An Agentic LLM-based Architecture for Automated Anomaly Detection and Adaptive Remediation

Paolo Palmiero[1], Antonio Iannaccone[2], Daniele Granata[2], and Roberto Nardone[2]

***Abstract*— Traditional cloud monitoring often relies on static remediation procedures that are difficult to adapt to dynamic and heterogeneous infrastructures. This paper proposes an agentic LLM-based architecture for adaptive remediation planning from confirmed cloud anomalies. The goal is not to replace anomaly detectors, but to transform confirmed anomaly events into structured, policy-constrained remediation artifacts suitable for human-supervised operational workflows.**

The proposed workflow combines anomaly intake, contextual validation, playbook retrieval, and constrained playbook generation through a message-driven Multi-Agent System. Retrieval-Augmented Generation is used to correlate current incidents with historical knowledge and existing procedures, while deterministic guardrails enforce schema validation, policy constraints, command allow/deny lists, critical-resource checks, and human approval for high-impact or previously unseen actions.

A containerised Proof of Concept demonstrates that confirmed anomalies can be transformed into CACAO-compatible remediation drafts within operationally reasonable time bounds. The evaluation focuses on generating and validating remediation plans under explicit operational constraints, rather than on anomaly detection benchmarking or on production-scale autonomous execution.

I. Introduction

The contemporary landscape of Artificial Intelligence (AI) is undergoing a fundamental evolution, transitioning from passive Large Language Models (LLMs) toward autonomous agentic systems capable of planning, reasoning, and collaborating within Multi-Agent Systems (MAS). This shift is particularly relevant for complex cloud and distributed infrastructures, where the scale, heterogeneity, and dynamism of modern Industry 5.0 environments challenge traditional, reactive monitoring approaches based on static, rule-based protocols. Such approaches require manual interpretation of heterogeneous telemetry data and rely on rigid remediation playbooks, leading to delayed responses and service degradation. Recent research has explored integrating LLMs into monitoring pipelines [3] to enhance early anomaly detection and the contextual understanding of telemetry data. While these approaches improve anomaly awareness, response actions are still commonly predefined and static, limiting their effectiveness in dynamic environments. In particular, automatically validating anomalies and adapting remediation strategies remain key limitations.

This paper explores how agentic LLM-based workflows can support adaptive remediation planning in cloud infrastructures after anomalies have already been detected and confirmed. The proposed approach leverages agentic reasoning within a Multi-Agent System (MAS) architecture to connect anomaly intake, contextual validation, strategy selection, and remediation planning within a unified message-driven workflow. Retrieval-Augmented Generation (RAG) and vector-based retrieval are used to correlate current anomalies with historical knowledge and previously validated procedures. Although the proposed architecture is sufficiently general to support both anomaly detection and adaptive remediation, the primary focus of this work is the orchestration workflow that connects anomaly confirmation, contextual validation, remediation planning, and structured playbook synthesis under explicit operational constraints, since constrained playbook generation represents a particularly challenging aspect in the assessment of practical feasibility.

The contributions of this work are: i) a MAS architecture that consumes confirmed anomaly events and supports adaptive remediation planning workflows; ii) a constrained workflow for the retrieval and generation of CACAO-compatible remediation playbooks using LLM-based reasoning and historical context; iii) a containerised Proof of Concept (PoC) demonstrating the feasibility of transforming confirmed anomalies into structured remediation drafts suitable for human-supervised validation.

The paper is organised as follows. Section II summarises the current state of the art; Section III proposes the system architecture and workflow; Section IV presents the implemented PoC and the obtained results; Section V outlines conclusions and future work.

II. Related Work

Recent advances in AI are driving a transition from passive LLMs toward autonomous agentic systems capable of planning, tool usage, and collaboration within Multi-Agent Systems (MAS). In complex and safety-critical infrastructures, reliability and correctness have traditionally been addressed through model-driven verification and validation approaches, as demonstrated in domains such as railway control systems [9]. Unlike traditional prompt-driven models, agentic architectures enable distributed reasoning and long-term task execution across multiple components. Consequently, an increasing

*This work has been partially funded by the European Union - Next-GenerationEU - National Recovery and Resilience Plan (NRRP) – MISSION 4 COMPONENT 2, INVESTMENT N. 1.1, CALL PRIN 2022 D.D. 1409 14-09-2022 – (DOSSIER - aDvanced mOnitoring SyStem wIth Enhanced secuRity) CUP I53D23006080001, and by the research projects MIRABILE (CUP C67B23000300004) - CDS001108 "Filiera Automotive 4.0" – PNRR M1C2 - 5.2 Filiere produttive - "Automotive" NextGenerationEU and ISTINTO (CUP C85I23000400004) - CDS001133 "Steel Network 4.0" – PNRR M1C2 - 5.2 Filiere produttive – "Metallo ed elettromeccanica" NextGenerationEU.

[1]P. Palmiero is with the IMT School for Advanced Studies Lucca, Lucca, Italy paolo.palmiero@imtlucca.it

[2]A. Iannaccone, D. Granata and R. Nardone are with the University of Naples Parthenope, Centro Direzionale, 80143 Napoli, Italy antonio.iannaccone001@studenti.uniparthenope.it, daniele.granata@uniparthenope.it, roberto.nardone@uniparthenope.it

body of research explores the adoption of LLM-based MAS across domains such as infrastructure monitoring, automated troubleshooting, and context-aware decision support. A significant research trend concerns the evolution of infrastructure monitoring from reactive diagnostic tools to *self-healing* and autonomous systems. Static, rule-based monitoring approaches have proven inadequate for the complexity of modern distributed environments, motivating the integration of LLMs to enhance anomaly awareness and response capabilities. Recent works demonstrate how agentic systems can augment traditional machine learning techniques to improve early anomaly detection and telemetry interpretation in cloud and networked environments [6], [1]. This trend is further extended by domain-specific agent frameworks designed to automate failure diagnosis and remediation in cloud platforms [10]. The increasing autonomy of such systems has also driven research into scalable coordination mechanisms and architectural patterns. Decentralised coordination models based on Directed Acyclic Graphs (DAGs) have been proposed to eliminate the overhead of centralised orchestration [12], motivating broader research directions on distributed agentic coordination models [13]. Complementary approaches, including layered knowledge-operational architectures [2] and message-driven pipelines for IoT environments [14], further illustrate the diversification of agentic coordination strategies across heterogeneous operational layers. In parallel, agentic architectures have been applied to personalised and context-aware recommendation systems, leveraging hierarchical edge–cloud deployments to balance responsiveness and computational efficiency [8]. As agent autonomy increases, security and governance become critical concerns in different domains [5], particularly in enterprise-scale deployments. Centralised registries and cryptographic control mechanisms have been proposed to regulate agent interactions and enforce policy compliance, as exemplified by the SAGA architecture [11]. These approaches aim to ensure reliability and controlled behaviour in increasingly autonomous systems [4]. Despite the growing interest in LLM-based operational agents, limited attention has been devoted to constrained remediation planning workflows that combine contextual retrieval, structured playbook generation, validation mechanisms, and human-supervised approval within a unified orchestration pipeline. This work focuses on that integration layer from a feasibility-oriented perspective.

III. Proposed Methodology and Workflow

This section presents the proposed architecture, illustrated in Figure 1. The framework employs a Multi-Agent System (MAS) to orchestrate anomaly-informed remediation workflows across cloud infrastructures. Its primary objective is to support contextual validation and adaptive remediation planning from confirmed anomaly events. In this context, a playbook represents a structured remediation procedure dynamically adapted to the observed anomaly conditions.

The architectural workflow operates as follows: monitored hosts continuously transmit telemetry data to an anomaly-detection module powered by a fine-tuned large language model (LLM). This module analyses incoming streams to identify faults or anomalous patterns. Upon detecting an anomaly, the detector agent publishes relevant metadata to a designated topic on a Message Broker, thereby initiating inter-agent communication.

A second component, the Strategy Selector Agent, subscribes to anomaly notifications from the Message Broker and performs validation using a Retrieval-Augmented Generation (RAG) system. This RAG framework operates on a vector database containing embeddings of previously encountered anomalies, where each entry comprises the anomaly descriptor, temporal metadata (detection timestamp), severity indicators, and references to successfully applied mitigation strategies. The embedding-based retrieval mechanism enables semantic similarity searches across historical anomalies. Following successful anomaly confirmation through vector similarity analysis, the Strategy Selector Agent queries the Playbook Collection to identify an appropriate mitigation strategy when a corresponding remediation procedure exists in the knowledge base. The agent then produces a validation report linking the anomaly to the corresponding playbook, which is subsequently published to a topic on the Message Broker for the Actuator Agent.

The Actuator Agent constitutes the final component in the pipeline, retrieving both anomaly specifications and corresponding mitigation strategy identifiers. For previously encountered anomalies with established remediation procedures, the agent executes the designated mitigation protocol through available tooling interfaces. When no predefined strategy exists, the Actuator Agent generates a new mitigation strategy based on previous solutions.

A. Anomaly Detector Agent

The Anomaly Detector Agent represents the upstream monitoring component responsible for producing anomaly candidates from infrastructure telemetry. In the current PoC, this component is not the main object of evaluation; instead, confirmed anomaly events are provided as structured JSON payloads to exercise the downstream remediation-planning pipeline. The proposed architecture is intentionally compatible with various anomaly detection backends, including rule-based monitors, statistical detectors, machine learning models, and LLM-assisted telemetry or log analysis modules.

This design choice separates anomaly identification from remediation synthesis. The proposed contribution does not claim that LLMs outperform state-of-the-art anomaly detection techniques. Rather, the focus of this work is the operational workflow that follows anomaly confirmation, namely contextual validation, retrieval of relevant historical knowledge, and constrained generation of draft remediation playbooks under explicit operational constraints.

Once an anomaly candidate is identified, the agent publishes the complete anomaly descriptor to the message broker. This event-driven notification mechanism ensures asynchronous and reliable communication with downstream components, particularly the Strategy Selector Agent, which performs secondary validation and mitigation planning as described in the next section.

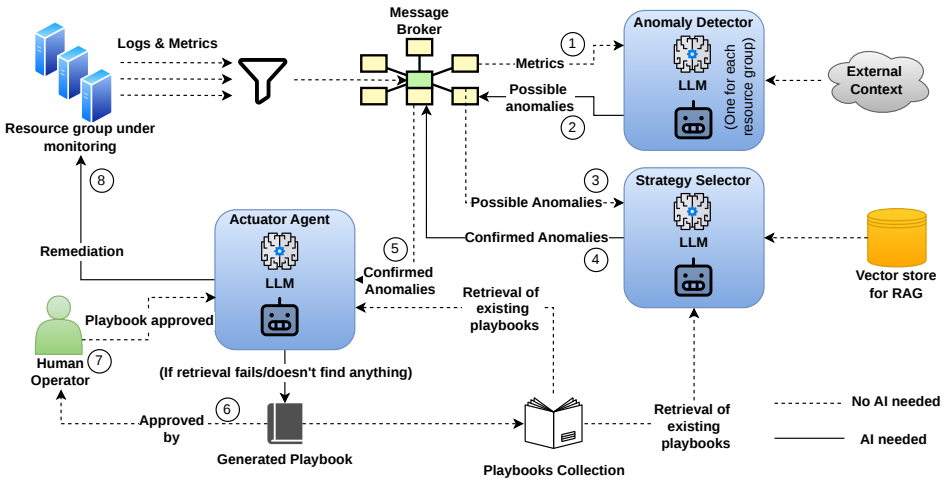

Fig. 1: Agentic LLM-Based Monitoring Architecture

B. Strategy Selector Agent

The Strategy Selector Agent functions as an intermediate validation and retrieval layer within the proposed pipeline. This component performs two critical operations: anomaly confirmation via historical correlation and retrieval of the mitigation strategy.

Upon receiving anomaly notifications from the Message Broker, the agent initiates a validation procedure by querying a vector database containing embeddings of previously detected and successfully mitigated anomalies within the current tenant. Each database entry stores anomaly metadata, including descriptor, timestamp, affected resources, and associated playbook. This structured representation facilitates multi-dimensional similarity assessment during retrieval operations. The vector store performs embedding-based semantic retrieval to identify structurally similar historical anomalies despite differences in surface-level manifestations.

Following successful confirmation of the anomaly, the Strategy Selector Agent employs a Retrieval-Augmented Generation (RAG) framework to search the Playbook Collection—a curated repository of playbooks using standards like CACAO [7] associated with known anomalies. The RAG pipeline retrieves relevant mitigation plays based on semantic similarity between confirmed anomaly characteristics and historical playbook metadata. When a suitable playbook is identified, the agent constructs a comprehensive anomaly report that contains the anomaly descriptor, validation metadata, and the retrieved playbook identifier. This structured report is then published to the Message Broker for consumption by the Actuator Agent.

In scenarios where no matching playbook exists in the repository, the Strategy Selector Agent generates an anomaly report without a playbook identifier, indicating the need for dynamic strategy synthesis. This condition triggers the Actuator Agent's generative capabilities, as detailed in Section III-C, enabling the system to address novel or previously unencountered anomalous conditions but also retrieve playbooks that need to be modified.

The dual-phase approach of confirmation followed by strategy retrieval ensures both accurate anomaly classification and efficient mitigation planning, while maintaining system adaptability through integration of generative fallback mechanisms.

C. Actuator Agent

The Actuator Agent serves as the execution layer of the proposed architecture, implementing mitigation strategies through automated remediation workflows. This component requires advanced reasoning capabilities provided by the underlying LLM, particularly for adaptive playbook generation and context-aware execution planning.

The operational logic of the Actuator Agent follows a hierarchical decision tree based on playbook availability and applicability. Upon receiving an anomaly report from the Strategy Selector Agent (Section III-B), the agent evaluates three distinct execution scenarios:

Direct Playbook Execution: When a playbook identifier is present in the anomaly report, the agent retrieves the corresponding remediation procedure from the Playbook Collection and initiates execution using the available tooling infrastructure.

Adaptive Playbook Modification: In cases where a relevant playbook exists but requires environment-specific customisation, such as updating IP addresses, adjusting commands for particular operating systems, or modifying resource identifiers, the Actuator Agent employs the LLM's reasoning capabilities to perform context-aware adaptations.

Novel Playbook Generation: When no suitable playbook exists, the agent enters a generative mode, synthesising a new mitigation strategy based on the anomaly characteristics and environmental constraints. This process leverages the LLM's capacity to reason about similar historical cases to construct logically sound remediation workflows.

For both adaptive modifications and novel playbook generation, the architecture incorporates a human-in-the-loop validation mechanism. Generated or modified playbooks are presented to authorised operators for review and approval prior to execution. This safety measure introduces an additional

validation layer for automatically generated remediation strategies before deployment in operational environments. Upon operator approval, the playbook is executed against the affected resources and, contingent upon successful mitigation, the validated playbook is persisted to the Playbook Collection for future reuse, thereby continuously enriching the system's knowledge base. This multi-modal execution strategy enables the Actuator Agent to balance automation efficiency with operational safety while progressively expanding the system's remediation capabilities through the accumulation of validated playbooks.

IV. Proof of Concept

The PoC evaluates the feasibility of transforming a confirmed anomaly into an actionable remediation artifact.

Specifically, it implements an *Actuator Agent* that ingests a previously confirmed anomaly generated by the *Strategy Selector Agent* and converts it into a structured response plan suitable for human validation and subsequent mitigation via a portable playbook artefact. The evaluation considers three locally deployed LLMs (detailed in the following analysis) and consumes anomaly information received from a message broker in JSON format.

The anomaly payload includes the contextual elements required for actionable remediation, namely: (i) anomaly severity, used to determine the degree of automation of the generated playbook and whether human approval is required; (ii) involved entities, such as hosts, services, or applications affected by the anomaly; (iii) evidences describing the anomaly source; (iv) predefined operational constraints acting as policies to guide LLM-based reasoning and prevent invalid or unsafe actions.

Upon receiving the anomaly payload, the Actuator Agent produces three outputs: (1) a structured response plan with a natural-language explanation, (2) a syntactic validation report, and (3) a CACAO-formatted draft playbook[1]. The use of the CACAO standard ensures portability, interoperability, and a well-defined action taxonomy including rollback procedures.

These artifacts are stored as a *Case* in a database and exposed to operators through an Approval UI for auditing and approval when required. The PoC intentionally avoids production-grade execution on real endpoints; the executor component is therefore stubbed. This design choice allows the evaluation to focus on the feasibility of LLM-driven planning, validation, and playbook synthesis under explicit constraints.

A. Implementation and Workflow

Figure 2 illustrates the implemented PoC architecture. A FastAPI ingestion service receives confirmed anomaly events through a REST endpoint and publishes them to a *Redis Streams* message broker, enabling asynchronous decoupling between ingestion and response generation.

A containerised worker process (the *Actuator Agent*) consumes the stream via a Redis consumer group, builds contextual information by querying a PostgreSQL-stored tool catalogue, and invokes a locally hosted LLM to generate a structured remediation plan.

The generated plan is validated against the schema (*json-schema*) and policy constraints, and converted into a CACAO draft playbook stored in PostgreSQL using SQLAlchemy (`psycopg`). A Streamlit-based Approval UI retrieves stored cases and presents anomaly details, generated plans, validation reports, and CACAO artifacts for human auditing. All components are deployed using containerised services to ensure reproducibility.

The implemented workflow proceeds as follows:

The PoC is publicly available through a GitHub repository[2].

B. Safety Guardrails and Operational Constraints

To reduce the risk of unsafe LLM-generated actions, the PoC implements deterministic guardrails around the planning process. These controls are enforced outside the LLM and therefore do not rely on model self-assessment. Generated plans must conform to a predefined JSON schema and to the CACAO playbook structure; malformed outputs are rejected and stored as failed cases. In addition, the tool catalogue defines the admissible action types and command templates that the Actuator Agent can reference, preventing the execution of commands outside the approved operational scope.

The system also applies policy-based filtering to restrict potentially disruptive actions on critical resources. Operations such as service restarts, firewall modifications, user lockouts, process termination, privilege changes, or resource deletions are treated as high-impact actions and require explicit human approval before execution. Generated playbooks must include rationale, affected assets, expected impact, and rollback information whenever disruptive actions are proposed.

Furthermore, the planner is constrained by severity and confidence metadata included in the anomaly payload. Low-severity or low-confidence anomalies cannot trigger disruptive remediation steps and are instead limited to diagnostic, containment, or evidence-collection actions. Finally, the executor in the current PoC is intentionally stubbed, meaning that no generated command is executed on real endpoints during evaluation. This design enables the assessment of constrained playbook generation without introducing uncontrolled modifications to production infrastructures.

C. Experimental Scenarios and Results

To evaluate the behaviour of the proposed workflow, three representative anomaly scenarios were considered: (i) a spike in CPU usage of a production database, (ii) a DNS beaconing pattern typically associated with malware command-and-control activity, and (iii) a Multi Factor Authentication (MFA) fatigue anomaly characterised by an abnormal volume of authentication attempts. Each scenario was encoded as a JSON anomaly payload and processed through the PoC pipeline. The evaluation was conducted using three locally deployed LLMs with

[1] https://docs.oasis-open.org/cacao/security-playbooks/v2.0/security-playbooks-v2.0.html

[2] https://github.com/fitnesslab-uniparthenope/PoC-for-An-Agentic-LLM-based-Architecture-for-Automated-Anomaly-Detection-and-Adaptive-Remediation-

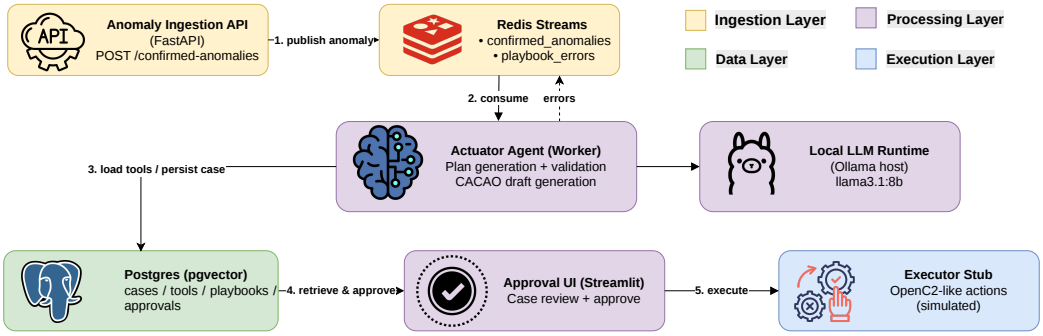

Fig. 2: PoC Agentic Anomaly Response Pipeline

comparable parameter scales: *Llama-3.1-8B*, *Qwen2.5:7B-Instruct*, and *Mistral-7B-Instruct-v0.3*.

Across the evaluated scenarios, the system consistently produced validated remediation playbooks with end-to-end generation times on the order of tens of seconds when executed on a local *MacBook M5*. The evaluation focused on two main aspects: the inference latency required to generate remediation plans and the quality of the resulting playbooks. Plan quality was measured using a *Plan Quality Score* (PQS), a composite metric that assesses multiple properties of the generated remediation strategies, including structural correctness, actionability, safety constraints, and compliance with the CACAO playbook format.

The evaluation targets the downstream remediation-planning stage of the proposed workflow, where confirmed anomaly payloads produced by upstream monitoring components are transformed into remediation artifacts. In this context, the experimental analysis assesses whether the Actuator Agent can generate syntactically valid, actionable, policy-compliant, and human-reviewable remediation playbooks from structured anomaly descriptions under explicit operational constraints.

The PQS is composed of five weighted components: (i) Coverage/Fit (0–30), which evaluates whether the generated plan addresses the expected remediation objectives for the considered anomaly class; (ii) Safety/Disruption (0–25), which evaluates the presence of unjustified disruptive actions, particularly when anomaly severity or confidence levels are limited; (iii) Completeness (0–20), which rewards procedural completeness and rollback support for disruptive operations; (iv) CACAO Validity (0–15), which measures compliance with the required CACAO structure; and (v) Actionability (0–10), which assesses whether the generated plan contains concrete executable or reviewable actions instead of generic recommendations. Plans that fail structural or policy validation are assigned a PQS of 0. The PQS is intended as an exploratory operational assessment metric for comparative evaluation within the proposed workflow, rather than as a standardised benchmark for remediation quality.

Figure 3 compares the LLM planning time with the corresponding PQS obtained for the evaluated models, highlighting the latency–quality trade-off. The results show that *Qwen2.5:7B-Instruct* generally achieves the lowest planning times but produces plans with relatively low PQS values, as indicated by the concentration of orange points in the lower-left region of the plot. In contrast, *Llama-3.1-8B* provides a more balanced

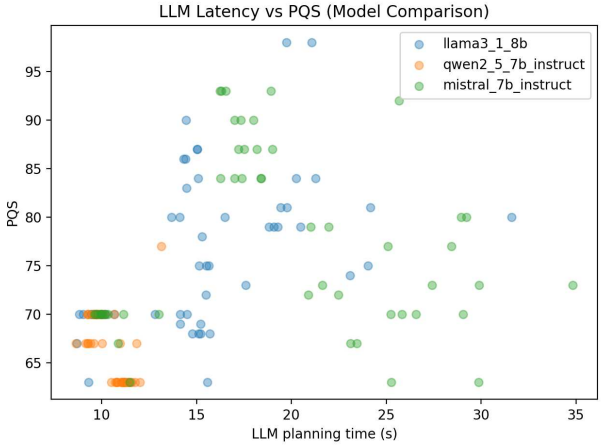

Fig. 3: Inference time compared to the Plan Quality Score of the different models used.

compromise between latency and plan quality, with most blue points distributed around the central area of the graph. Finally, *Mistral-7B-Instruct-v0.3* tends to require slightly longer inference times than Llama but achieves marginally higher PQS values on average, as reflected by the green cluster located in the upper-central portion of the plot. Overall, the results suggest that Llama offers the most favourable trade-off between planning efficiency and remediation quality within the evaluated workflow.

Figure 4 reports the breakdown of the PQS across the different anomaly categories considered in the evaluation. The results highlight how the performance of the evaluated models varies across anomaly types. In particular, *Mistral-7B-Instruct-v0.3* achieves the highest PQS values for authentication-related and availability anomalies, indicating stronger capability to generate effective remediation plans for these scenarios. Conversely, *Llama-3.1-8B* performs better in network-related anomalies, where it consistently produces higher-quality plans compared to the other models. As already suggested by the previous analysis, *Qwen2.5:7B-Instruct* consistently achieves lower PQS values across all anomaly categories, confirming its comparatively weaker remediation plan quality within the evaluated workflow.

Finally, Figure 5 summarises the PQS distribution across multiple anomaly instances. The boxplots highlight the stability of the generated remediation plans, showing consistent plan

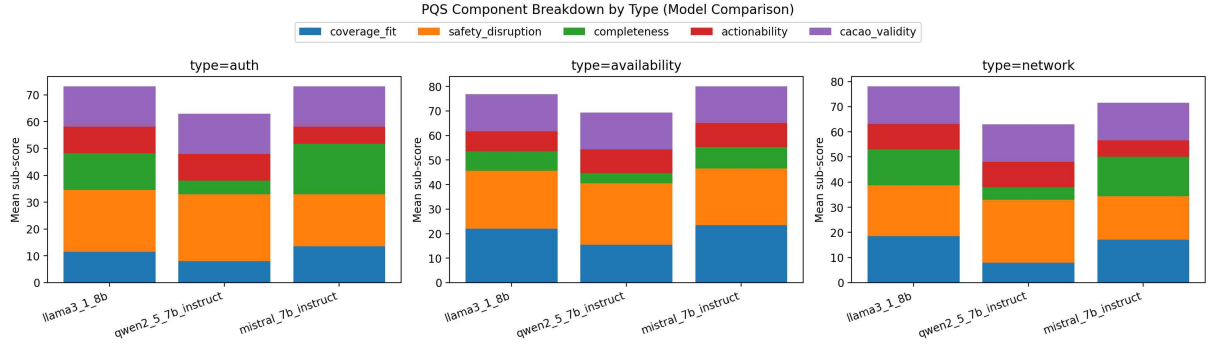

Fig. 4: Comparison of the different Plan Quality Scores breakdown divided by anomaly type for each used model.

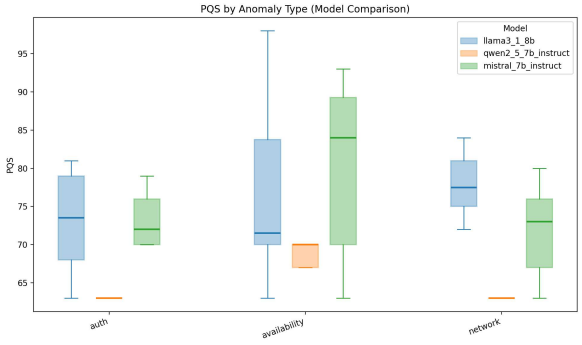

Fig. 5: Boxplot comparison of the different Plan Quality Scores across different anomaly receptions for each used model.

quality across anomaly instances. The experimental evaluation suggests that the proposed workflow can consistently transform confirmed anomalies into structured remediation playbooks under explicit operational constraints

V. Conclusion and Future Works

This paper presented an agentic AI-based architecture for adaptive remediation planning from confirmed cloud anomalies. The proposed MAS bridges the operational gap between anomaly confirmation and context-aware mitigation by transforming structured anomaly events into validated, CACAO-compatible remediation drafts. The architecture combines message-driven orchestration, retrieval of historical knowledge, LLM-based playbook synthesis, deterministic guardrails, and human-in-the-loop approval mechanisms.

The reported PoC demonstrates the feasibility of the downstream remediation-planning workflow, showing that confirmed anomalies can be transformed into structured remediation artifacts within operationally reasonable time bounds. The evaluation focuses on generating syntactically valid, policy-constrained, auditable, and human-reviewable remediation plans under explicit operational constraints.

Future work will focus on extending the experimental evaluation through larger-scale and longer-running campaigns to further assess the robustness and operational applicability of the proposed workflow. This includes reducing playbook-generation latency across different anomaly classes, integrating feedback-driven refinement mechanisms, and exploring automated rule- and playbook-generation strategies. Future developments will also investigate tighter integration with existing monitoring and orchestration platforms, enhanced semantic validation mechanisms, and comparative evaluations against non-agentic remediation approaches.

References

[1] Oluwatosin Oladayo Aramide. Autonomous network monitoring using llms and multi-agent systems. *World Journal of Advanced Engineering Technology and Sciences*, 2024.

[2] Marco Becattini, Roberto Verdecchia, and Enrico Vicario. Sallma: A software architecture for llm-based multi-agent systems. In *2025 IEEE/ACM International Workshop New Trends in Software Architecture (SATrends)*, pages 5–8. IEEE, 2025.

[3] Daniele Granata and Antonio Iannaccone. An ai-supported architecture for trusted cyber threat intelligence in industrial iot systems. In *ITASEC/SERICS*, 2026.

[4] Daniele Granata, Massimiliano Rak, and Giovanni Salzillo. Automated threat modeling approaches: Comparison of open source tools. *Communications in Computer and Information Science*, 1621 CCIS:250 – 265, 2022.

[5] Daniele Granata, Massimiliano Rak, Giovanni Salzillo, and Umberto Barbato. Security in iot pairing & authentication protocols, a threat model and a case study analysis. volume 2940, page 207 – 218. CEUR-WS, 2021.

[6] Yihong Jin, Ze Yang, Juntian Liu, and Xinhe Xu. Anomaly detection and early warning mechanism for intelligent monitoring systems in multi-cloud environments based on llm. In *2025 5th International Symposium on Computer Technology and Information Science (ISCTIS)*, 2025.

[7] Bret Jordan and Allan Thomson. Cacao security playbooks version 2.0. Oasis standard, OASIS Open, June 2023. Accessed: 2026-02-03.

[8] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.XXXXX*, 2024.

[9] Stefano Marrone, Francesco Flammini, Nicola Mazzocca, Roberto Nardone, and Valeria Vittorini. Towards model-driven v&v assessment of railway control systems. *International Journal on Software Tools for Technology Transfer*, 16(6):669–683, 2014.

[10] Kinjal Patel, Eshan Chandra Pandey, Inshu Misra, and Deepti Surve. Agentic ai for cloud troubleshooting: A review of multi agent system for automated cloud support. In *International Congress on Information and Communication Technology (ICICT)*, 2025.

[11] Georgios Syros, Anshuman Suri, Jacob Ginesin, Cristina Nita-Rotaru, and Alina Oprea. Saga: A security architecture for governing ai agentic systems. *arXiv preprint arXiv:2504.21034*, 2025.

[12] Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *arXiv preprint arXiv:2501.XXXXX*, 2025.

[13] Yingxuan Yang, Mulei Ma, Yuxuan Huang, Huacan Chai, Chenyu Gong, Haoran Geng, Yuanjian Zhou, et al. Agentic web: Weaving the next web with ai agents. *arXiv preprint arXiv:2501.XXXXX*, 2025.

[14] Talha Zeeshan, Abhishek Kumar, Susanna Pirttikangas, and Sasu Tarkoma. Large language model based multi-agent system augmented complex event processing pipeline for internet of multimedia things. *arXiv preprint arXiv:2501.XXXXX*, 2025.