|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A black and white logo  Description automatically generated with low confidence | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2025-2028 | | **Focus Group on AI Native for Telecommunication Networks  (FG-AINN)** | |
| **FG-AINN-139** | |
| **Original: English** | |
| **Question(s):** | | WG4 | Geneva, 11 July 2025 | |
| **INPUT DOCUMENT** | | | | |
| **Source:** | | Chairs, WG4 | | |
| **Title:** | | AI-Native Autonomous Agent for Real-Time 6G Network Security, Optimization, and Slice Assurance | | |
| **Contact:** | | Preksha Shah Indian Institute of Technology Bombay, Mumbai, India | | Tel:  E-mail: 30005100@ee.iitb.ac.in |
| **Contact:** | | Vishnu Ram OV Independent Consultant  India | | Tel:  E-mail :vishnu.n@ieee.org |

|  |  |
| --- | --- |
| **Keywords:** | AI Native Network; PoC; 5G Core; AI Agent; Network Optimization; Anomaly Detection; Network Slicing; Resilience |
| **Abstract:** | This contribution proposes a Proof-of-Concept (PoC) under FG-AINN exploring an **AI-native multi-agent architecture** for 5G/6G networks. Unlike monolithic threshold-driven monitoring, the system integrates **six collaborative agents** (Data Quality, QoS, Failure Prediction, Traffic Forecast, Energy Optimization, Security) communicating through a **Redis message bus**. Each agent transforms validated network telemetry into actionable outputs, enabling proactive anomaly detection, intent-driven orchestration, and SLA-centric optimization.  By leveraging reinforcement learning, explainable AI, and closed-loop automation, the framework improves resilience, reduces mean time to recovery (MTTR), and builds operator trust. |

1. **Introduction**

The motivation for this PoC stems from limitations in current network management approaches that are reactive, rigid, and require significant manual intervention. Existing systems rely on static thresholds, pre-configured policies, and operator-driven remediation, which result in slow recovery from anomalies, inefficient resource usage, and poor adaptability in dynamic environments.

The proposed PoC aims to demonstrate an **AI-native agent** that learns network context from real-time core metrics (AMF, SMF, UPF), detects anomalies beyond static thresholds, and takes **intent-driven actions** to ensure SLA compliance and slice performance. The solution will be evaluated by simulating slice overload, signaling storms, and UE surges in a 5G/6G testbed, measuring recovery time, SLA adherence, and efficiency improvements.

This PoC aligns with FG-AINN’s Terms of Reference by providing concrete contributions to AI-native network architecture, intent-based automation, and resilience-centric inference systems.

1. **Proof of Concept Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Submission Id* | | | | *G-AINN-PoC-xxx*  *(Id number is to be assigned by the secretariat)* |
| *Title* | | | | AI-Native Autonomous Agent for Real-Time 6G Network Security, Optimization, and Slice Assurance |
| *Created by* | | | | Harsh Sahu |
| *Creation date* | | | | 17/09/2025 |
| *Category* | | | | *AI Agents; Intent-Based Networking; Emergency & Resilience-Centric AI Inference Systems* |
| *PoC Objective* | | | | * To demonstrate context-aware anomaly detection in 5G/6G core networks. * To validate intent-based autonomous control loops for slice assurance and QoS guarantees. * To showcase resilience-driven AI automation that reduces downtime and improves SLA compliance. * To highlight how AI-native methods outperform legacy static monitoring and rule-based approaches. |
| *Description* | | | | The PoC implements an AI-native agent integrated with a 5G/6G core testbed (Open5GS + UERANSIM + Prometheus/Grafana). The agent leverages reinforcement learning and context inference for real-time decision-making, optimizing network slices dynamically under varying traffic and fault scenarios. Key Features:   * **Context-aware AI**: Detects nuanced anomalies (e.g., signaling floods, slice starvation) beyond threshold rules. * **Intent-based automation**: Operators declare *what they want* (e.g., “maintain <10ms URLLC latency”), and the agent autonomously enforces policies. * **Explainability**: Each AI action is logged with reasoning for operator trust. * **Resilience optimization**: Fast failover, dynamic scaling, and resource reallocation reduce downtime and operational cost.   **Specialized Agents**   * **Data Quality Agent** – Input: raw KPIs; Output: validated data, integrity alerts. * **QoS Agent** – Input: clean KPIs; Output: QoS anomalies, root causes. * **Failure Prediction Agent** – Input: anomalies, energy use; Output: failure risk scores. * **Traffic Forecast Agent** – Input: load, failures; Output: throughput/utilization forecast. * **Energy Optimization Agent** – Input: traffic forecasts; Output: power adjustment plans. * **Security Agent** – Input: auth logs, KPIs; Output: threat detection, mitigation actions.  3.2 Message Bus Protocol Redis pub/sub channels:   * anomalies.alerts – anomalies from QoS, security, or data quality. * optimization.commands – traffic and energy recommendations. * actions.approved – coordinator approvals. * actions.executed – results from applied actions. * operator.commands – manual overrides. |
| *Feedback to WG1* | *Gaps Addressed* | | | Lack of context-aware AI in current monitoring systems.  Absence of explainable decision-making in existing automation.  Inefficiency of static slice/resource provisioning under dynamic traffic.  Limited resilience against real-time anomalies like signaling storms. |
| *POCs Test Setup* | | | | * **5G Core**: Open5GS with AMF, SMF, UPF. * **RAN Simulator**: UERANSIM to generate traffic loads. * **Monitoring Layer**: Prometheus for metrics collection. * **Visualization**: Grafana for dashboards. * **AI Agent**: Custom-built, reinforcement learning–based with policy engine. |
| *Data Sets* | | | | * Synthetic traffic traces (eMBB, URLLC, mMTC). * Network anomaly traces (overload, DoS-like behavior, slice congestion). * Realistic telecom KPIs (latency, jitter, throughput, registration success). |
| *Feedback to WG2* | | | *Simulated Use cases* | * **Use Case 1**: URLLC slice overload detection and automated rerouting. * **Use Case 2**: Signaling flood anomaly detection and mitigation. * **Use Case 3**: Adaptive scaling of UPF resources during IoT surges. * **Use Case 4**: SLA assurance for enterprise slices (e.g., <1% packet loss). |
| *Feedback to WG3* | | *Architectural concepts* | | * AI-native closed-loop architecture integrated at the 5G core control plane. * Intent-driven orchestration replacing static rule-based configurations. * Explainable AI interfaces for operator trust. * Resilience-first AI inference enabling self-healing networks. * Test Setup: * **Core:** Open5GS (AMF, SMF, UPF). * **RAN:** UERANSIM. * **Monitoring:** Prometheus + Grafana. * **Agent Framework:** Python async + ML (Isolation Forest, LSTM, DBSCAN). * **Orchestrator:** Kubernetes for scaling. |
| *Demo and Evaluation* | | | | * **Scenario 1 (Congestion):** QoS → Traffic → Energy → Coordinator approves load balancing. * **Scenario 2 (Security):** Security → Data Quality → QoS → Security mitigation.   **Metrics:**   * Anomaly detection accuracy (AI vs static). * SLA compliance under overload. * MTTR reduction (target < 5s). * Resource efficiency (avoid over-provisioning).   . |
| *PoC Observation and discussions* | | | | *The AI-native approach significantly reduces downtime compared to traditional systems.*  *Intent-driven automation reduces operator workload and improves agility.*  *Explainability mechanisms increase trust in AI-driven decisions.* |
| *Conclusion* | | | | The PoC demonstrates that AI-native, context-aware, and intent-driven agents can outperform legacy threshold-based and rule-driven systems in 5G/6G networks. By enabling real-time anomaly detection, SLA assurance, and resilience optimization, the proposal provides a foundation for standardization in FG-AINN. |
| *Open Problems and Future Work* | | | | **Scaling AI agents across multi-domain/multi-operator environments.**  **Defining standard APIs for intent-to-action translation.**  **Developing benchmarks for explainable AI in telecom automation.**  **Extending PoC for cross-layer optimization (RAN + Core + Transport).** |
| *References* | | | |  |

# 

# 

# 

# 

# 

# 

# 

# 3. Implementation proposal (Proposal Sample)

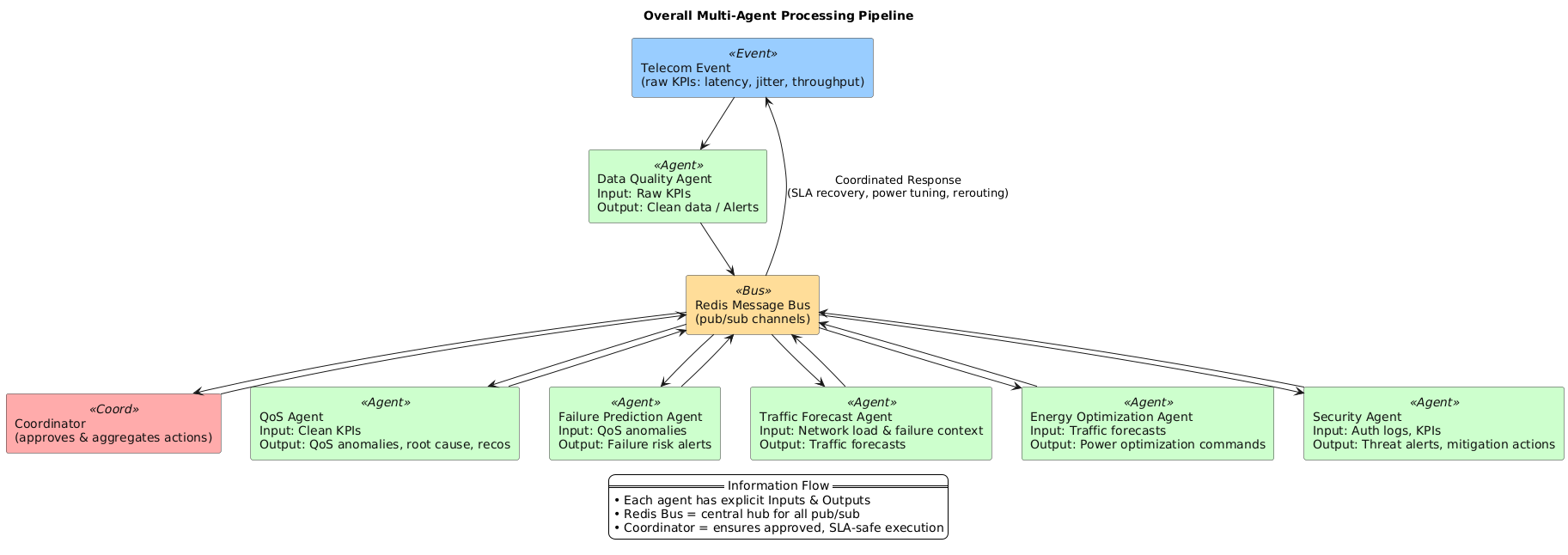
This clause describes the implementation proposal for the PoC submission above. The implementation is structured in phases, with incremental enhancement from basic test setup demonstration to advanced AI-native features.

### 3.a Description of the Test Setup

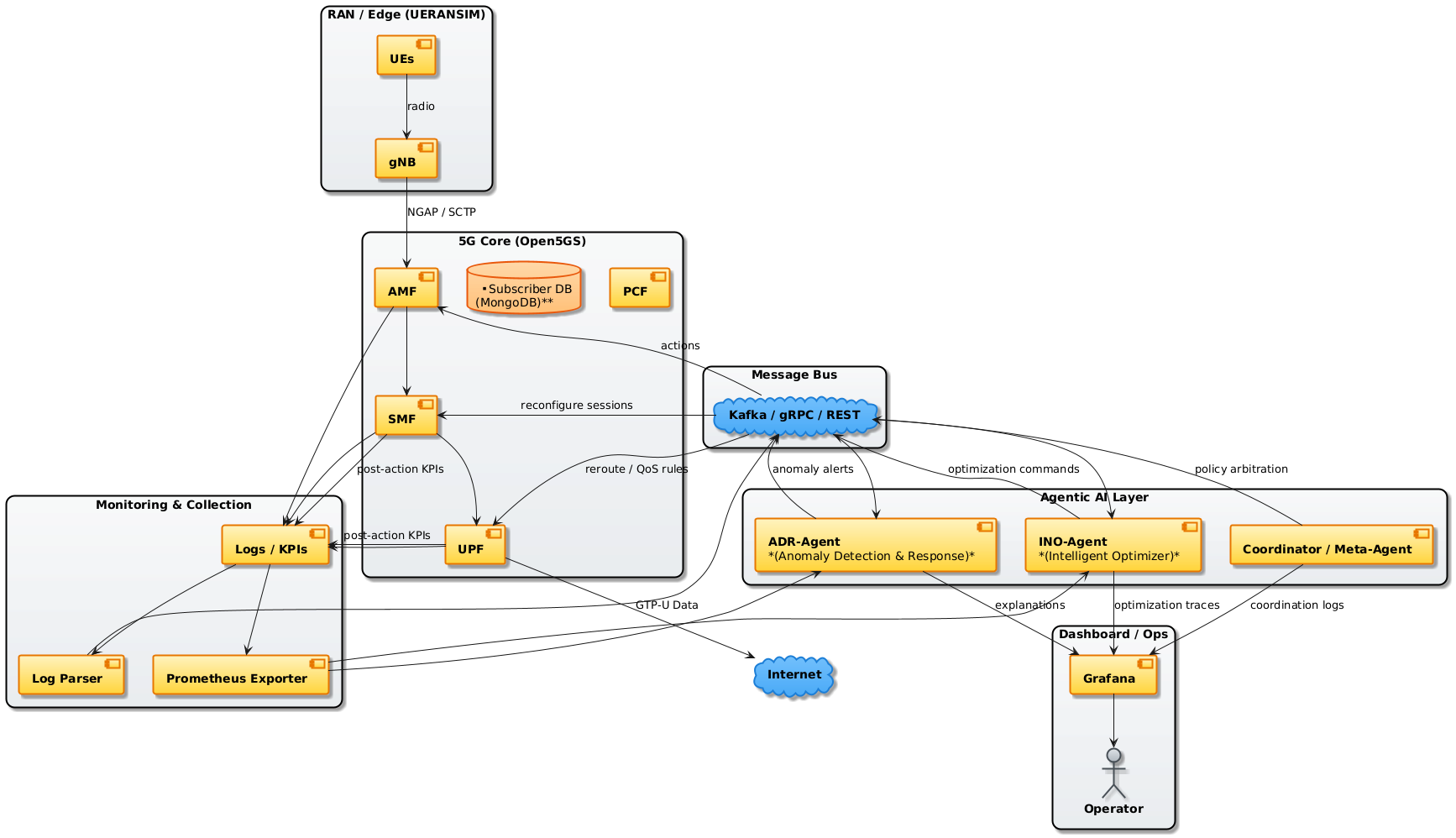
### 

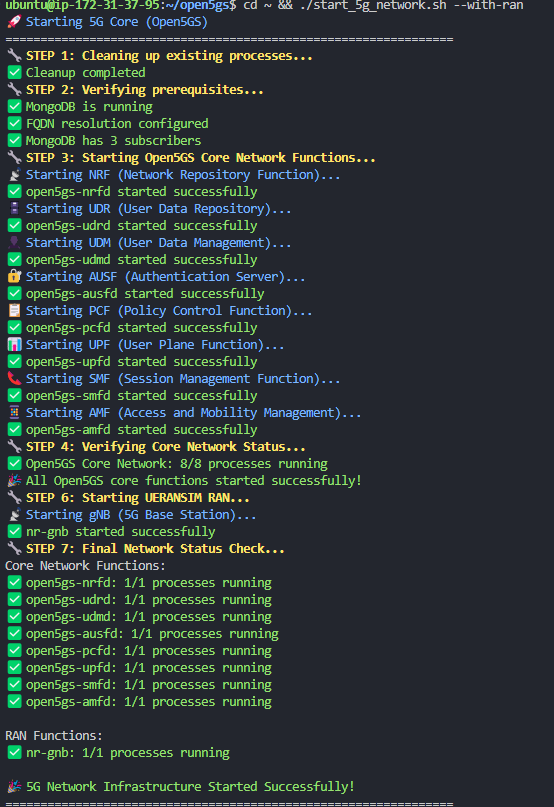
The PoC test setup involves the following components:

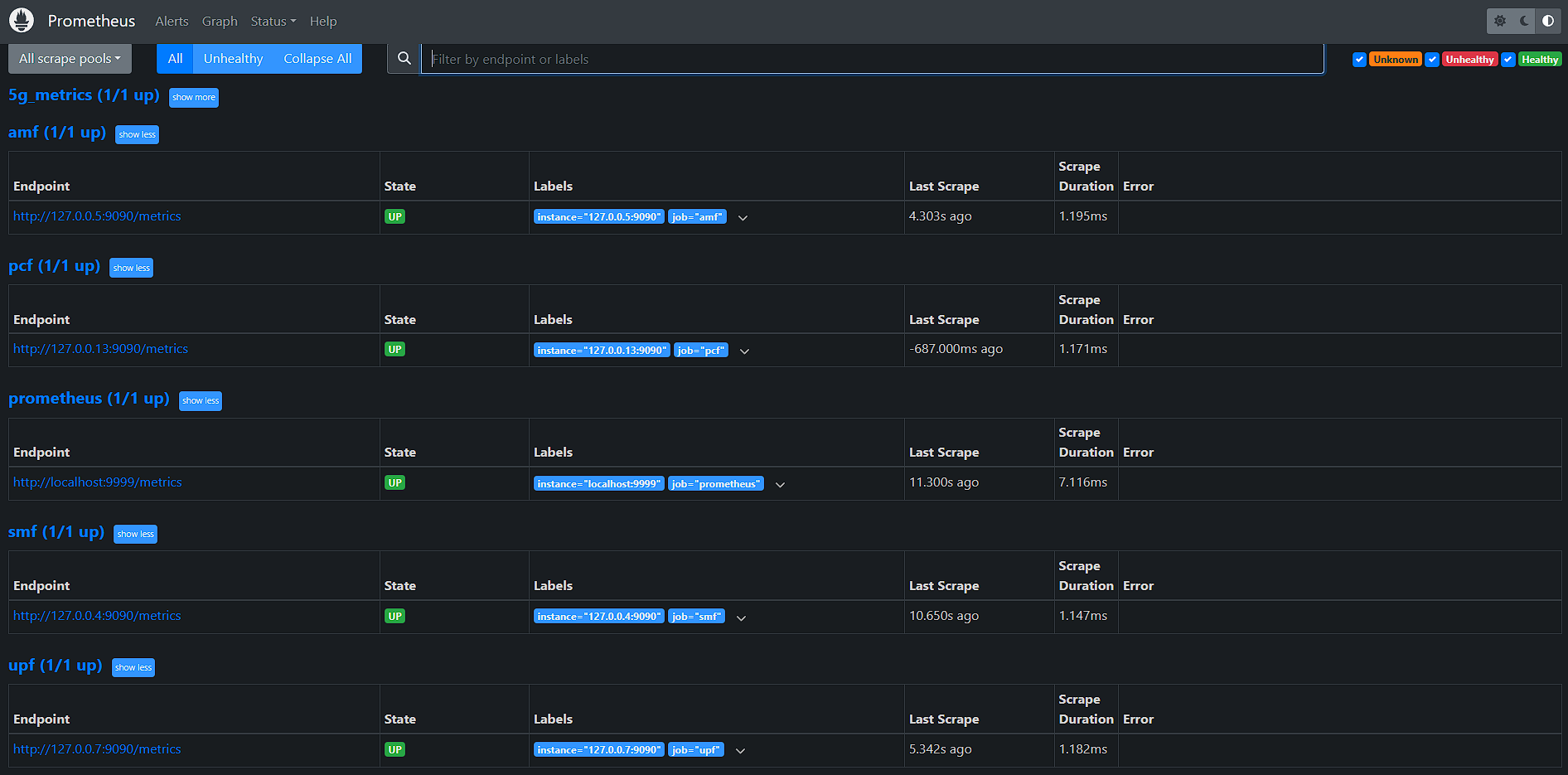
* Code Generation Model:  
   Lightweight AI inference model trained on synthetic and real telecom KPI datasets, supporting anomaly classification and intent translation. Implemented using PyTorch and TensorFlow Lite for fast inference.
* Service Orchestrator:  
   Kubernetes-based service orchestrator integrated with Open5GS core functions, enabling dynamic scaling of AMF/SMF/UPF instances.
* Agent Framework:  
   Custom AI-native autonomous agent built with reinforcement learning and policy engines. The agent continuously ingests telemetry from Prometheus, reasons on context, and triggers orchestrator actions based on intent.
* Simulator:  
   UERANSIM is used to emulate multiple UEs, variable traffic patterns (eMBB, URLLC, mMTC), and overload/failure conditions.
* Monitoring Layer:  
   Prometheus scrapes metrics from Open5GS functions (AMF, SMF, UPF, PCF). Grafana visualizes KPIs and AI agent actions.
* Datasets:  
  + Synthetic traffic traces (generated from UERANSIM).
  + Public datasets such as [OAI traffic traces] and [5G KPI datasets].
  + Custom anomaly injection traces (signaling floods, slice starvation, CPU/memory overload).



### 3.b Description and Reference to Base Code

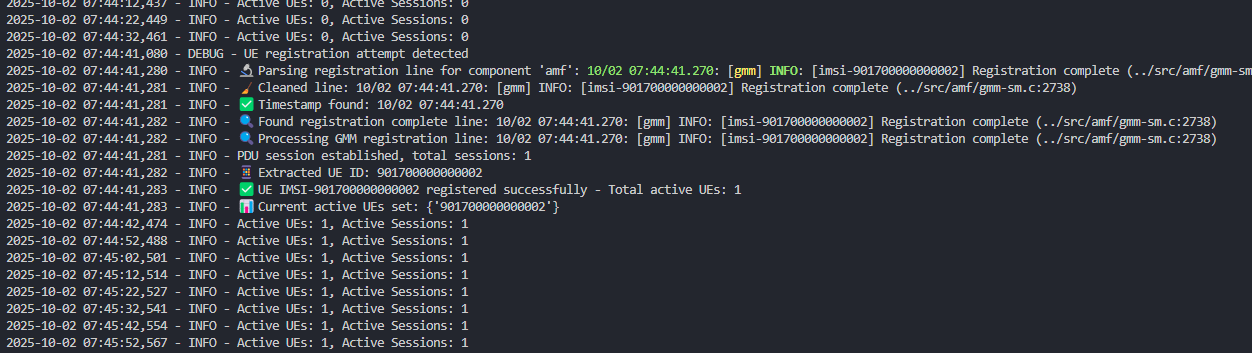
* Base Code Used:  
  + [Open5GS GitHub Repository](https://github.com/open5gs/open5gs)
  + [UERANSIM GitHub Repository](https://github.com/aligungr/UERANSIM)
  + [Prometheus GitHub Repository](https://github.com/prometheus/prometheus)
  + [Grafana GitHub Repository](https://github.com/grafana/grafana)
  + Custom AI Agent (to be released under PoC repo).
* Components Relevant for Demo:  
  + Open5GS Core functions (AMF, SMF, UPF, PCF).
  + UERANSIM for load/stress simulation.
  + Prometheus/Grafana stack for metric ingestion and visualization.
  + AI Agent for context-aware anomaly detection and intent-based orchestration.
* As-Is Usage:  
  + Open5GS deployed with default configuration.
  + UERANSIM used as-is to emulate realistic traffic.  
    
  + Prometheus/Grafana stack used for metric collection and visualization.
* Modified Components:  
  + AI Agent integrated with Prometheus exporters and Kubernetes orchestrator.
  + Custom exporters for slice-level KPIs (latency, jitter, throughput).
  + Policy engine for intent-to-action translation.

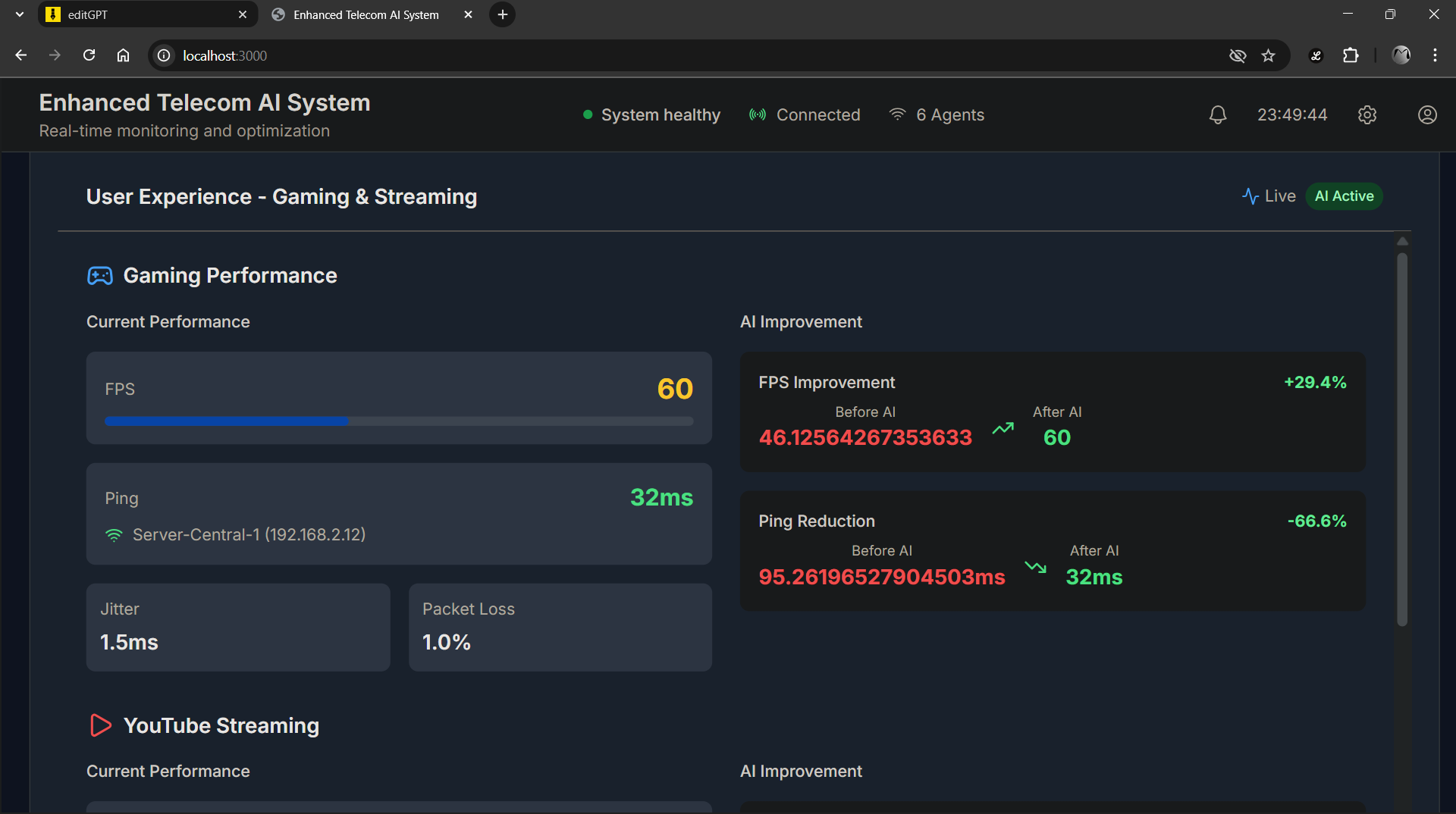
Open 5G’s Setup on AWS Cloud instance:  


Prometheus Interfacee:  


* Justification for Changes:  
   These changes enable context-aware inference and intent-based orchestration not available in baseline Open5GS/Prometheus setups.

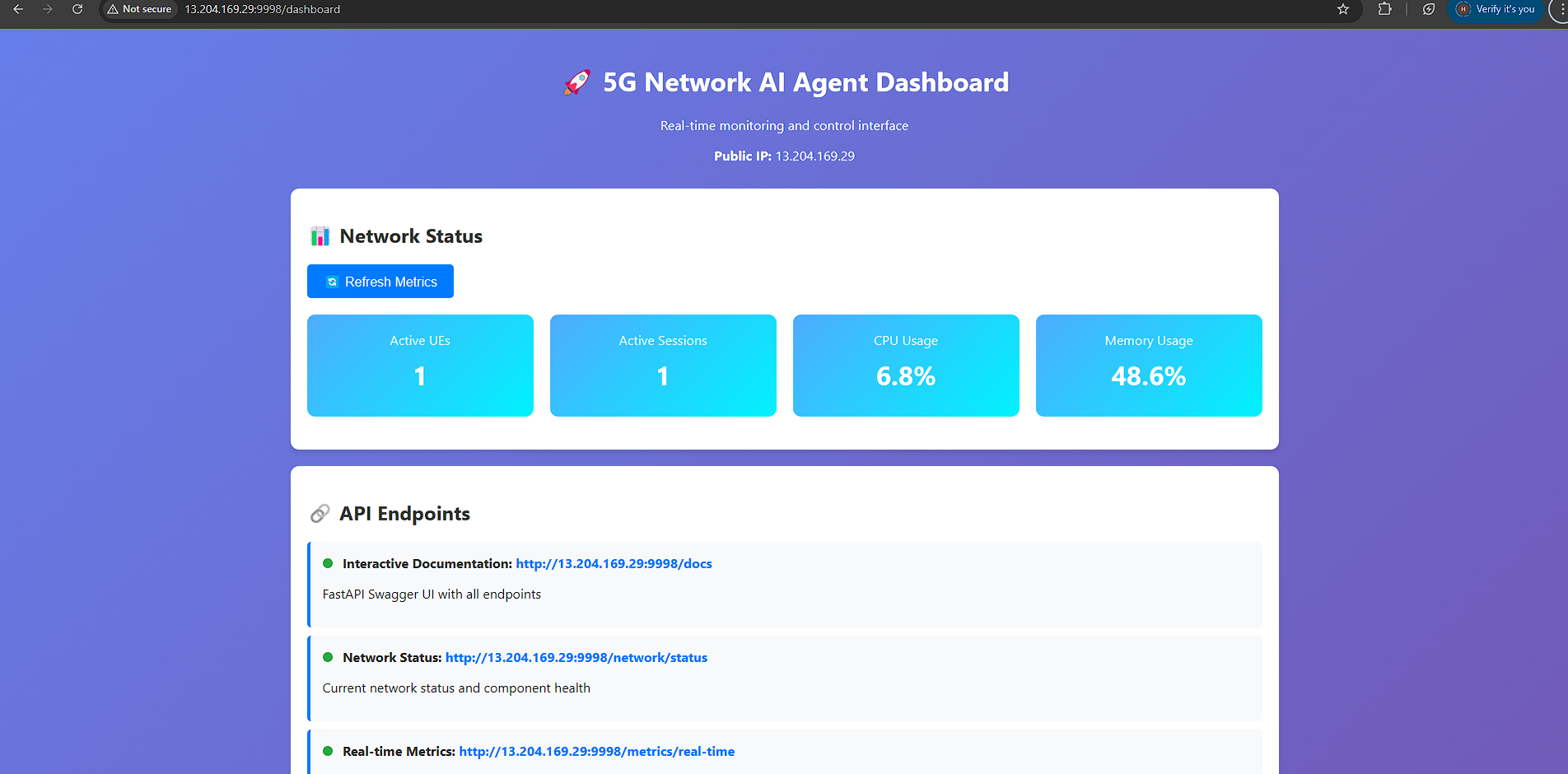
UE registration and IMSI Checkpoint :

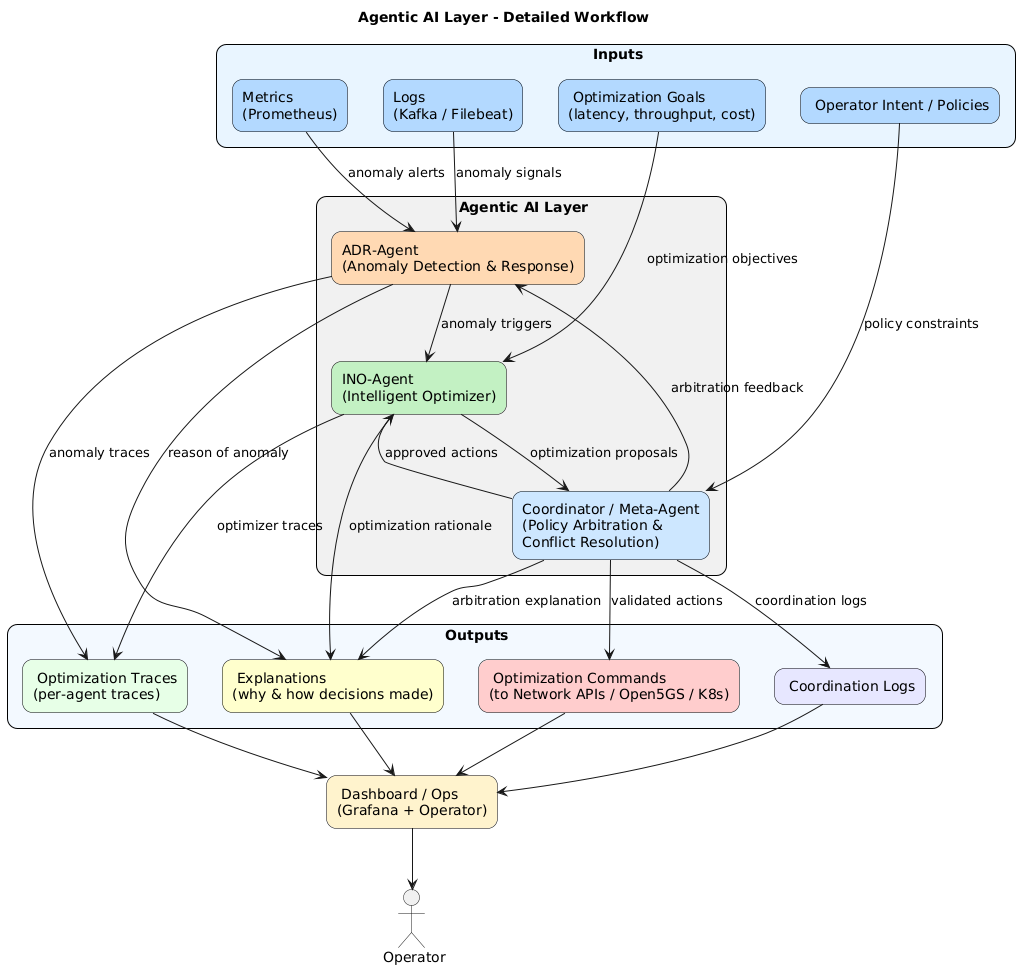




A screenshot of a video

AI-generated content may be incorrect.

5G AI Network Agent Dashboiard  




### 3.c Mapping to Demo Proposal in Clause 2

Functional Requirements:

* Req-1: AI agent must interface with Prometheus exporters and Kubernetes orchestrator for real-time decision-making.
* Req-2: Service templates (URLLC, eMBB, mMTC) must be defined in the agent’s knowledge base for intent translation.
* Req-3: The agent must ensure existing services remain unaffected during slice reallocation or scaling.

Datasets Needed:

* Synthetic KPI datasets from UERANSIM traffic.
* Anomaly injection datasets (overload, DoS, slice starvation).

Toolsets:

* Open source: Open5GS, UERANSIM, Prometheus, Grafana, PyTorch.
* Proprietary (optional): Kubernetes orchestration platform.

Test Cases:

* TST-1 (Self-X anomaly recovery):  
   Trigger a simulated failure in SMF via UERANSIM overload.  
   *Expected Result:* AI agent detects anomaly, reroutes traffic to standby SMF within 1 second, minimizing downtime.
* TST-2 (Knowledge base effect):  
   Compare anomaly detection performance with and without predefined slice templates in knowledge base.  
   *Expected Result:* SLA violations reduced by >40% when KB is active.
* TST-3 (MCP server integration):  
   Evaluate performance with and without integration of orchestrator (MCP server).  
   *Expected Result:* System with MCP server achieves faster scaling (20% improvement in MTTR).

**Bibliography**

[FGAINN intro] <https://www.itu.int/en/ITU-T/focusgroups/ainn/Pages/default.aspx>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_