
ISG ZSM PoC Proposal

1 PoC Project Details

1.1 PoC Project

PoC Number: 19	
PoC Project Name:	AI-enabled predictive closed loops: network slice automation and performance optimization for collaborative robots
PoC Project Host:	Nextworks
Short Description:	<p>This PoC will showcase a zero-touch network and service management solution, capable to deploy virtualized AI-enabled closed loop functions in support of prediction of network slice performance degradation in collaborative robots application scenarios. The PoC targets the AI-enabled and automated operation of vertical services and network slices through zero-touch closed loops, with deployment of services for prioritized traffic (URLLC) in support of collaborative robot operations. The goal is to validate the zero-touch deployment and operation of virtualized closed loop functions, integrated with an MLOps framework for training and deployment of ML models, that make use of ML-enabled analytics (for prediction of network slice performance) and ML-enabled decision (for network slice congestion identification and remediation) to guarantee ultra low latency communications for collaborative robots.</p> <p>This PoC is developed using the network and service management solutions, including an MLOps platform for lifecycle management of AI/ML models, developed in the European Commission (EC) funded project, under the Horizon Europe programme, ADROIT6G [1]. The implementation of these solutions is aligned with the ZSM standards. More specifically, the design architecture of the use case follows the automation principles of ZSM 003, while the AI-driven closed loop based automation, which integrates a closed loop governance and an MLOps platform, is aligned with the ZSM 009-1, ZSM 009-2 and ZSM 009-3.</p>

1.2 PoC Team Members

	Organisation name	ISG ZSM participant (yes/no)	Contact (Email)	PoC Point of Contact (*)	Role (**)	PoC Components
1	Nextworks	Yes	Giacomo Bernini g.bernini@nextworks.it Matteo Ravalli m.ravalli@nextworks.it	X	Supplier	i) Architectural design of PoC, ii) Service and resource orchestration, Closed loop management, MLOps platform components.
2	Orange Romania	No	Catalin Brezeanu catalin.brezeanu@orange.com Marius Iordache marius.iordache@orange.com		Network/Service Provider	i) Use case definition, ii) architectural design of PoC, iii) Testbed infrastructure and 5G network components
3	CAFA Tech	No	Märten Rannu marten.rannu@cafatech.com Tanel Järvet tanel.jarvet@cafatech.com		Supplier	i) Use case definition, ii) architectural design of PoC, iii) Collaborative Robots applications
4	Industrial System Institute	Yes	Christos Verikoukis cveri@isi.gr		Research Center	i) Use case definition, ii) architectural design of PoC, iii) support on AI/ML models for AI-enabled closed loops
(*) Identify the PoC Point of Contact with an X. (**) The Role will be network/service provider, supplier, or other (universities, research centers, test labs, Open Source projects, integrators, etc...).						

All the PoC Team members listed above declare that the information in this proposal is conformant to their plans at this date and commit to inform ETSI timely in case of changes in the PoC Team, scope or timeline.

1.3 PoC Project Scope

1.3.1 PoC Topics

PoC Topics identified in this clause need to be taken for the PoC Topic List identified by ISG ZSM and publicly available in the ZSM WIKI. PoC Teams addressing these topics commit to submit the expected contributions in a timely manner.

PoC Topic Code	PoC Topic Description	Related WI	Expected Contribution	Target Date
3	Intent-driven Closed-Loop automation	ZSM-009-1, ZSM-009-2, ZSM-009-3	<ul style="list-style-type: none"> Demonstration of closed loop governance functionalities for automated and on-demand deployment and operation of 	June-July 2026

			<p>virtualized monitoring, analytics, decision and execution closed loop functions for predictive optimization of network slices, aligned with ZSM 009-1 and ZSM 009-2 principles</p> <ul style="list-style-type: none"> • Demonstration of AI-enabled closed loops, with AI/ML enabled decision and analytics functions, aligned with ZSM 009-3 principles • Demonstration of integrated closed loop governance and MLOps platform for on-demand selection and deployment of AI/ML models in support of AI-enabled closed loop functions. 	
--	--	--	--	--

1.3.2 Other topics in scope

List here any additional topic for which the PoC plans to provide input/feedback to the ISG ZSM.

PoC Topic Code	PoC Topic Description	Related WG/WI	Expected Contribution	Target Date
2	Automation in Multi-Stakeholder Ecosystem	ZSM-001, ZSM-002, ZSM-003, ZSM-005	<ul style="list-style-type: none"> • Demonstration of automated end-to-end vertical service orchestration, with integrated network slice management and resource orchestration • Demonstration of network slice management and orchestration capabilities, aligned with ZSM-003 • Demonstration of automated management of AI/ML models lifecycle (focusing on training and inference ML tasks) through a MLOps platform providing domain intelligence capabilities in support of closed loop automation as per ZSM-002 	June-July 2026

1.4 PoC Project Milestones

PoC Milestone	Milestone description	Target Date	Additional Info
P.S	PoC Project Start	February 2026	
P.P	PoC Proposal submission	March 2026	Submission to ZSM PoC Management Team for approval.
P.M1	PoC internal milestone 1: components deployment and initial integration	April 2022	Baseline PoC components deployed and integrated in the Orange Romania testbed. Preparation of 5G network setup. Identification of network metrics and KPIs to be collected for AI/ML models training
P.M2	PoC internal milestone 2: implementation of AI-enabled closed loop components	May 2022	Closed loop functions for networks slice performance prediction and optimization ready. AI/ML models ready and trained.
P.M3	PoC internal milestone 4: PoC execution	May 2022	All PoC components integrated, including management and orchestration stack, closed loop governance and closed loops, MLOps platform, vertical applications, and ready for evaluation.
P.D1	PoC demo #1	June 2026	Showcasing PoC results at 2026 EuCNC & 6G Summit conference.
P.D2	PoC demo #2	July 2026	Showcasing Final PoC results at 2026 IEEE MeditCom conference.
P.R	PoC Report	July 2026	Final feedback and PoC report document for discussion within ETSI ZSM community
P.E	PoC Project End	July 2026	

NOTE: Milestones need to be entered in chronological order.

1.5 Additional Details

More information on the objectives, use cases and deliverables from the ADROIT6G project can be found in the respective webpage [1].

2 PoC Technical Details

2.1 PoC Overview

This PoC aims at demonstrating the AI-driven and automated operation of vertical services and network slices through zero-touch AI-enabled closed loops. The goal is to deploy and operate network slices for ultra-low latency prioritized traffic and satisfy the performance requirements of vertical applications executed at the extreme-edge and edge of the network through AI-enabled closed loops.

This is achieved through a complete zero-touch network and service management solution, capable to orchestrate end-to-end vertical services and related applications on top of extreme-edge/edge cloud-native infrastructures, with seamless integration of network slice management capabilities to guarantee ultra-low latency network performances for collaborative robot operations. A ZSM compliant closed loop automation solution, comprised by a closed loop governance framework for deployment and operation of AI-enabled closed loops is used to introduce full automation in network and service management.

The PoC will showcase these zero-touch network and service automation capabilities with the closed loop governance of AI-enabled closed loop in a collaborative robot scenario. In particular, the PoC emulates an industrial collaborative robot scenario with multiple robots. The scenario covers robots that need to coordinate actions with each other in an industrial

site, in which different types of AGVs need to sync and coordinate their movements, with and without human interaction in joint tasks, lifting and unloading/loading material. In addition to the coordination among the interacting entities, robot application process data amongst involved entities needs to be exchanged, requiring network reliability, and ultra-low latency performances during all steps. The PoC application emulate various types of communications of the robots, which include on-board camera video streams, other stationary cameras video streams, sensor data flows and command and control operations. The PoC AI-enabled closed loop automation solution is adopted to guarantee network slice performances for these emulated robots application data and communications.

The zero-touch network and service management solution with AI-enabled operational closed loops, and the emulated robot applications are deployed and validated into a production-grade and 3GPP compliant 5G network infrastructure provided by Orange Romania in their Bucharest lab. This allows to validate and demonstrate the PoC solutions into a high TRL infrastructure.

2.2 PoC Architecture

2.2.1 PoC Infrastructure

The PoC leverages the infrastructure from the ADROIT6G project [1], and specifically the lab testbed of Orange Romania and available in Bucharest. This testbed is part of the Orange Romania 5G Laboratory testbed infrastructure (<https://5glab.orange.ro/en>) [2], deployed in a distributed context, in Bucharest and Iasi Laboratories.

The **Bucharest 5G Lab**, which is used in this PoC, implements an indoor full 5G Standalone (SA) infrastructure, comprising 5G RAN, Core, Edge Computing, advanced Software Defined Networking (SDN) and IPFABRIC elements in the associated Datacentre. The overall architecture of the testbed is detailed in Figure 1.

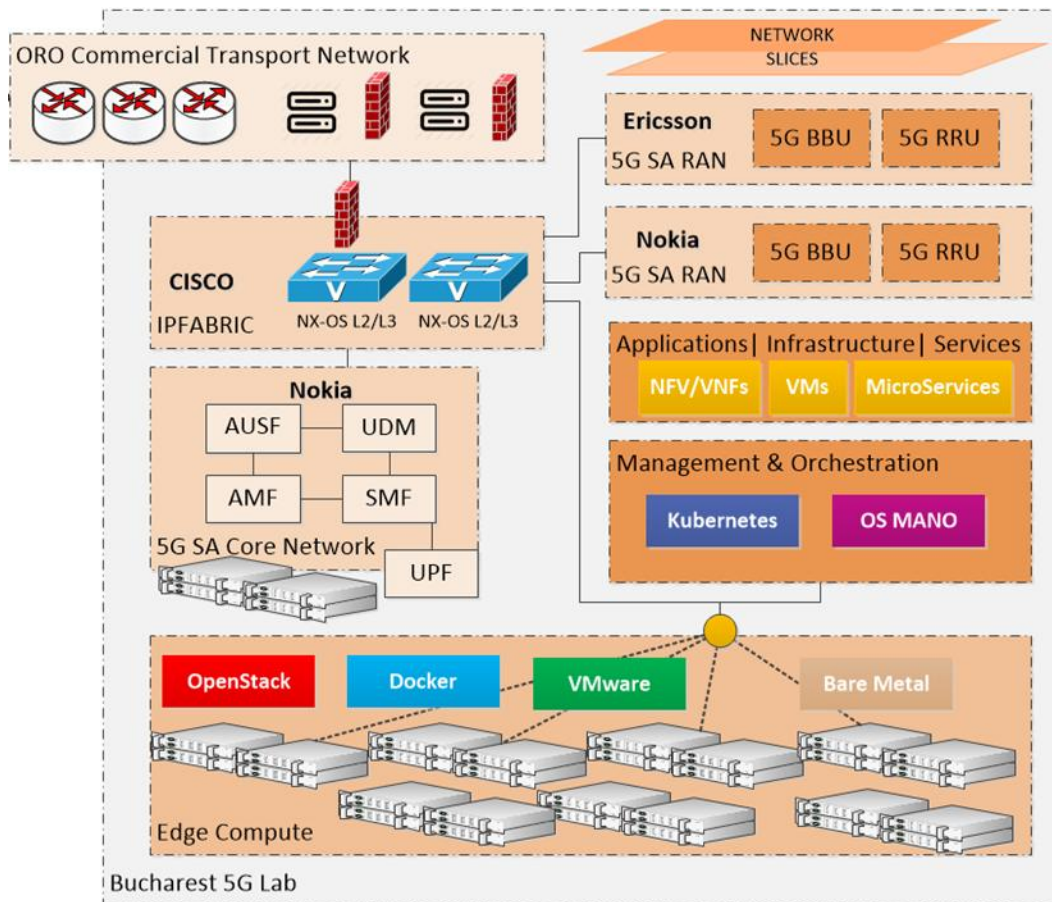


Figure 1: Orange Romania Bucharest 5G Lab Architecture

The testbed is based on virtualized 5G network solution, with a 5G SA Core architecture that uses the 5G SA Compact Mobility Unit (CMU) Core Network solution from Nokia, running on a virtualized infrastructure in the lab. Three main components are integrated, the Cloud Mobile Gateway (CMG), the Cloud Mobility Manager (CMM), and the Authentication and Policy Control (APC).

The 5G Labs 5G SA RAN architecture follows the 3GPP Option 2, N78 100MHz available spectrum; in this approach the RAN consists of only gNBs that are connected to the 5G Core network. The gNBs connect to AMF over the NG-C interface for control plane signalling and to UPF over the NG-U interface for user plane data transfer. gNBs provide the NR (New Radio) air interface access for the UEs. The deployed 5G SA RAN components also include support for network slicing, leveraging on priority management with differentiated 5G QoS flows, prioritized UE TCP sessions and 5G QoS flows to DRB mapping – in this approach the network maps different IP flows with the same QoS requirements into the same QoS flow.

Connected to the 5G Core Network and RAN infrastructure, through the IPFABRIC transport network, is the 5G Labs edge computing facility. This facility consists of 25+ compute servers, totalizing in 3000 vCPU, 15 TB of RAM, 100 TB of storage. For advanced analytics and data processing, the lab is equipped with 4 GPUs (64 GB VRAM), integrated in the computing cluster. Beyond this, for the purpose of the PoC, the testbed has been equipped with two Intel NUC 12 Pro mini-PCs, connected to the 5G SA network through dedicated 5G modems, acting as extreme-edge devices, where the robot-side of the PoC emulated robot applications can be deployed and executed. The 5G Lab's edge computing ecosystem supports multiple types of applications, ranging from bare metal solutions to virtualized or containerized variants. In particular, the facility integrates an OpenStack deployment, as cloud operating system that controls large pools of compute, storage, and networking resources throughout the lab edge datacenter, providing standard IaaS functionality, orchestration, service management and user applications deployment, Cloud Infrastructure for Virtual Machines, and Containers. The facility also integrates Kubernetes for automated deployment and orchestration of cloud-native applications across the testbed extreme-edge and edge computing infrastructure.

The testbed implements two different network slices based on 3GPP NSSAIs parameters, providing eMBB with throughput up to ~1.5Gbps for DL and ~150Mbps for UL and URLLC with a one-way delay of <4ms. At the RAN level the slicing performance is ensured by leveraging gNB-specific 5G Quality Indicators (5GQIs) functionality for proper slice implementation (Uplink Preallocation, SR Period Configuration). The network slicing is implemented based on 3GPP-defined parameters; the S-NSSAI identifies network slices in NG-RAN and 5GC and is composed of a Slice Service Type (SST) and Slice Differentiator (SD), SST identifying a type of network slice with specific capabilities and characteristics, while SD pointing out different network slices belonging to the same type. Furthermore, the 5G Labs slicing architecture allows for dynamic creation of user profiles featuring access to specific or multiple slices by means of dedicated network slice management APIs.

The testbed also provides network and services monitoring capabilities, integrating different opensource tools used to collect and store monitoring data and to present them through dashboards (Prometheus [3] and Grafana [4]). The monitoring capabilities are focused on three main areas:

- Infrastructure monitoring: collecting the data from the infrastructure components, e.g., UEs, RAN, CN, Transport, virtualized edge infrastructure.
- Network and service performance monitoring: evaluating network resources, services status, users.
- Network slice and device monitoring: collecting data related to the traffic volumes specific to certain network slices, as well as information related to the radio link quality.

For the purpose of the PoC, and to facilitate the AI-enabled closed loop automation solution proposed, the testbed exposes a set of network and edge resource management APIs:

- Monitor system APIs, to expose service and network monitoring data towards external applications.
- Slice Management APIs, to configure and dynamically adapt end-to-end network slice capabilities. They provide slice management capabilities, including inventory, adapted to 5G testbed network elements available system, exposed to external systems
- Kubernetes APIs, to expose the capability to deploy cloud-native applications in the extreme-edge/edge computing infrastructure available in the testbed

2.2.2 PoC Components

The proposed PoC architecture includes a zero-touch network and service management solution, featuring an AI-enabled closed loop automation framework and an MLOps platform, all integrated with the PoC infrastructure described in clause 2.2.1 which provide 5G SA and extreme-edge/edge computing capabilities.

The PoC architecture is shown in Figure 2, and a detailed description of the PoC components is provided in Table 1, along with their mapping to ZSM services. The lower part of Figure 2 depicts a simplified view of the PoC infrastructure hosted at the Orange Romania testbed lab, and it highlights the main testbed components, APIs and extreme-edge/edge computing nodes that are exposed towards the PoC components to realize, validate and demonstrate the AI-enabled zero-touch closed loop automation solution proposed in this PoC. In particular, two different types of UE devices are used:

- 2x Intel NUC 12 Pro mini-PCs, acting as extreme-edge devices, where the virtual robot application component of the PoC emulated robot application is deployed and executed
- 1x Milesight MS-C826 camera, as static 5G-enabled camera to generate additional traffic and emulate congestion conditions.

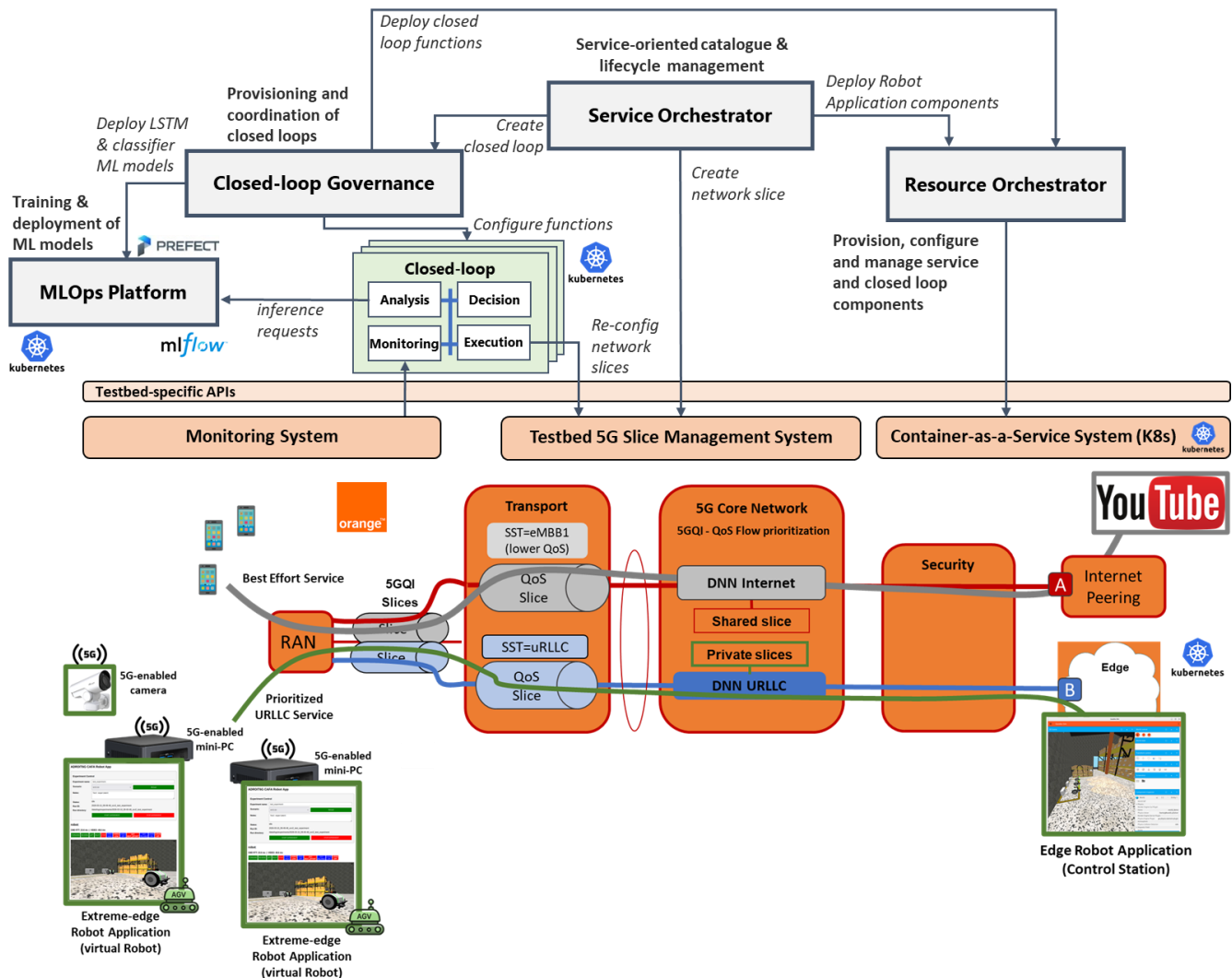


Figure 2: PoC Architecture

As shown in the figure, two different network slices are used according to the Orange Romania testbed capabilities. A URLLC network slice is the main target for this PoC demonstration, and it is provisioned, configured and proactively reconfigured through the AI-enabled closed loop automation solution proposed in the PoC. Another best effort eMBB slice is also statically configured and available for regular internet traffic generated by legacy 5G UE devices.

More details on the PoC Robot Application setup are provided in the following section 2.2.3.

Table 1 PoC components and their alignment with ZSM services

Component	Description	Provided ZSM services
Service Orchestrator	Responsible for the lifecycle management of cloud-native vertical services deployed in 5G infrastructures and spanning across extreme-edge, edge and cloud domains. It supports provisioning, runtime operations and termination of the service. It interacts with the Resource Orchestrator for the deployment of cloud-native applications, with the Orange Romania testbed network slice management APIs to select, configure and re-configure network slices, and with the Closed Loop Governance to deploy service-specific closed loops in support of runtime automation. It exposes both proprietary and standard APIs, aligned with TM Forum API (TMF 633 [5], TMF 640 [6]) to expose a catalogue of service offerings and to dynamically provision, activate, configure, and operate vertical service instances.	E2E Orchestration
Resource Orchestrator	Manages and orchestrates the edge computing resources in the extreme-edge and edge domains, supporting resource inventory and dynamic discovery of available nodes, and deployment of service and application components in multi-node clusters. It interacts with the Orange Romania Kubernetes clusters at the extreme-edge (i.e. those at the device level hosted in the mini-PCs) and at the edge using the related APIs for the deployment, termination and management of the emulated robot application cloud-native applications. It exposes proprietary interfaces towards the Service Orchestrator and the Closed Loop Governance for the deployment of vertical applications and closed loop functions.	Domain Orchestration
Closed Loop Governance	Manages and orchestrates customizable, cross-layer, zero-touch AI-enabled closed loop functions aligned with ETSI ZSM 009-1/2/3 specifications. It introduces full automation in network and service management providing dynamic provisioning, configuration, runtime updating, and termination of virtualized closed-loop functions, deployed as cloud-native applications through the Resource Orchestrator. It exposes proprietary APIs towards the Service Orchestrator, and closed loops are modelled through Closed-Loop Descriptors (CLDs), based on the 3GPP closed loop automation data models and with additional parameters for dynamic lifecycle management capabilities. The Closed Loop Governance Framework interacts with the MLOps platform to query, select and request the on-demand deployment of AI/ML models to be integrated with AI-enabled closed loop functions.	Closed Loop Governance
MLOps Platform	Manages the lifecycle of AI/ML models, supporting on-demand and dynamic training and deployment of models in the extreme-edge/edge compute continuum. It provides automated ML tasks and pipelines management, and ML model versioning and catalogue functionalities, specifically by adopting and integrating opensource tools, respectively Prefect [7], MLFlow [8] and Seaweed Fs [9]. It leverages on the Resource Orchestrator to deploy ML tasks and models as containerized applications in the extreme-edge/edge nodes. It exposes proprietary APIs to trigger AI/ML models training and deployment actions, which are consumed by the Closed Loop Governance when creating and configuring AI-enabled closed loop functions.	Domain Intelligence
Closed Loop	The Closed Loop functions follow the principles defined in ZSM 009-1, and	-

Functions	<p>represent the actual automation enabler in this PoC. Closed loop functions are service-specific, and customized to specific automation goals. In the PoC, the automation goal is predictive network slice optimization, to guarantee network performance in terms of latency and throughput in support of the emulated robot applications communications between virtual robots and the control station (see next section 2.2.3 for more application related details). Four different types of closed loop functions are used in the PoC. A monitoring closed loop function is dynamically created and used to access the Orange Romania monitoring system APIs and retrieve pre-defined time-windows of data related to network slice performances (such as aggregated latency measures, uplink throughput, downlink throughput, etc.), and make them available to the analytics function. An AI-enabled analysis function is used to predict network slice performance, through an AI/ML LSTM model which is exposed by the MLOps platform. An AI-enabled decision function processes the predictions of the analysis step, and classify/identify actual slice congestion scenario, through a random forest classifier model which is exposed by the MLOps platform. An execution function, in case the network slice congestion is predicted, implement a remediation action to increase the network slice QoS profile (i.e. scale the network slice) through the Orange Romania 5G network slice management APIs. The four closed loop functions are implemented following a common template/SDK, which allows to easily enable communication and data exchange among through a message bus.</p>	
-----------	---	--

2.2.3 PoC Application

The CAFA Robot Application is a distributed robotics test application designed to generate and measure network traffic between multiple robots and a central control station. The primary purpose of the application is to evaluate the impact of different network (slice) configurations on application performance, particularly communication latency.

The application simulates a fleet of robots operating in a shared environment using the Gazebo simulator [10]. Each robot produces realistic data streams, including video and LiDAR sensor data, and receives control commands from a Control Station. These data streams are relayed through 5G-enabled mini-PCs which emulates real robot devices and execute the virtual robot applications.

The PoC robot application provides interactive robot control through a web-based dashboard, real-time monitoring of command and video latency, scripted robot scenarios for repeatable experiments, latency measurements for post-analysis, configurable video and LiDAR data streams to generate network load. The application is intended to support controlled experiments that measure the performance of 5G network setups and any available network optimization solutions to be validated. The PoC robot application consists of two main components: i) a Control Station to be deployed as a cloud-native edge application, ii) a set of virtual Robot applications (one for each emulated robots) to be deployed as cloud-native extreme-edge applications in the mini-PCs available in the Orange Romania testbed. Together they emulate a distributed robotic system communicating through a 5G network infrastructure.

The Control Station is the central component responsible for running the Gazebo simulation environment, managing robot control commands, streaming robot video and LiDAR data, measuring communication latency, running scripted robot scenarios, provide a web-based dashboard for user interaction. The Control Station components are listed in

The Control Station is built by the integration of several internal components, as listed in Table 2. The Control Station is packaged as a Docker container so that the full system can be deployed and reproduced in cloud-native infrastructures.

Table 2 PoC robot application components – Control Station

PoC Application – Control Station Components	Description
Gazebo Simulator	Simulates the robot environment and sensors.
ROS 2	Middleware used for robot communication
ros_gz_bridge	Connects Gazebo topics to ROS 2 topics
Stream Switch	Dynamically enables/disables video and LiDAR streams
Flask Server	Provides the web dashboard and experiment control
Dashboard	Web interface for robot control and experiment management

Each robot is represented by a 5G-enabled mini-PC device that acts as a relay between the Control Station and the emulated robot. The 5G-enabled mini-PCs are extreme-edge nodes with computing capabilities, and host Kubernetes nodes to allow for dynamic and on-demand deployment of cloud-native virtual robot applications. These virtual robot applications receive sensor streams (video and LiDAR), forward them through the network to the Control Station, receives control commands from the Control Station and acknowledge control commands for latency measurement. These nodes emulate the communication behaviour of real robots and allow network traffic to traverse the 5G network infrastructure.

The CAFA robot application also includes built-in latency measurements for both control commands and sensor/video streams. For movement commands, the Control Station timestamps each command at the moment it is issued (e.g., from the dashboard or a scripted scenario). When the command is received and applied on the robot side (via the Raspberry Pi relay), an acknowledgment message is returned. The round-trip time (RTT) is calculated at the Control Station as the difference between the original send timestamp and the acknowledgment reception timestamp. This value represents the end-to-end command latency over the 5G network and is continuously displayed in the dashboard.

For video (and optionally LiDAR), each published frame is associated with a timestamp generated at the Control Station before forwarding toward the Raspberry Pi and back to the dashboard. When the frame is received for display, the application computes the latency as the difference between the display time and the original frame timestamp. The current frame latency and a rolling average are shown in the dashboard in real time. All measured values (command latency, video latency, and optionally LiDAR message latency) are logged to file in CSV format, including timestamps, robot ID, and configuration (video mode, QoS, etc.), enabling offline statistical analysis and comparison of different 5G configurations.

These mechanisms allow to execute and evaluate virtual robot applications and 5G network performance comparison with and without the proposed PoC solution for zero-touch network and service management solution with AI-enabled operational closed loops, which is indeed part of the PoC demonstration procedure described in clause 2.3 below.

2.3 PoC Demonstration Procedure

The PoC demonstration will be structured in two different scenarios, which aim at validating the benefits introduced by the proposed AI-enabled closed loop automation solution in the management and orchestration of 5G-enabled collaborative robot services and applications. In particular, the following scenarios will be executed:

- Scenario#1: on-demand and automated robot application service deployment and operation, without the use of AI-enabled closed loop automation. In this scenario the goal is to demonstrate that when the closed loop automation features are not activated, the vertical service for the collaborative robot application can be automatically deployed in the extreme-edge/edge infrastructure through the Service and Resource Orchestrators, combined with the provisioning and configuration of a 5G URLLC network slice capable to satisfy the network performance requirements (in terms of latency and throughput) for a given number of virtual robots. However, in case of unexpected increase of network traffic, e.g. due to additional (and not initially planned) devices generating traffic (such as static cameras, additional robots, etc.), the network slice gets congested and the robot application performances (in terms of application latency and throughput) degrades, affecting the quality and efficiency of the robot service.
- Scenario#2: network slice performance optimization through AI-enabled operational closed loops. In this scenario

Figure 3 shows procedure for the execution of the PoC Scenario#1. In the Step 1, the robot application vertical service is deployed on-demand upon the request from the ZSM consumer, i.e. the vertical actor. The application components (virtual robots and control station) are deployed respectively at the extreme-edge and edge nodes, and the URLLC network slice required to satisfy the latency and throughput requirements of the application is also provisioned automatically. In Step 2, the robot application vertical service is up and running, the virtual robots can start emulating the video streams and their trajectories and collaborative actions. In Step 3, the network traffic in the URLLC is intentionally increased with the aim of introducing a congestion in the network slice. The goal is to saturate the network slice capacity, and force a degradation of the robot application performances (in terms of latency and throughput among the involved components). Therefore, without any closed loop automation in place, the runtime operation of the robot application service is subject to performance degradations that should be remediated manually and only after their occurrence.

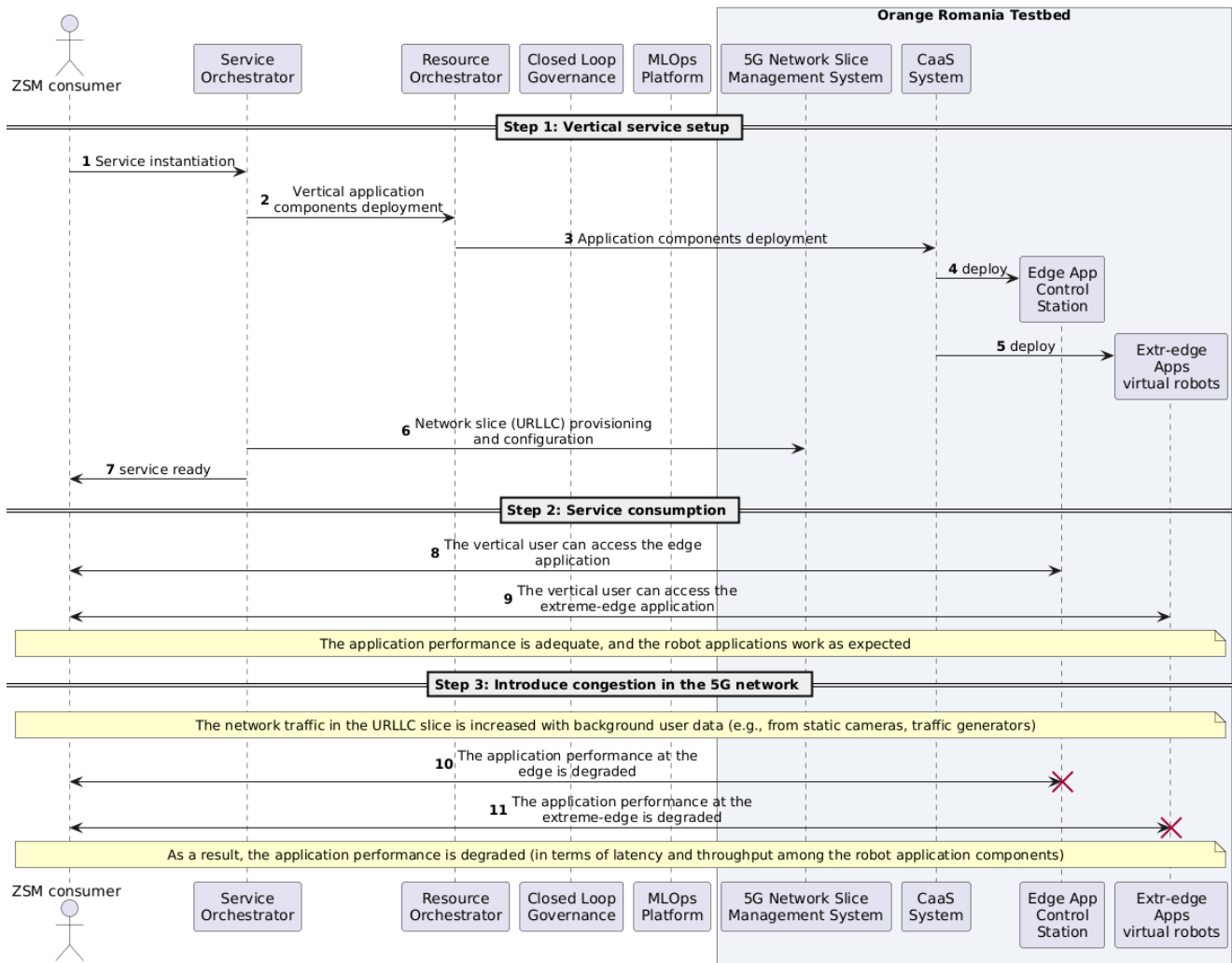


Figure 3: PoC demonstration – Scenario #1, without AI-enabled closed loop automation

On the other hand, Figure 4 and Figure 5 show the procedure for the execution of the PoC Scenario#2. The first part of Step 1, related to the deployment of robot application components and network slice configuration, is the same as for Scenario #1. In addition, in this Scenario#2 case, the Service Orchestrator takes care to trigger the deployment of the AI-enabled operational closed loop specific for this network slice optimization case, thus involving the Closed Loop Governance for the management of the deployment operation, and in turn the MLOps platform for the deployment of the AI/ML models (LSTM for network slice performance prediction, and Random Forest for the classification and network slice congestion prediction). As a result, four closed loop functions are created and deployed a configured: a monitor function, an AI-enabled analysis function (that can trigger inferences to the LSTM model), a decision function (that can trigger inferences to the Random Forest model), an execution function. In Step 2, the robot application vertical service is up and running, the virtual robots can start emulating the video streams and their trajectories and collaborative actions. In parallel, the AI-enabled

closed is periodically executed to monitor and analyse the network slice to predict potential performance degradations. For this, the monitor function retrieve time-window of network slice performance data (e.g. uplink/downlink throughput, latency, etc.), which is then used by the analysis function to invoke the LSTM model to predict the network slice performance in the next time-window. The prediction is then processed by the decision function, which invoke the Random Forest model to decide if a congestion will occur in the next time-window. In Step 3, the network traffic in the URLLC is intentionally increased with the aim of introducing a congestion in the network slice. In particular, it is increased with background user data (e.g., from static cameras, traffic generators). Prior to network slice and application performance degradation, the AI-enabled closed loop predicts a congestion event. Indeed during the execution of a closed loop cycle, the combination of AI-enabled analysis and decision functions predict a congestion for the next time-window, and trigger the execution function to invoke the APIs exposed by the testbed network slice management system to upgrade the profile of the URLLC network slice to accommodate the predicted network traffic increment. As a result, the vertical service performances are kept adequate, so that the robot application components continue to work as expected despite the network traffic increment.

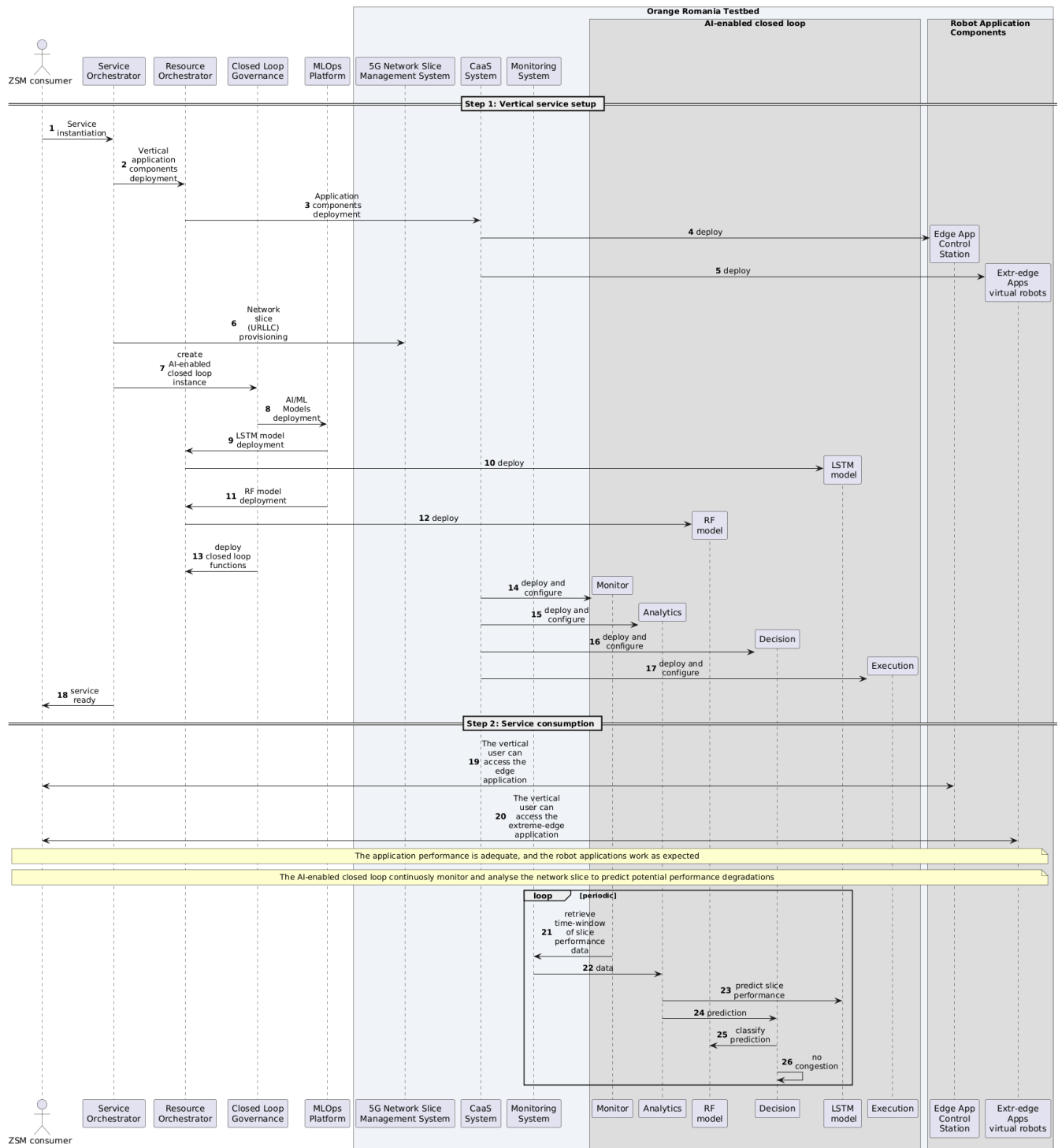


Figure 4: PoC demonstration – Scenario #2, with AI-enabled closed loop automation (step 1 and step 2)

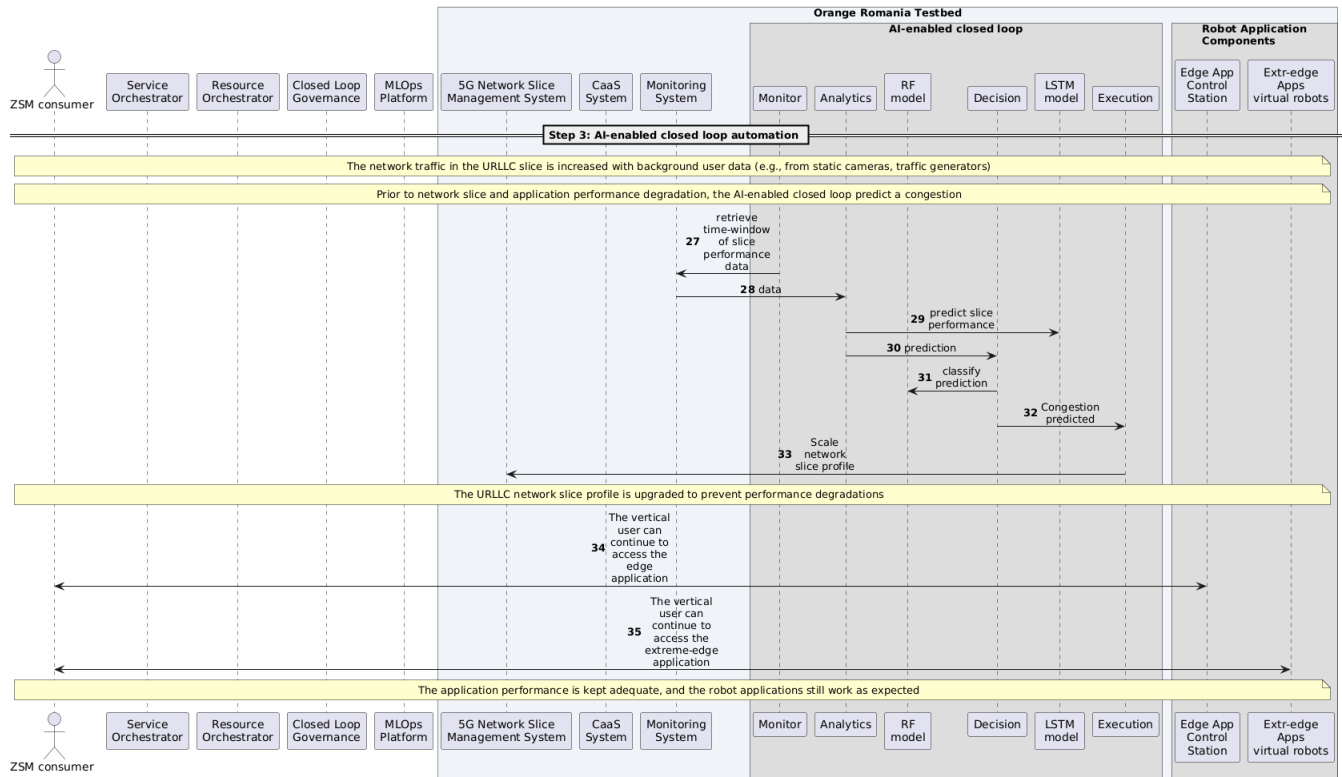


Figure 5: PoC demonstration – Scenario #2, with AI-enabled closed loop automation (step 3)

2.4 Additional information

- [1] Horizon Europe ADROIT6G project [Online]. Available: <https://adroit6g.eu/>
- [2] Orange Romania 5G Laboratory testbed infrastructure, <https://5glab.orange.ro/en>
- [3] Prometheus, <https://prometheus.io/>
- [4] Grafana, <https://grafana.com/>
- [5] TMF 633, Service Catalog API User Guide v4.0.0, Jan 2021
- [6] TMF 640, Service Activation and Configuration API User Guide v4.0.1, Jul 2020
- [7] Prefect, <https://www.prefect.io/>
- [8] MLFlow, <https://mlflow.org/>
- [9] Seaweed Fs, <https://seaweedfs.com/>
- [10] Gazebo, <https://gazebo.org/>