data generated by the P4-based packet network shown in Figure 27. This dataset is important for the training operations of the multi-agent systems responsible for prediction of exceeded end-to-end latency in different segments. The collection of this data is done using synthetic traffic generation by means of a traffic generator and analyser. The data mainly refers to latency values measured by the switches, and includes other attributes such as the queueing length and the timestamps. The dataset is collected either at the P4 collector stage or at specific network point of measurement. Different data collections will be considered based on the type of telemetry utilized (e.g., standard in-band network telemetry - INT, postcard telemetry, statistical INT, INT augmented with UE info) and will be described in detail in the next deliverable. A preliminary dataset collection already available has been retrieved from a portion of the P4 network emulating a single P4 node with controlled congestion. The portion of the network is done by two P4 switches and a Juniper router, as described below.

### 5.2.2. Attributes, variable type and a brief description

The collected dataset shows the forwarding delay in a P4 switch with a physical queuing behaviour; The experiments were conducted in two scenarios. In the first scenario the experiment is run for a duration of 5 minutes and the traffic is increased in constant steps of 100Mbps every 30 seconds. While in the second scenario the experiment is run for 60 seconds with a constant traffic flow near the maximum capacity of the link (1Gbps) with a burst introduced at the 30th second of the experiment run time. The burst is of 5Mbps and a duration of 1 second. The experiments in both scenarios are repeated four times to ensure the accuracy of the collected data. The hardware used in the experiment is a Spirent SPT N4U connected through a 10 Gbps interface to a physical server running a Native In-Kernel P4 Software

Switch (NIKSS-vSwitch[7]), traffic is then looped back to the Spirent traffic generator through another 10Gbps interface. And to simulate the queuing behaviour, the P4 switch first sends the traffic to Juniper M10i router through two 1Gbps interfaces considering the load balancing of the traffic between the two links. The traffic is then returned to the P4 switch through a single 1Gbps interface between the Juniper router and the P4 switch. The setup is illustrated in Figure 38a. The two 1Gbps ports towards the juniper router ensure that they carry enough traffic to saturate the 1Gbps link towards the P4 switch while each of the works at around 50% of capacity with the proper load-balancing.

| Attributes | Variable type | Brief description |
|---|---|---|
| Forwarding Delay | Integer | Measures the time taken by a packet to cross the P4 switch, this includes the queueing delay in addition to the processing delay. |
| Timestamps | Integer | Represents the time intervals at which the measurements were taken. Reference time relative to the switch bootstrap time (ingress and egress timestamps). |
| Queue length | Integer | Number of packets currently in the egress port queue, indicating the level of congestion of the switch |

TABLE 31. PACKET-BASED TIME SERIES DATASET ATTRIBUTES

By looking at the collected data set in the first scenario (plotted in Figure 38c) (increasing steps of 100Mbps) it is possible to notice two thresholds. The first one happens at the one-minute mark (equating to 300 Mbps of traffic). And the second more obvious threshold happens at around four minutes and thirty seconds of experiment time (equating to 900Mbps) with the latency increasing by around 25% and starting to experience sharper spikes. This indicates that queuing starts to happen as the

[7] https://github.com/NIKSS-vSwitch/nikss

instantaneous traffic rate have a higher impact on the forwarding of the packets leading to congestion and queueing of the packets, and at this stage countermeasures to reduce the congestion need to be taken. The second scenario (Figure 38b), involves a one-minute experiment at near 99% link capacity (the link between Juniper and P4 switch on right hand side of figure 1) with a sudden burst of 5Mbps and a duration of one second. The burst causes congestion on the 1Gbps link and leads to the queues of the juniper router filling up. The result is an increase in the latency from around 300 microseconds to a maximum of 4 milliseconds, after the burst is over the queues start to empty and the latency decreases over a period of 10 seconds to reach its normal pre-burst levels.

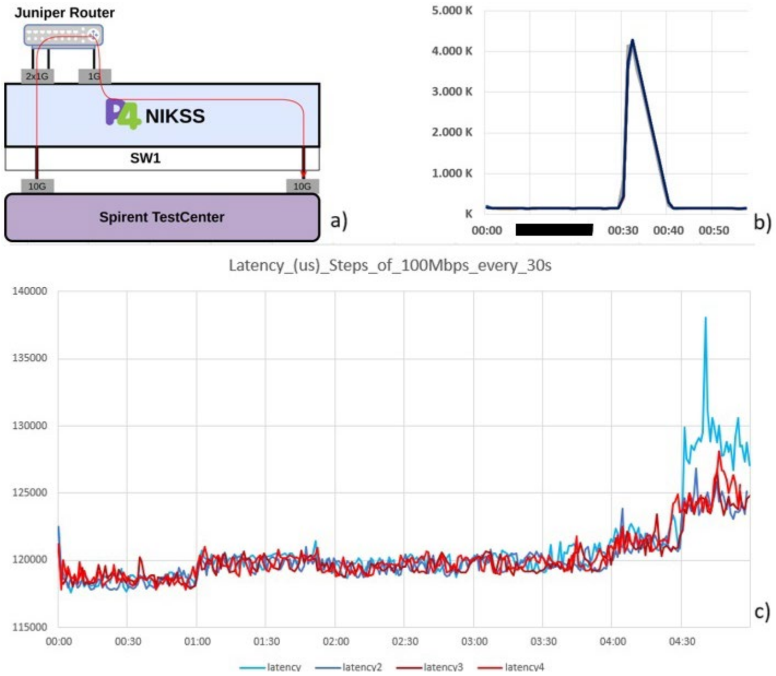


FIGURE 38. INT DATASET: LATENCY AS A FUNCTION OF THE FLOW LOAD AND CONGESTION EVENTS

## 5.3. Digital Twin application

### 5.3.1. General dataset description

This dataset focuses on the data generated by the Digital Twin system. This includes both upstream data, which consists of SLAM data from the LiDAR, data flows from the camera, odometry and joint states from the robot, as well as downstream data, which relates to the remote-control commands of the robot. The collection of this data will be captured and stored in ROS bag files, which are comprehensive files capable of recording and replaying ROS message data. These files will then be further analysed and used to optimize the system.

### 5.3.2. Attributes, variable type and a brief description

The attributes of the Digital Twin application dataset are described in detail in the table below.

| Attributes | Variable type | Brief description |
|---|---|---|
| LiDAR SLAM data | **sensor_msgs/LaserScan:**<br>• float32 angle_min<br>• float32 angle_max<br>• float32 angle_increment<br>• float32 time_increment<br>• float32 scan_time<br>• float32 range_min<br>• float32 range_max | Spatial data into the robot's environment. |
| Camera flows data | **sensor_msgs/Image:**<br>• uint32 height<br>• uint32 width<br>• string encoding | Visual data captured from the robot's onboard camera |
| Robot joint states | **sensor_msgs/JointState:**<br>• string[] name<br>• float64[] position<br>• float64[] velocity<br>• float64[] effort | Information on the position and status of the robot's joints |
| Robot odometry | **nav_msgs/Odometry:**<br>• geometry_msgs/PoseWithCovariance pose<br>• geometry_msgs/TwistWithCovariance twist | Estimation of the robot's position and velocity in free space |
| Robot high state | **unitree_legged_msgs/HighState:**<br>• IMU imu | High-level robot state, necessary to publish joint |

| | | |
|---|---|---|
| | • MotorState[20] motorState<br>• int16[4] footForce<br>• int16[4] footForceEst<br>• uint8 mode<br>• float32 progress<br>• uint8 gaitType<br>• float32 footRaiseHeight<br>• float32[3] position<br>• float32 bodyHeight<br>• float32[3] velocity<br>• float32 yawSpeed<br>• float32[4] rangeObstacle<br>• Cartesian[4] footPosition2Body<br>• Cartesian[4] footSpeed2Body | states and odometry messages on ROS topics. |
| Robot control commands | **geometry_msgs/Twist:**<br>• Vector3 linear<br>• Vector3 angular<br><br>**unitree_legged_msgs/HighCmd:**<br>• uint8 mode<br>• uint8 gaitType<br>• uint8 speedLevel<br>• float32 footRaiseHeight<br>• float32 bodyHeight<br>• float32[2] position<br>• float32[3] euler<br>• float32[2] velocity<br>float32 yawSpeed | Commands issued for the robot's movements and operations |

TABLE 32. DIGITAL TWIN APPLICATION DATASET ATTRIBUTES

## 5.4. DLT network transactions

### 5.4.1. General dataset description

The DLT dataset provides a log of all transactions taking place within the DESIRE6G DLT solution defined in WP3. As shown in Figure 39, the solution consists of two distinct networks:

- **Inter-Domain DLT:** This network facilitates interactions between different Administrative Domains (ADs) at the orchestration level, focusing on service federation aspects. In this context, an AD represents a service provider equipped with SMO infrastructure.
- **Intra-Domain DLT:** Dedicated to internal domain interactions, focusing on the DESIRE6G MAS security.

This verbose log helps for identifying any anomalies, understanding network behaviour, and ensuring the integrity of the data.

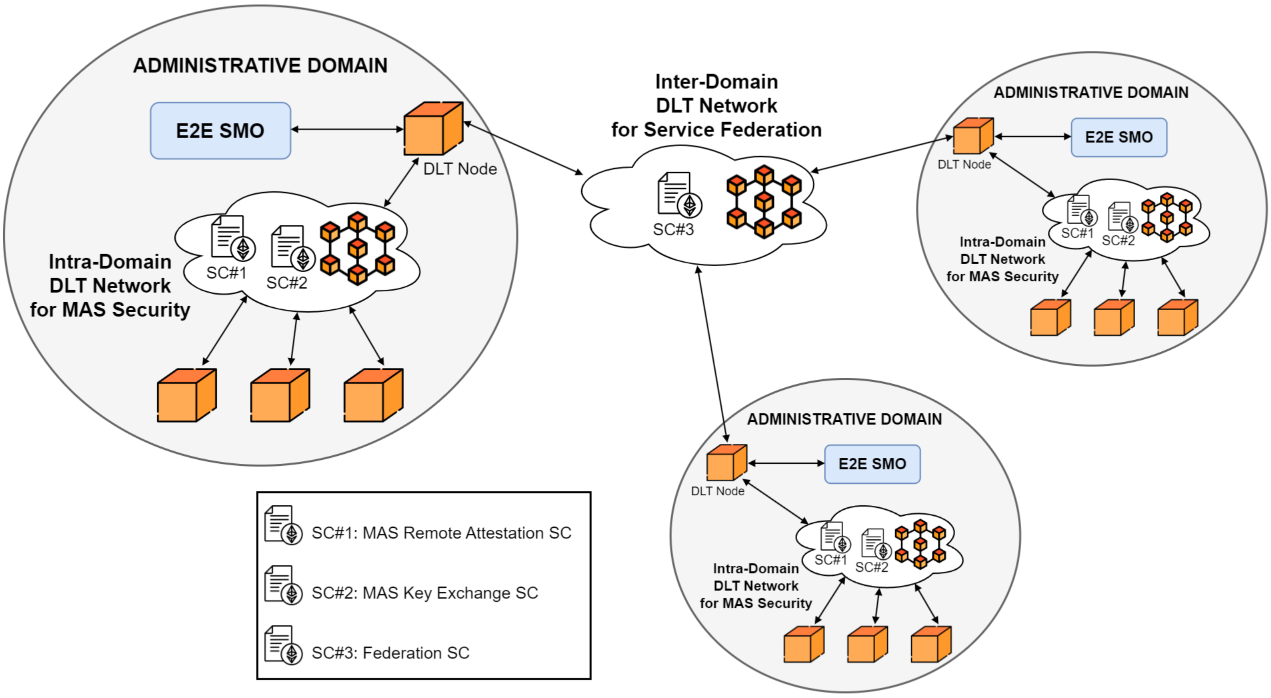


FIGURE 39. OVERVIEW OF DESIRE6G DLT SOLUTION

The primary target group for this dataset consists of partners from WP3. This includes the DLT service federation of UC3M, the DLT MAS security of UPC for VXLAN key exchanges, and the DLT MAS security approach of TSS for mutual remote agent attestation using BHT.

### 5.4.2. Attributes, variable type and a brief description

The detailed attributes of this dataset can be found in the table that follows.

| Attributes | Variable type | Brief description |
|---|---|---|
| Transaction hash | Text/String | Hash of the transaction, serving as its identifier |
| Timestamp | DateTime | Time at which the transaction occurred |
| Transaction Data | Text/String | Specific details and content of the transaction |
| Transaction Status | Text/String | Current state of the transaction (e.g., completed, pending) |

TABLE 33. DLT NETWORK TRANSACTIONS DATASET ATTRIBUTES

## 5.5. RIS application at EDGE nodes

The dataset is related to RISs (reflecting intelligent surfaces). Loosely speaking, RIS is specialized structure in wireless communication that uses reconfigurable elements to dynamically change the channel between transmitters and receivers of the system, see Figure 40. This is accomplished by changing signal propagation between the transmitters and respective receivers by intelligently configuring the RIS elements.

The RIS application dataset is considered mainly to demonstrate the robustness of the proposed federated learning (FL) algorithms at the edge for heterogeneous data. The RIS data set is created as a byproduct of the initial experiments that were conducted to evaluate the algorithms. The proposed FL algorithms are unrestricted in that they do not impose limitations based on the dataset, allowing for consideration of any labeled dataset. However, in the project, the dataset is simply used to showcase robust and privacy-preserving AI at the EDGE site.

### 5.5.1. General dataset description

The dataset contains Channel State Information of RIS assisted wireless environments, together with configuration indexes as labels. Specifically, it provides realizations of the channels $h$ and $g$ from a single antenna transmitter to the RIS and from the RIS to the single antenna receiver, respectively. Each realization is labelled by the optimal RIS configuration that maximizes the receiver's rate.

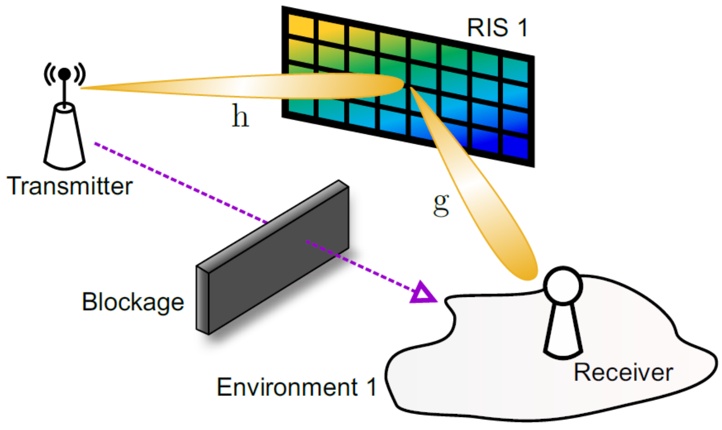


FIGURE 40. RIS ENVIRONMENTS

The target group for this dataset can typically be engineers and researchers from wireless communications, signal processing, and machine learning. The rate-maximization by tuning the reflecting elements of RISs in a RIS setting is a combinatorial optimization problem, and therefore to compute an approximate configuration for the reflecting elements in an efficient manner is of crucial importance in practice. In this respect, the above dataset can be used to train a neural network for efficiently computing a reasonable configuration for reflecting elements.

The dataset consists of 50,000 samples spitted heterogeneously across 4 RIS environments indexed by $r \in \{1,2,3,4\}$.

### 5.5.2. Attributes, variable type and a brief description

The detailed attributes of the dataset documented in Table 34 below.

| Attributes | Variable type | Brief description |
|---|---|---|
| Channel: Transmitter-RIS ($h^r$) | Complex vectors of length $N^r$ | Channel between a single antenna transmitter and the RIS in the RIS environment $r$. |
| Chanel: RIS-Receiver ($g^r$) | Complex vectors of length $N^r$ | Channel between the RIS and a single antenna receiver in the RIS environment $r$. |
| Best configuration of reflecting elements $y^r$ | Integer to represent the configuration index | Set of angles of the reflecting elements is partitioned into a finite number of clusters. Each cluster has a representative configuration for the reflecting elements. the configuration index is simply the best cluster index. |

TABLE 34. HETEROGENEOUS RIS ENVIRONMENTS DATASET ATTRIBUTES

# 6. Conclusion

The work conducted within WP5 so far in the DESIRE6G project has allowed us to select the most promising technologies from DESIRE6G WP3 and WP4 to develop Proof-of-Concepts supporting the various use cases identified in WP1. In this deliverable, we have presented three individual real-world test sites to experiment with the various PoCs and report meaningful performance measures highlighting the value proposition of DESIRE6G solution.

Two global real-world testbeds are defined namely (1) 5TONIC and (2) ARNO. The testbeds provide a solid foundation for the assessment and evaluation of the technology solution being developed within DESIRE6G project. We then presented the two amin PoCs (at the 12-month stage of the project), namely: Digital Twin and the Intelligent and resilient VR/AR applications with perceived zero latency. We presented the physical and logical architectures of each PoCs, as well as detailed specifications for the integration and deployment of the individual components comprising each PoC. Integration and validation timelines were also presented. This deliverable also provides detailed demo scenarios and demo execution plans of the PoCs under development in the DESIRE6G project. Finally, this deliverable presents the identified datasets that will be collected and/or used in the project from the network infrastructure but also form the applications.

In the upcoming year of the project, the different technical components of the DESIRE6G solution and the two PoCs presented in this deliverable will be further elaborated, tested, evaluated, and validated through various performance metrics.

Work Package 5 has set its eyes in 2024 on the upcoming mid-term trials scheduled in September 2024 to experiment, validate, and demonstrate the PoCs with a subset of components presented in this deliverable. For 2024, two major events have been targeted, namely (1) Optical Fiber Communication (OFC) conference 2024 in San Diego, California, USA, where individual DESIRE 6G components (e.g., persuasive monitoring, DLT-based MAS security) will be presented and (2) the EuCNC & 6G Summit 2024 in Antwerp Belgium where a preliminary version of the PoCs that integrate different components of the DESIRE 6G system will be demonstrated.

# 7. References


[1] V. Lefebvre, A. Nanos and C. Papagianni, *D3.1: Initial report on the intelligent and secure management, orchestration, and control platform,* Zenodo, 2023.

[2] A. Sgambelluri, R. Gonzalez and A. Mihály, *D4.1: Initial report on the DESIRE6G unified programmable data plane layer,* Zenodo, 2023.

[3] S. Hanks, T. Li, D. Farinacci and P. Traina, *Generic Routing Encapsulation (GRE),* RFC Editor, 1994.

[4] R. Lopez Da Silva, M. Groshev, S. Ertas and C. Papagianni, *D2.1: Definition of Use Cases, Service Requirements and KPIs/KVIs,* Zenodo, 2023.

[5] D. De Vleeschauwer, C. Papagianni and A. Walid, "Decomposing SLAs for network slicing," *IEEE Communications Letters,* vol. 25, p. 950–954, 2020.

[6] M. Saimler and others, *D3.1: Initial Report on the Intelligent and Secure Management, Orchestration, and Control Platform,* 2023.

[7] C. S.-H. Hsu, D. De Vleeschauwer and C. Papagianni, "SLA Decomposition for Network Slicing: A Deep Neural Network Approach," *IEEE Networking Letters,* 2023.

[8] DESIRE6G Datasets Repository, 2023, URL: https://gitlab.netcom.it.uc3m.es/desire6g/mainrepo/-/tree/main/Datasets

[9] *Opendata Torino Traffic Data,* 2023, URL: https://smartcityweb.net/progetti/progetto-s-dot-i-mo-dot-ne

# 8. Annexes

This section provides detailed reference specifications for the Hardware and Software components used in the different testbeds.

## 8.1. List of components used in the 5TONIC testbed

| Component | Type | Operational frequency | Main features |
|---|---|---|---|
| Ericsson AIR 6488 | Indoor AAS | B41K, B42 series, B43 | • High order spatial multiplexing and multi-user MIMO<br>• NR-TDD<br>• 64TX/64RX branches<br>• eCPRI |
| OneCell RP-A2014 | Indoor Radio Point | B10, B12, B13, B17, B2, B25, B4 | • LTE MIMO 2x2 |
| Ericsson Dot 5G 4479 | Indoor Radio Dot | Band n78 | • 4x4 MIMO<br>• Carrier aggregation<br>• Network slicing<br>• 256 QAM in TDD/FDD mid-bands |
| Ericsson AIR 6468 | Outdoor AAS | - | • High gain adaptative beamforming<br>• High order spacial multiplexing and multi-user MIMO<br>• LTE TDD<br>• 64TX/64RX branches |
| Ericsson 8823 RRU | Outdoor Remote Radio Unit | Band n78 | • WCDMA<br>• LTE FDD<br>• 4x4 MIMO |
| Ericsson 2203 RRU | Indoor & Outdoor Remote Radio Unit | 3GPP Bands B1, B3, B7, B8, B66A, B5, B2/B25 | • WCDMA<br>• LTE FDD<br>• 2TX/2RX branches<br>• 4x4 MIMO |

| | | | |
|---|---|---|---|
| SIM8200EA-M2 5G HAT for Raspberry Pi (x7) | Communication Module | 3GPP Bands | • 5G/4G/3G Support<br>• Snapdragon X55<br>• Multi-Mode<br>• Multi-Band |
| Wi-Fi 6 AP (x1 ASUS AX3000, x1 Commscope Wi-Fi 6E) | Access Point | 6 GHz (Wi-Fi 6 / Wi-Fi 6E) | • High-efficiency Wi-Fi for both indoor and outdoor environments |

TABLE 35: REFERENCE SPECIFICATIONS OF RAN COMPONENTS IN 5TONIC TESTBED

| Component | Processor | Memory | Disk | Networking |
|---|---|---|---|---|
| Dell PowerEdge R430 v3 (x3) | Intel Xeon CPU E5-2609 v3 @ 1.90GHz | 16GB DDR4 2133 MHz | 1TB (RAID 1) | 2x 10Gb (X540-AT2), 4x1Gb (NetXtreme BCM5720) Ethernet |
| Dell PowerEdge R430 v4 (x3) | Intel Xeon CPU E5-2609 v4 @ 1.70GHz | 16GB DDR4 2133 MHz | 4TB | 2x 10Gb (Intel X520), 4x1Gb (Intel I250), 4x1Gb (Broadcom BCM5720) |
| Dell PowerEdge R630 (x2) | Intel Xeon E5-2620 v4 (16 cores, 32 threads) | 128GB RAM | 1TB | 2x X550 10Gbps, 4xI350 1Gbps |
| Dell PowerEdge R630 Dual (x1) | 2x Intel Xeon E5-2620 v4 (16 cores, 32 threads) | 128GB RAM | 1TB | 4x X520 10Gbps, 4xI350 1Gbps |
| Dell PowerEdge R730 (x1) | Intel Xeon CPU E5-2609 v3 @ 1.90GHz | 192GB (12x16) DDR4 2133 MHz | 1TB (RAID 1) + 400GB SSD | 4x 10Gb (NetXtreme II BCM57810), 4x1Gb (NetXtreme BCM5720) Ethernet, 8x10Gb (X710) SFP+ |

TABLE 36. 5TONIC DATA CENTER SERVER SPECIFICATIONS

## 8.2. List of components used in the ARNO testbed

**Optical Transport**

| **Disaggregated whiteboxes** |
|---|
| 2 OpenConfig-enabled Fujitsu T600 transponder (OpenConfig agent) |
| 1 Edgecore AS9726 with 100-40ZR coherent pluggables (SONiC operating system) |
| 1 Lumentum ROADM-20 (OpenROADM agent) |
| **Commercial ROADM** |
| Ericsson SPO-1400, 100Gbps PM-QPSK coherent |
| **Optical Line System (OLS)** |
| Finisar WSS, EDFA, 4+ 80km fiber spans (OpenROADM and OpenConfig agents) |

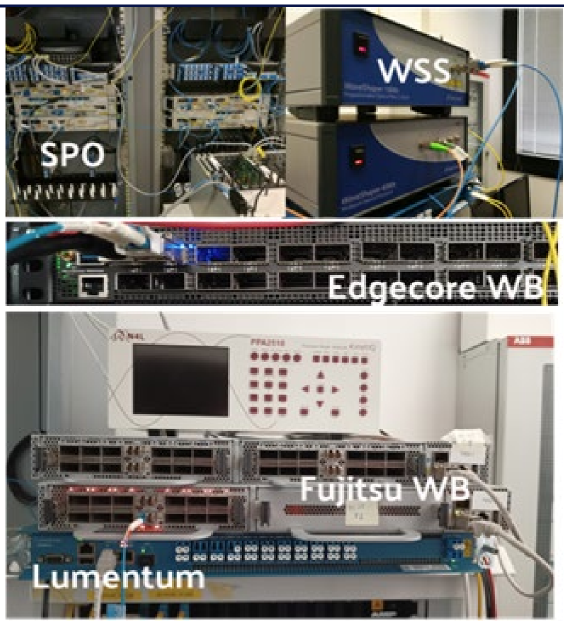


FIGURE 41. ARNO: OPTICAL TRANSPORT SEGMENT EQUIPMENTS

**Far Edge / Cloud Resources**

| **Dell Servers** |
|---|
| 2 DELL EMC PowerEdge 740RX |
| 2 DELL EMC PowerEdge R740 |
| 2 DELL EMC PowerEdge R7515 |
| 3 DELL EMC PowerEdge R760 (2 fully dedicated to DESIRE6G development) |
| **Connectivity** |
| Up to 2x 100 GbE (dual port), Mellanox ConnectX5 NIC) |
| **Resources** |

| | |
|---|---|
| Equipped with GPU (NVIDIA A16, NVIDIA Tesla or NVIDIA V100) | |
| **Experimental platforms** | |
| BRAINE Edge Micro Data Center with modular design (CPU, GPU boards interconnected via Mellanox switch), innovative liquid cooling design |   FIGURE 42. ARNO: CLOUD/EDGE SEGMENT DEVICES |

**Far Edge Platforms**

| | |
|---|---|
| **Development Kits** | |
| 1 NVIDIA Jetson AGX Xavier SDK | |
| 1 NVIDIA ORIN 64GB |   FIGURE 43. ARNO: FAR EDGE SEGMENT DEVICES |

**Cloud Networking Devices**

| | |
|---|---|
| **Aggregation Switches** | |
| 2 Mellanox Spectrum1 SN2010 with 1,10,100GbE ports (Onyx+SONiC SO, limited P4 programmability with proprietary Mellanox P4 compiler) | |

| |
|---|
| 1 Mellanox Spectrum2 SN3420 60 1GbE->100GbE ports (Onyx OS) |


FIGURE 44. ARNO: AGGREGATION SWITCHES

**Programmable Networking**

| **Bare metal Hardware** |
|---|
| 1 APS Tofino switch BF2556X 56 ports (8x100 GbE) |
| 6 NetFPGA Sume Virtex-7 (4x10GbE) |
| **Software switches** |
| 2 DELL servers dedicated to P4 switch softwares (4 100GbE interfaces). Available backends: BMv2, eBPF NIKSS) |
| **DPU** |
| 1 NVIDIA Bluefield-1 |
| 2 NVIDIA Bluefield-2 dual port 100GbE |
| 2 NVIDIA Bluefield-2 dual port 25GbE |
| 1 NVIDIA Bluefield-2 dual port 100GbE + embedded GPU |
| **SmartNICs** |
| 2 Xilinx Alveo SN1000 dual port 100 GbE (P4 programmable) |


FIGURE 45. ARNO: PROGRAMMABLE DATA PLANE HARDWARE

**Access / 5G RAN**

| |
|---|
| **Radio Hardware** |
| 2 RRU Ettus X310 |
| 2 RRU Ettus B210 |
| 2 Quectel 5G + 2 Evaluation Board Quectel |
| **FPGA Hardware** |
| 2 Intel DE10-Pro Stratix® 10 GX/SX FP |
| 2 Intel® FPGA Programmable Acceleration Card (PAC) N3000 |
| 2 Intel® PAC with Intel® Arria® 10 GX FPGA |
| **Radio Software (evaluation in progress)** |
| OAI 5G RAN and FlexRIC |
| SRSran |

FIGURE 46. ARNO: 5G RAN EQUIPMENT

**Traffic generators and analyzers**

| |
|---|
| **Hardware** |
| Spirent N4U (GbE and 10GbE interfaces, aggregated and stream traffic, custom packet configurator, analyzer (packet loss, throughput, latency, frame errors) |
| Viavi ONA-1000 (100GbE and 400 GbE testing ports, QSFP-DD and SFP-DD interfaces, L2/L3 stream configurations |
| **Software** |
| Cisco Trex (https: //trex-tgn.cisco.com/) |

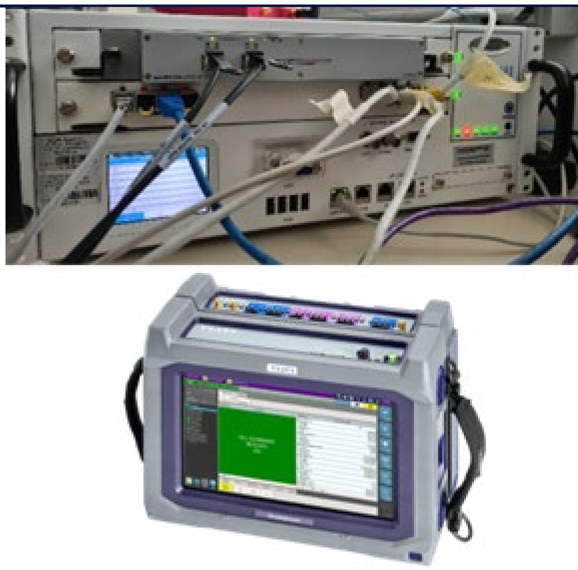

FIGURE 47. ARNO: TRAFFIC GENERATORS EQUIPMENT

**Terminals and devices for verticals**

| |
|---|
| Small factor UAVs (DJI Mini) (not suitable for Demo1) |
| 10 X-86 boards equipped with dual Wi-Fi antennas and GPS |
| 2 SuperDroid Mobile 4WD Robot Development Platform – IG3 with Wi-Fi and smart camera |
| 1 PupilLabs Core Smart Glass with pupil tracking platform |
| 5 smart D-Link webcams |
| 1 Robotis OP3 humanoid robot platform |
| To be purchased: Headsets for Demo 1 (Meta Quest 2), industrial UAV |

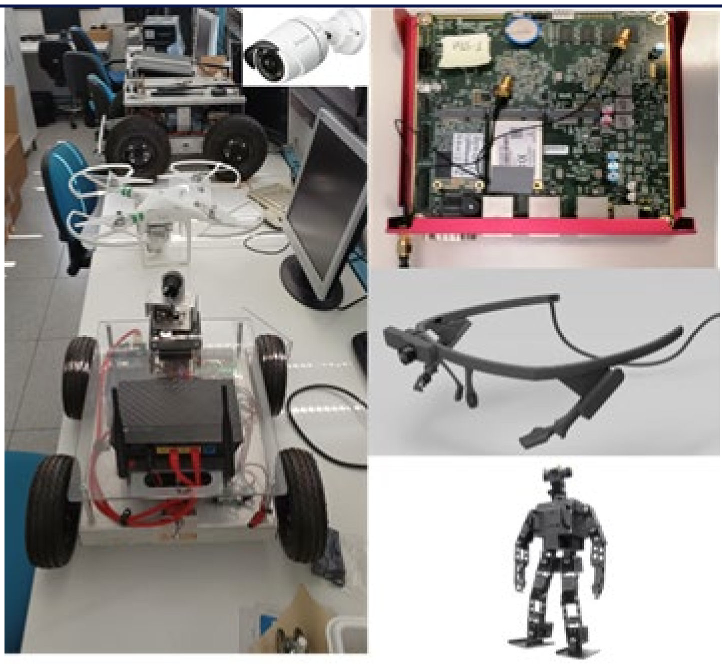

FIGURE 48. ARNO: MOBILITY END USER TERMINALS

## 8.3. List of components used in the Multi-Agent System testbed

| Component | Processor | Memory | Disk | Networking |
|---|---|---|---|---|
| 3x Compute Server Type A | Intel(R) Core (TM) i9-9980XE CPU @ 3.00GHz (36 cores) | 128GB DDR4 2400 MHz | 256GB+ 1TB NVMe SSDs | 2x 1Gb Ethernet |
| 6x Compute Server Type B | Intel(R) Core (TM) i9-13900K CPU @ 3.00GHz (32 cores) | 128GB DDR4 3200 MHz | 512GB+ 1TB NVMe SSDs | 2x 1Gb Ethernet |
| 12x Compute Server Type C | Intel(R) Core (TM) i7-4770k CPU @ 3.40GHz (8 cores) | 16GB DDR3 1600Mhz | 512GB SSDs | 2x 1Gb Ethernet |

TABLE 37. GCO-UPC DATA CENTER SERVER SPECIFICATIONS

## 8.4. List of components used in the Technology & Automation LAB testbed

| Component | Main features |
|---|---|
| TSN switches (x2) | • Time-aware traffic shaping: 802.1Qbv (on TX of all 8 Ethernet ports of the design).<br>• Preemption: 802.1Qbu & 802.3br (on TX of all 8 Ethernet ports of the design).<br>• FRER redundancy: 802.1CB<br>• Frame-based ingress policing: 802.1Qci<br>• TSN flow identification, routing, and VLAN tagging: 802.1Q (on all 8 Ethernet ports of the design). |
| PowerEdge R730 server (x1) | • 2x Intel(R) Xeon(R) CPU E5-2695 v3<br>• 8x 16GiB DIMM DDR4<br>• 3TB storage in RAID system (PERC H730 Mini)<br>• 14x 10G fibre ports |
| Edgecore CSP-7550 switches (x2) | • One CSP-7550 with FPGA (Intel Stratix MX)<br>• 2x Intel Xeon Gold 5218 (16-Core, 2.3 GHz)<br>• 256GB (32GB x 8)<br>• 240GB M.2 SSD<br>• Tofino BFN-T10-064Q-B0<br>• 100G: 32 x 100G QSFP28 |

TABLE 38. TECHNOLOGY & AUTOMATION LAB SPECIFICATIONS

## 8.5. List of components used in the NPT-Budapest testbed

| Component | Location | Main features |
|---|---|---|
| Epyc servers (Type-A) (x4) | ERI-HU | • AMD Epyc 7402P 24C 2.8GHz<br>• 128GB RAM<br>• 2x100GbE Mellanox ConnectX-6 Dx NIC |
| Epyc servers (Type-B) (x3) | ERI-HU | • AMD Epyc 7402P 24C 2.8GHz<br>• 128GB RAM<br>• 2x100GbE NVIDIA Bluefield-2 DPU,<br>• one of the servers is equipped with a NVIDIA Quadro GPU with GPUDirectRDMA support |
| Intel servers (x3) | ERI-HU | • Intel Xeon E-2246G 6C 3.6GHz<br>• 64GB RAM<br>• 2x40GbE Intel XL710 NICs |
| Delta AG9064 switch (x1) | ERI-HU | • Intel Tofino-1 6.4 TBPS P4 programmable switch<br>• 64 GbE ports |
| Management switch (x1) | ERI-HU | • 24x1GbE ports |
| Intel Xeon server (x1) | ELTE (remote site) | • dual socket Intel Xeon Silver 4110, 2.10GHz<br>• 128GB RAM, 4TB disk<br>• 2x10GbE Intel NICs<br>• Used for network simulations (NS-3, Netbench), development VMs with Intel Tofino SDE, FPGA SDE, etc.<br>• Located in Martonvásár, connected to the testbed via VPN |
| AMD server (Type-1) (x2) | ELTE | • AMD Ryzen Threadripper 1900X 8C<br>• 128GB RAM, 4 TB disk<br>• 2x10GbE Intel NICs<br>• 2x100GbE Chelsio NICs |
| AMD server (Type-2) (x1) | ELTE | • AMD Ryzen Threadripper 3970X 32C<br>• 64GB RAM, 4 TB disk |

| | | |
|---|---|---|
| | | • 2x100GbE Mellanox ConnectX-5<br>• NVIDIA Quadro GPU<br>• Silcom N5013 (2x100GbE, Intel Stratix10 DX FPGA-based DPU) |
| Stordis/APS Networks BF2556-1T switch (x1) | ELTE | • Intel Tofino-1, 2 TBPS P4 programmable switch<br>• 32 ports (10-25-100 GbE ports) |
| UfiSpace S9180-32X-F-PDU-V1 switch (x1) | ELTE | • Intel Tofino-1, 3.2 TBPS P4 programmable switch<br>• 32x100GbE ports |
| PcEngines APU 4D4 (x2) | ELTE | • AMD Embedded G series GX-412TC, 1 GHz quad Jaguar core<br>• 4GB RAM<br>• 4x1GbE ports (Intel i211AT)<br>• Optionally connected to the management switch in the testbed |
| Management switch (x1) | ELTE | • 16x1GbE ports |

TABLE 39. BUDAPEST TESTBEDS FOR NETWORK PROGRAMMABILITY