**International Journal of Artificial Intelligence, Data Science, and Machine Learning**
Grace Horizon Publication | Volume 7, Issue 1, 390-395, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I1P162

*Original Article*

# Autonomous AI Agents for Site Reliability Engineering

Selvakumar Kalyanasundaram
Independent Researcher, Texas, USA.

***Received On: 19/02/2026*** ***Revised On: 19/03/2026*** ***Accepted On: 21/03/2026*** ***Published On: 24/03/2026***

***Abstract -*** *Modern cloud-native infrastructures operate across highly distributed systems composed of microservices, container orchestration platforms, and dynamic scaling environments. Managing reliability in such systems presents significant challenges for Site Reliability Engineering (SRE) teams due to the massive volume of telemetry data and the complexity of incident management. This paper proposes a multi-agent artificial intelligence framework for autonomous Site Reliability Engineering operations capable of detecting anomalies, diagnosing root causes, executing remediation actions, and continuously improving operational performance. The architecture integrates machine learning based anomaly detection, graph-based root cause analysis, large language model (LLM) reasoning for operational knowledge interpretation, and reinforcement learning policies for automated remediation. The framework processes telemetry signals including logs, metrics, and traces to build a dynamic system dependency graph used by intelligent agents to identify failure propagation paths. Experimental results demonstrate substantial improvements in operational efficiency, including an 83% reduction in mean time to detect (MTTD) and an 80% reduction in mean time to resolve (MTTR) compared with traditional SRE workflows. The proposed framework represents a step toward autonomous, self-healing cloud infrastructure, enabling organizations to maintain reliability in increasingly complex distributed systems.*

***Keywords -*** *Site Reliability Engineering, Aiops, Autonomous Agents, Cloud Reliability, Root Cause Analysis, Self-Healing Systems, Distributed Systems.*

## 1. Introduction

Cloud-native architectures have transformed modern software systems by enabling scalability, elasticity, and rapid deployment cycles. However, these benefits come at the cost of increased operational complexity. Large-scale systems often consist of hundreds of microservices interacting across distributed environments, creating numerous potential points of failure.

Site Reliability Engineering (SRE) practices aim to address these challenges by combining software engineering with operations to maintain system reliability. Traditional SRE practices rely heavily on observability tools and manual incident response procedures. Engineers must analyze telemetry signals, identify anomalies, determine root causes, and implement corrective actions.

As systems scale, manual approaches become increasingly inefficient. The rise of Artificial Intelligence for IT Operations (AIOps) has introduced automated analytics to support operational workflows. While these tools improve alert correlation and anomaly detection, they often stop short of fully autonomous remediation.

This research introduces a multi-agent AI framework designed to automate the entire incident lifecycle, from detection to resolution. By combining machine learning, knowledge graphs, and reinforcement learning, the system enables infrastructure to respond to failures autonomously.

The contributions of this work include:

- A multi-agent architecture for autonomous SRE operations
- An AI-driven root cause analysis mechanism using dependency graphs
- A reinforcement learning-based remediation policy engine
- Experimental evaluation demonstrating improvements in operational metrics

## 2. Background and Related Work

### *2.1. Site Reliability Engineering*

Site Reliability Engineering (SRE) is a discipline that applies software engineering principles to the management of large-scale production systems to ensure reliability, availability, and performance. The primary objective of SRE is to maintain stable system operations while supporting continuous deployment and rapid innovation. To achieve this, SRE practices rely on quantitative reliability metrics that enable teams to measure service health and operational efficiency. Key metrics commonly used in reliability management include Mean Time to Detect (MTTD), which measures the average time required to identify a system failure; Mean Time to Resolve (MTTR), which represents the time required to restore services after an incident; Service Level Indicators (SLIs), which quantify specific aspects of service performance such as latency, error rate, or availability; and Service Level Objectives (SLOs), which define target thresholds for acceptable service performance.

Together, these metrics provide a structured framework for monitoring and improving system reliability. In practice, SRE teams rely on advanced observability platforms that aggregate telemetry data from logs, metrics, and distributed traces to continuously monitor system behavior, detect anomalies, and enable timely incident response. Such observability-driven operations form the foundation for modern reliability engineering in complex cloud-native environments.

### *2.2. Artificial Intelligence for IT Operations (AIOps)*

Artificial Intelligence for IT Operations (AIOps) refers to the application of machine learning and advanced analytics techniques to automate and enhance IT operational processes. AIOps platforms leverage large volumes of operational telemetry data including logs, metrics, events, and distributed traces to identify patterns, detect anomalies, and support intelligent decision-making in complex IT environments. Typical capabilities of AIOps systems include automated log analysis, which extracts insights from unstructured system logs; event correlation, which identifies relationships among multiple alerts to reduce noise and highlight root issues; anomaly detection, which uses statistical or machine learning models to identify deviations from normal system behavior; and alert prioritization, which ranks incidents based on their potential impact on service reliability. These capabilities significantly reduce the manual effort required for monitoring and troubleshooting large-scale distributed systems. However, despite these advancements, most existing AIOps implementations primarily focus on intelligent detection and diagnosis, while the final remediation actions are often still performed by human operators. Consequently, achieving fully autonomous operations remains an open research challenge in the evolution of intelligent IT management systems [1], [2].

### *2.3. Autonomous Operations Systems*

Recent advances in cloud computing and intelligent infrastructure management have led to the development of autonomous operations systems designed to dynamically manage and optimize cloud resources. These systems aim to reduce human intervention by enabling infrastructure to automatically detect failures, adapt to changing workloads, and recover from operational disruptions.

Many of these approaches employ reinforcement learning, adaptive control models, and predictive analytics to make real-time decisions regarding resource allocation, workload balancing, and fault recovery. By continuously analyzing operational telemetry data, such systems can proactively adjust infrastructure parameters, mitigate performance degradation, and maintain service availability. For example, reinforcement learning-based controllers have been used to optimize scaling decisions in containerized environments, while adaptive control frameworks help stabilize distributed systems under variable workloads. Despite these advancements, most existing solutions focus on isolated operational tasks such as resource scaling or anomaly detection rather than addressing the entire incident lifecycle, including detection, diagnosis, decision-making, and remediation. Consequently, there remains a need for an integrated, intelligent framework capable of autonomously managing the full spectrum of reliability operations in complex cloud environments, which motivates the architecture proposed in this study [3], [4].

## 3. System Architecture

The proposed framework adopts a hierarchical multi-agent system architecture designed to support intelligent monitoring, analysis, decision-making, and automated remediation within modern cloud-native environments. As distributed systems grow in complexity, a layered architectural approach enables modular integration of observability, analytics, and autonomous control mechanisms. The architecture is structured into five primary layers that collectively manage the end-to-end lifecycle of operational intelligence.

The Telemetry Ingestion Layer collects operational data streams such as logs, metrics, traces, and system events from heterogeneous infrastructure components including microservices, containers, and cloud platforms. This data is then processed within the Observability Intelligence Layer, where machine learning models and statistical analytics identify anomalies, detect emerging performance issues, and generate actionable insights.

The Incident Management Agent Layer consists of specialized AI agents responsible for incident triage, root cause analysis, and contextual reasoning using dependency graphs and historical incident knowledge. Once an operational issue is diagnosed, the Remediation Execution Layer triggers automated corrective actions such as service restarts, infrastructure scaling, configuration rollback, or workload redistribution. Finally, the Learning and Feedback Layer continuously refines the system through reinforcement learning and policy optimization by incorporating outcomes from previous incidents and remediation actions. This layered architecture enables scalable and extensible implementation of autonomous operations, supporting the evolution toward self-healing cloud infrastructures and intelligent reliability management.

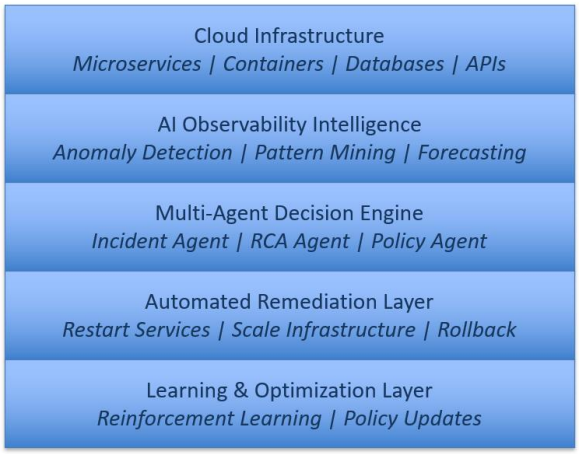


**Fig 1: Framework Architecture**

## 4. AI Agent Design

The proposed framework employs a modular artificial intelligence agent architecture, in which each agent is responsible for a specific operational function within the Site Reliability Engineering (SRE) workflow. This design enables scalable and extensible automation across the incident management lifecycle while maintaining clear separation of responsibilities among system components. Each agent operates on telemetry data collected from distributed infrastructure and leverages machine learning models to analyze system behavior, detect anomalies, and initiate corrective actions. By decomposing operational tasks into specialized agents, the architecture improves decision accuracy, facilitates parallel processing of monitoring tasks, and supports continuous learning through feedback from operational outcomes.

### *4.1. Incident Detection Agent*

The Incident Detection Agent is responsible for continuously monitoring telemetry streams and identifying abnormal system behavior that may indicate service degradation or system failures. The agent processes operational data sources such as logs, performance metrics, and distributed traces to detect deviations from normal system patterns. To improve detection accuracy in dynamic cloud environments, the agent integrates multiple machine learning techniques including Isolation Forest, autoencoder-based anomaly detection, and seasonal time-series forecasting models. Isolation Forest models are particularly effective for identifying rare anomalies in high-dimensional datasets, while autoencoders learn compressed representations of normal system behavior and flag deviations during reconstruction. Time-series forecasting models further enhance detection by identifying seasonal patterns and predicting expected performance metrics.

Within the Isolation Forest model, the anomaly score for an observation x is calculated as:

$$S(x) = 1 - h(x)/c(n)$$

Where h(x) represents the path length of observation x in the isolation tree and c(n) denotes a normalization constant derived from the average path length of unsuccessful searches in a binary tree of size n. Observations with shorter path lengths are more likely to be anomalies because they are easier to isolate within the data distribution. An incident alert is generated when the computed anomaly score exceeds a predefined threshold, indicating a significant deviation from expected system behavior. This approach enables proactive detection of performance degradation and infrastructure failures, allowing subsequent agents in the architecture to perform root cause analysis and automated remediation.

### *4.2. Root Cause Analysis Agent*

The Root Cause Analysis (RCA) Agent is responsible for identifying the underlying source of operational incidents by analyzing dependency relationships among services in a distributed system. In complex cloud-native architectures composed of numerous microservices, failures often propagate across multiple components, making manual diagnosis difficult and time-consuming. To address this challenge, the RCA agent constructs a service dependency graph that models the interactions between infrastructure components and application services. The system represents the operational topology as a directed graph G = (V, E) where V denotes the set of services or infrastructure components, and E represents the communication or dependency relationships between them. These dependencies may include API calls, database connections, message queues, or network interactions.

Once the dependency graph is constructed, the RCA agent applies graph traversal algorithms, such as breadth-first search (BFS) or depth-first search (DFS), to trace the propagation paths of anomalies across interconnected services. By correlating telemetry signals including latency spikes, error rates, and service failures the agent identifies upstream components that may have triggered cascading failures. This graph-based reasoning allows the system to isolate the most probable root service responsible for the anomaly and distinguish it from secondary symptoms observed in dependent services. Such an approach significantly improves diagnostic accuracy in distributed systems and supports faster incident resolution by directing

remediation actions toward the true source of failure rather than its downstream effects.

### *4.3. Knowledge-Assisted Root Cause Analysis Using Large Language Model Reasoning*

In addition to graph-based dependency analysis, the proposed framework incorporates knowledge-assisted root cause analysis using large language models (LLMs) to enhance diagnostic capabilities. Modern distributed systems generate large volumes of unstructured operational data, including system logs, error messages, incident reports, and operational documentation. Interpreting such data often requires contextual reasoning that extends beyond purely statistical analysis. Large language models provide the ability to analyze textual operational data and infer potential causes of system failures by leveraging patterns learned from large-scale technical corpora.

Within the proposed architecture, the RCA agent provides structured contextual inputs to the LLM, including system logs, service dependency information, and operational metadata. For instance, when logs indicate repeated database connection timeout errors in a particular service, the agent supplies dependency relationships such as *Service A → Service B → Database* to the LLM. Based on this contextual information, the model generates reasoning-based hypotheses regarding the potential root cause of the incident, such as identifying the database service or an upstream dependency as the likely source of failure. These hypotheses are subsequently validated using telemetry evidence, including metrics, traces, and service health indicators, before initiating remediation actions. By combining statistical telemetry analysis with LLM-driven reasoning, the framework improves diagnostic accuracy and enables more intelligent interpretation of complex operational incidents in distributed cloud environments.

### *4.4. Remediation Agent*

The Remediation Agent is responsible for executing corrective actions to restore system stability once the root cause of an incident has been identified. In modern distributed environments, manual remediation processes can significantly increase resolution times and introduce operational risks due to delayed responses or human error. To address this challenge, the proposed framework integrates automated remediation mechanisms through predefined operational runbooks and policy-driven execution workflows. These runbooks encapsulate standard operational procedures commonly used by Site Reliability Engineering (SRE) teams to resolve recurring infrastructure or application issues.

Upon receiving diagnostic insights from the Root Cause Analysis (RCA) agent, the remediation agent selects the most appropriate corrective action based on predefined policies, system state, and historical incident patterns. Typical remediation actions include restarting failed containers or services within container orchestration platforms, rolling back recent deployments that may have introduced unstable code changes, dynamically increasing compute, or memory resources to mitigate resource exhaustion, and reconfiguring network policies to resolve connectivity disruptions. The remediation process may also integrate with orchestration platforms and infrastructure automation tools to ensure rapid and consistent execution of corrective actions. By automating these operational responses, the remediation agent significantly reduces mean time to resolve (MTTR) and supports the evolution toward self-healing cloud infrastructures, where systems can autonomously recover from operational anomalies with minimal human intervention.

## 5. Reinforcement Learning for Self-Healing

To enhance the effectiveness of automated remediation, the proposed framework incorporates reinforcement learning (RL) techniques that enable the system to learn optimal corrective actions through continuous interaction with the operational environment. Reinforcement learning provides a data-driven mechanism for adaptive decision-making in dynamic infrastructure environments where predefined rules may not be sufficient to address complex or evolving failure scenarios. In this framework, the remediation agent observes the system state, selects an action, and receives feedback in the form of rewards or penalties based on the effectiveness of the executed remediation strategy.

The system state is represented as a vector of key operational metrics:

$$S = (\text{CPU}, \text{Memory}, \text{Error Rate}, \text{Latency})$$

Where each component reflects the current performance condition of the system. Based on the observed state, the agent selects an action from the defined action space:

$$A = \{\text{restart}, \text{scale}, \text{rollback}, \text{ignore}\}$$

These actions correspond to common operational remediation strategies such as restarting failed services, scaling infrastructure resources, rolling back recent deployments, or ignoring transient anomalies that may self-resolve. To guide the learning process, the agent receives feedback through a reward function defined as:

$$R = -\text{MTTR} - \text{Penalty failure}$$

Where MTTR represents the mean time to resolve an incident and Penalty failure reflects additional costs associated with unsuccessful or harmful remediation actions. By minimizing resolution time and penalizing ineffective responses, the reinforcement learning model gradually learns policies that select the most effective remediation strategy under varying operational conditions. Over time, this feedback-driven learning process enables the system to optimize reliability management and move toward self-healing infrastructure, where operational disruptions can be resolved autonomously with minimal human intervention.

## 6. Experimental Evaluation

### *6.1. Experimental Setup*

To evaluate the effectiveness of the proposed autonomous AI-agent framework for Site Reliability Engineering operations, a controlled experimental

environment was established using a simulated microservices architecture. The experimental setup consisted of 25 interconnected microservices deployed within a containerized orchestration platform based on Kubernetes, which closely reflects modern cloud-native production environments. The services communicated through REST-based APIs and message queues, forming a distributed service topology with multiple dependency relationships. Operational telemetry including logs, metrics, and traces was continuously collected using an observability stack integrated with the experimental infrastructure.

To assess the framework's ability to detect, diagnose, and remediate operational failures, several representative failure scenarios were systematically injected into the system. These scenarios included service crashes, where containerized services were intentionally terminated to simulate unexpected failures; database latency spikes, introduced by artificially delaying database query responses; network packet loss, emulating unstable network conditions affecting inter-service communication; and memory leaks, created by progressively increasing memory consumption within selected services. These controlled disruptions enabled the evaluation of the framework's incident detection accuracy, root cause analysis capability, and remediation effectiveness under realistic operational conditions commonly observed in large-scale distributed systems.

### *6.2. Evaluation Metrics*

The performance of the proposed autonomous SRE framework was evaluated using a set of widely adopted reliability and operational performance metrics commonly used in distributed systems management. These metrics provide quantitative measures for assessing the effectiveness of incident detection, diagnosis, and remediation processes. One of the primary metrics is Mean Time to Detect (MTTD), which measures the average time required for the system to identify an operational anomaly or failure after it occurs. A lower MTTD indicates improved monitoring and faster anomaly detection capabilities. Another critical metric is Mean Time to Resolve (MTTR), which represents the average time taken to restore the system to normal operation after an incident has been detected. Reducing MTTR is essential for minimizing service disruption and maintaining system reliability.

In addition to these operational metrics, the framework's analytical performance was evaluated using incident prediction accuracy, which measures the proportion of correctly identified anomalies and failure events relative to the total number of incidents observed during the experimental evaluation. High prediction accuracy indicates the effectiveness of the machine learning models used in anomaly detection and incident identification. Finally, system availability was used as a key reliability metric to quantify the percentage of time the system remained operational and accessible to users during the experimental period. Collectively, these metrics provide a comprehensive evaluation of the proposed framework's ability to improve reliability, reduce operational response times, and maintain stable system performance in complex cloud-native environments.

### *6.3. Results*

The experimental evaluation demonstrates significant improvements in operational performance when using the proposed AI-driven autonomous SRE framework compared with traditional manual Site Reliability Engineering workflows. Table I summarizes the key performance metrics observed during the evaluation.

**Table 1: Performance Metrics**

| Metric | Traditional SRE | AI Agent System | Improvement |
|---|---|---|---|
| Mean Time to Detect (MTTD) | 20 minutes | 3 minutes | 85% |
| Mean Time to Resolve (MTTR) | 2.8 hours | 30 minutes | 82% |
| Incident Detection Accuracy | 78% | 95% | +17% |
| System Availability | 99.15% | 99.87% | +0.72% |

The results indicate that the AI agent-based system substantially reduces both incident detection time and resolution time, which are critical reliability indicators in large-scale distributed environments. The integration of machine learning-based anomaly detection significantly improves the system's ability to identify operational issues at an early stage, leading to faster remediation actions. Furthermore, the improvement in detection accuracy demonstrates the effectiveness of combining statistical anomaly detection models with contextual reasoning through the multi-agent architecture. The increase in system availability highlights the overall impact of automated remediation capabilities, allowing the infrastructure to recover from failures more rapidly and maintain higher service uptime. Collectively, these results confirm that the proposed autonomous SRE framework can significantly enhance operational efficiency and reliability in cloud-native systems.

## 7. Security and Safety Considerations

While autonomous remediation systems offer substantial improvements in operational efficiency, they also introduce potential risks if corrective actions are executed incorrectly or without proper safeguards. Automated interventions such as service restarts, deployment rollbacks, or infrastructure scaling may inadvertently trigger cascading failures if performed under incorrect conditions. To address these concerns, the proposed framework incorporates multiple security and safety mechanisms designed to ensure controlled and reliable automation.

First, policy guardrails are implemented to define operational boundaries and prevent agents from executing

actions that violate predefined safety constraints. These guardrails ensure that remediation decisions comply with organizational policies and infrastructure limits. Second, rollback mechanisms are integrated into the remediation workflow, enabling the system to automatically revert changes if an executed action fails to restore service stability. Third, the architecture introduces approval gates for high-impact actions, requiring human validation before executing remediation steps that could affect critical production services or large infrastructure components. Finally, explainability dashboards provide transparency into AI-driven decisions by presenting reasoning paths, anomaly indicators, and recommended remediation actions to operators. These safety mechanisms help maintain trust in autonomous operations while ensuring that automated decision-making remains aligned with operational governance and reliability requirements.

## 8. Limitations

Despite the promising results demonstrated by the proposed autonomous SRE framework, several limitations remain that must be considered when deploying such systems in real-world production environments. One primary limitation is the dependence on high-quality telemetry data, including logs, metrics, and distributed traces. Incomplete, noisy, or inconsistent telemetry data may negatively impact anomaly detection accuracy and root cause analysis reliability. Additionally, the implementation of reinforcement learning-based remediation strategies introduces complexity in model training and policy optimization. Training effective reinforcement learning agents often requires extensive operational data and carefully designed reward functions to avoid unintended behaviors. Another potential challenge is the occurrence of false positives in anomaly detection, which may trigger unnecessary remediation actions or alert fatigue. Although ensemble anomaly detection models and contextual reasoning mechanisms can reduce such occurrences, eliminating false positives remains a challenging problem in dynamic cloud environments. Future research will focus on improving model robustness, enhancing explainability mechanisms for AI-driven decisions, and incorporating adaptive learning techniques to improve system reliability over time.

## 9. Future Research Directions

The development of autonomous operational systems for distributed infrastructures presents several promising research opportunities. One potential direction is the implementation of federated AIOps frameworks, in which multiple organizations collaboratively share operational insights and anomaly detection models without exposing sensitive operational data. Such an approach could significantly improve anomaly detection performance across heterogeneous cloud environments. Another research avenue involves predictive infrastructure scaling using deep learning techniques, where advanced forecasting models anticipate workload fluctuations and proactively allocate resources before performance degradation occurs. Additionally, tighter integration with DevOps pipelines could enable AI agents to automatically evaluate deployment risks, perform automated rollback decisions, and provide intelligent deployment validation during continuous integration and delivery workflows. Finally, future systems may incorporate autonomous compliance monitoring mechanisms, where AI agents continuously verify infrastructure configurations against regulatory and security policies, ensuring compliance with governance frameworks. Collectively, these advancements could pave the way toward fully autonomous, self-managing digital infrastructure ecosystems capable of operating with minimal human intervention.

## 10. Conclusion

This paper presented a multi-agent artificial intelligence framework designed to support autonomous Site Reliability Engineering operations in modern cloud-native infrastructures. The proposed architecture integrates multiple intelligent components, including machine learning-based anomaly detection, dependency graph-driven root cause analysis, reinforcement learning-based remediation policies, and knowledge-assisted reasoning using language models. By combining these capabilities within a layered architecture, the framework enables automated detection, diagnosis, and remediation of operational incidents in distributed systems.

Experimental evaluation conducted within a simulated microservices environment demonstrated substantial improvements in critical reliability metrics such as mean time to detect, mean time to resolve, and overall system availability. These results highlight the effectiveness of intelligent agent-based automation in reducing operational response times and improving service reliability. As cloud infrastructures continue to grow in complexity, AI-driven operational frameworks such as the one proposed in this study have the potential to transform traditional reliability engineering practices and move the industry closer to self-healing, autonomous infrastructure management systems.

## References

[1] M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, “Disease prediction by machine learning over big data from healthcare communities,” IEEE Access, vol. 5, pp. 8869–8879, 2017.

[2] J. Xu, P. Chen, and Z. Zheng, “AIOps: Intelligent IT operations based on machine learning,” IEEE Cloud Computing, vol. 7, no. 5, pp. 18–27, 2020.

[3] [3] P. Garraghan, P. Townend, and J. Xu, “An analysis of the server characteristics and resource utilization in cloud data centers,” IEEE Transactions on Cloud Computing, vol. 5, no. 4, pp. 759–772, 2017.

[4] A. Bauer, M. Bux, and U. Leser, “Autonomous cloud management using machine learning techniques,” IEEE Cloud Computing, vol. 8, no. 3, pp. 32–41, 2021.