

A Declarative Multi-Agent Framework for Reducing On-Call Operational Burden

Khushan Adatiya

Senior Software Engineer, Google

Email: khushan@gmail.com

Twinkle Hemant Kariya

Senior Software Engineer, Salesforce

Email: kariyatwinkle2@gmail.com

Abstract—Large-scale cloud-native systems generate continuous streams of operational alerts across distributed microservice architectures. On-call engineers must manually triage these alerts by correlating signals from heterogeneous observability tools, a process that is time-consuming, cognitively demanding, and prone to error. Despite advances in monitoring and anomaly detection, incident triage remains largely manual.

This paper presents a declarative, large language model (LLM)-driven multi-agent approach to automating incident triage and Service Level Objective (SLO) monitoring. The proposed design constraints agent behavior using domain-expert authored investigation workflows, enabling deterministic execution and reproducibility while preserving operational safety. The framework integrates a unified tool execution layer for interacting with diverse observability systems and an enhanced retrieval-augmented generation (RAG) pipeline optimized for operational knowledge retrieval.

The approach has been evaluated in a production cloud environment spanning multiple microservices and geographic regions. Results show reductions in high-severity incident triage time from approximately 30 minutes to under 5 minutes, alert acknowledgement latency from minutes to seconds, and service onboarding effort from weeks to days. These findings suggest that constrained multi-agent systems can substantially reduce on-call cognitive load while maintaining reliability and human oversight.

Index Terms—AIOps, on-call engineering, incident triage, multi-agent systems, large language models

I. INTRODUCTION

Cloud platforms increasingly rely on distributed microservice architectures to achieve scalability and fault isolation. While effective for system evolution, this architectural shift has significantly increased operational complexity. A single user visible incident may involve

multiple services, infrastructure layers, and failure modes, requiring rapid investigation under time pressure.

On-call engineers are responsible for diagnosing these incidents by interpreting alerts, querying metrics and logs, consulting historical incidents, and coordinating responses. This process is often repetitive and relies heavily on tacit knowledge accumulated through experience. As a result, on-call duties contribute to elevated mean time to resolve (MTTR), alert fatigue, and engineer burnout.

Recent advances in large language models have motivated interest in agent-based approaches for operational automation [1]. However, unconstrained agent behavior introduces

risks related to non-determinism, hallucinated actions, and deviation from established operational procedures. In production on-call environments, such risks are unacceptable.

This paper investigates whether LLM-based agents can be applied safely to on-call operations when constrained by declarative workflows and deterministic execution controls. We present a general framework and empirical evaluation demonstrating that such constraints enable meaningful automation of incident triage while preserving operational safety.

II. OPERATIONAL CHALLENGES IN ON-CALL ENGINEERING

A. Incident Triage as a Bottleneck

Incident triage accounts for a large fraction of on-call effort. Engineers must determine whether an alert represents a genuine service degradation, identify affected components, and gather sufficient context to guide remediation. These steps are largely diagnostic rather than corrective, yet they consume critical time during incidents.

B. Cognitive Load and Alert Fatigue

On-call work is characterized by high cognitive load, particularly during off-hours. Engineers must reason about unfamiliar services, incomplete information, and ambiguous signals. Frequent false positives and

redundant alerts further exacerbate fatigue and reduce response effectiveness. Steps cannot be skipped, reordered, or substituted. This plan fidelity enforcement prevents unintended behavior and enables reproducible investigations.

C. Limitations of Existing Automation

Existing automation primarily focuses on alert generation or anomaly detection [2]. While useful, these systems do not perform structured investigations or synthesize findings into actionable summaries. As a result, human operators remain responsible for most triage activities.

D. Execution Safety and Auditing

All tool interactions are mediated by a deterministic execution layer. Agents operate in a read-only mode, and all actions are logged for auditability. This design supports post-incident analysis and compliance requirements.

III. DESIGN GOALS

The proposed framework is guided by the following design goals:

- Determinism: Agent behavior must be predictable and reproducible.
- Alignment with operational practice: Automation should reflect existing investigation procedures.
- Tool abstraction: The system must operate across heterogeneous observability platforms.

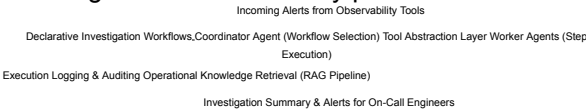


Fig. 1. High-level system architecture (half-page fit)

- Human oversight: Final decisions and remediation actions remain with human operators.
- Low onboarding cost: New services should be integrated without modifying agent logic.

IV. SYSTEM ARCHITECTURE

A. Declarative Investigation Workflows

Investigation logic is defined using declarative workflow specifications authored by domain experts [3]. Each workflow encodes an ordered sequence of investigation steps, required inputs, permitted tools, and expected outputs. This approach separates operational knowledge from agent implementation.

B. Multi-Agent Execution Model

The system follows a manager–worker agent model. A coordinating agent selects an appropriate workflow based on incoming alerts and orchestrates execution. Worker agents perform individual investigation steps, such as querying metrics or retrieving logs, and return structured results.

C. Plan Fidelity Enforcement

To ensure adherence to operational procedures, agent actions are validated against the workflow specification.

V. TOOL ABSTRACTION LAYER

Operational environments typically employ multiple observability systems with distinct query languages and interfaces. The framework introduces a unified execution layer that abstracts these systems behind a common interface.

Each tool implementation conforms to a standardized contract, allowing workflows to remain agnostic to underlying platforms. This abstraction simplifies integration and improves portability across environments.

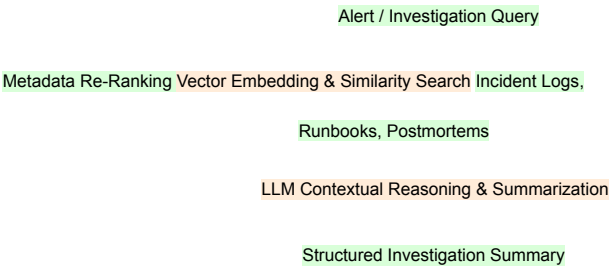


Fig. 2. Retrieval-Augmented Generation (RAG) pipeline for operational knowledge enrichment

TABLE I
INCIDENT TRIAGE PERFORMANCE COMPARISON

Approach	Triage Time	Ack Latency	Onboarding	Manual On-call
Rule-based Automation	~15 min	~1 min	Weeks	~30 min
Proposed Framework	<5 min	Seconds	Days	

VI. OPERATIONAL KNOWLEDGE RETRIEVAL

Incident investigations often rely on historical context such as past incidents, runbooks, and postmortems. The framework incorporates a retrieval-augmented generation (RAG) pipeline to surface relevant knowledge during investigations [4].

To improve retrieval precision, the system combines vector similarity search with query-time metadata re-ranking based on alert attributes and workflow context. This approach reduces irrelevant results and improves the usefulness of retrieved context.

VII. EVALUATION

A. Deployment Context

The framework was evaluated in a production cloud environment spanning multiple microservices deployed across several geographic regions.

B. Metrics

Evaluation focused on incident triage duration, alert acknowledgement latency, and effort required to onboard new services.

C. Results

Across evaluated high-severity incidents, median summaries rather than raw telemetry, enabling faster and more confident decision-making.

Importantly, the system functions as a decision-support mechanism rather than an autonomous remediation system, preserving human control over production environments.

IX. LIMITATIONS AND FUTURE WORK

The evaluation is limited to a single organizational context and does not include controlled user studies. The system currently operates in a read-only mode. Future work includes broader evaluations, formal user studies on cognitive load reduction, and cautious exploration of supervised remediation.

X. RELATED WORK

Prior work in AIOps has focused on anomaly detection, alert correlation, and log analysis [2]. Recent research explores LLM-based operational agents, often emphasizing autonomy. This work differs by prioritizing constrained execution, reproducibility, and alignment with established on-call practices.

XI. CONCLUSION

This paper demonstrates that declarative, constrained multi agent systems can meaningfully reduce on-call operational burden in large-scale cloud environments. By combining LLM-based reasoning with deterministic execution controls and tool abstraction, the proposed approach achieves measurable efficiency gains while maintaining safety and human oversight.

REFERENCES

- [1] X. Chen, Y. Zhang, and P. Liu, "AIOps in Cloud Operations: Automation and Challenges," *IEEE Cloud Computing*, vol. 10, no. 2, pp. 25–35, 2023.
- [2] J. Xu, A. Kumar, and M. Zhou, "Alert Correlation and Reduction Techniques in Large-Scale Systems," *ACM Computing Surveys*, vol. 52, no. 3, pp. 1–34, 2019.
- [3] L. Zhang, H. Li, and K. Wang, "Declarative Workflows for Automated Incident Management," in *Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2022, pp. 110–118.
- [4] P. Lewis, E. Perez, and S. Khandelwal, "Retrieval-Augmented

triage time decreased from approximately 30 minutes to under 5 minutes. Alert acknowledgement latency decreased from minutes to seconds, and service onboarding effort decreased from weeks to days. These improvements indicate a substantial reduction in repetitive investigative work during on-call incidents.

VIII. IMPACT ON ON-CALL ENGINEERS

By automating data gathering and preliminary analysis, the framework reduces the cognitive burden placed on on-call engineers. Engineers interact with structured investigation

Generation for Knowledge-Intensive NLP Tasks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.