International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

From AIOps to Autonomous Operations: An Agentic AI Architecture for Self-Healing Enterprise IT Infrastructure

Mihir Patel

Enterprise Architect, Tata Consultancy Services, Olympus, Thane, Maharashtra, India

Abstract: ***Enterprise IT estates spanning data centres, multiple clouds, networks and end-user devices have outgrown reactive operating models. AIOps platforms have improved event correlation and anomaly detection, yet in most enterprises they stop at insight, leaving diagnosis and remediation to engineers. This paper proposes a vendor-neutral reference architecture that extends AIOps towards autonomous, self-healing operations using agentic AI. The architecture unifies observability data, IT service management records, configuration and topology information and historical incident knowledge into an operational data layer. Specialised agents for detection, correlation, root cause analysis, remediation planning, execution and validation work on this layer. Large language models support reasoning and runbook interpretation, while every action runs through governed automation workflows. A four-stage maturity model is defined, progressing from reactive to predictive, preventive and autonomous operations, along with governance controls such as risk-based approval thresholds, human-in-the-loop checkpoints, change management integration, rollback and audit trails. Application scenarios across Service Desk, End User Computing, Data Centre, Network and NOC operations are discussed. The approach offers enterprises and managed service providers a practical path to reduce manual effort, shorten incident resolution and move IT operations from firefighting to proactive service assurance.***

Keywords: AIOps, Agentic AI, Self-Healing Infrastructure, Autonomous IT Operations, IT Service Management

1. Introduction

Enterprise IT infrastructure has changed considerably over the last decade. A typical large organisation now runs workloads across on-premise data centres, one or more public clouds, software-defined networks and a large, distributed base of end-user devices. Each layer produces its own telemetry, is often monitored by its own tools and is frequently supported by separate operations teams. The result is an operating environment where the volume of events is far greater than what people can reasonably assess, and where the root cause of a service disruption is often several layers away from where the symptom first appears.

Most IT operations organisations still run on a largely reactive model. Alerts are raised, tickets are logged, and engineers correlate information across tools to identify what has gone wrong and how to fix it. Service desks spend a large share of effort on repetitive, well-understood incidents such as password resets, disk space issues, service restarts and connectivity problems. Even where automation exists, it is usually triggered manually or scripted for narrow cases, and it is rarely connected to a broader understanding of the environment.

AIOps, the application of machine learning and analytics to IT operations data, has helped address part of this problem. Event correlation, noise reduction and anomaly detection are now mature capabilities in many enterprises. However, in practice most AIOps deployments stop at insight. They tell engineers that something is wrong and, in better cases, where to look, but diagnosis and remediation still depend on human effort. The recent progress in large language models (LLMs) and agentic AI, where AI systems can plan, reason and act through tools, opens up the possibility of closing this gap.

This paper proposes a vendor-neutral reference architecture that takes IT operations from AIOps-assisted monitoring towards autonomous, self-healing operations using agentic AI. The paper makes three contributions: (i) a layered reference architecture built around specialised operational agents; (ii) a four-stage maturity model for moving from reactive to autonomous operations; and (iii) a governance framework that allows autonomy to be introduced safely in production environments. The perspective is that of an enterprise architect working on large infrastructure managed services and transformation programmes, and the focus is on practical adoption rather than on any specific product.

2. Literature Survey

The idea of self-managing IT systems is not new. Kephart and Chess introduced the vision of autonomic computing, describing systems that are self-configuring, self-healing, self-optimising and self-protecting, built around a monitor, analyse, plan and execute loop driven by shared knowledge [1]. While the vision was widely accepted, its realisation was limited by the rule-based techniques available at the time and by the heterogeneity of enterprise environments.

The term AIOps gained traction as machine learning began to be applied to operational data at scale. Dang, Lin and Huang described the practical challenges of AIOps in large cloud environments, including data quality, the cost of labelled incidents and the difficulty of moving models into day-to-day operations [2]. Notaro, Cardoso and Gerndt surveyed AIOps methods for failure management and organised the field into failure prevention, prediction, detection, root cause analysis and remediation, noting that remediation remained the least developed area [3]. Chen et al. similarly argued for intelligent incident management across the full incident lifecycle rather than detection alone [4].

Site Reliability Engineering practices popularised the notion of reducing toil through automation, error budgets and blameless post-incident learning [5], while ITIL 4 provides the service management context, including incident, problem and change management, in which operational automation must operate [6]. These frameworks remain relevant because autonomous operations still need to respect service management processes and controls.

More recently, LLMs have shown promise in operational tasks. Ahmed et al. evaluated LLMs for recommending root causes and mitigation steps for cloud incidents and found that they can produce useful recommendations, especially when grounded in historical incident data [7]. Retrieval-augmented generation (RAG) offers a way to ground model output in enterprise knowledge such as runbooks and past incidents [8]. In parallel, the ReAct pattern showed that interleaving reasoning with tool use improves the reliability of LLM-based agents [9], and surveys of LLM-based autonomous agents describe architectures for planning, memory and action [10]. From a governance perspective, the NIST AI Risk Management Framework sets out principles for managing the risks of AI systems, including accountability, transparency and human oversight [11].

Taken together, the literature provides strong building blocks: detection and correlation from AIOps, reasoning and knowledge grounding from LLMs, tool-using agents, and governance principles. What is less well covered is how these pieces fit together into an enterprise architecture that can act safely on production infrastructure. This paper addresses that gap.

3. Problem Definition

The core problem addressed in this paper can be stated as follows: how can an enterprise architect IT operations so that AI systems not only detect and explain issues but also resolve them, safely and with appropriate control, across a heterogeneous infrastructure estate? Five specific challenges shape this problem:

- **Fragmented data:** monitoring, logs, tickets, configuration data and knowledge articles sit in separate tools with inconsistent identifiers, which makes correlation and context difficult.
- **Insight without action:** AIOps outputs are usually advisory; remediation still requires an engineer to interpret, decide and execute.
- **Narrow automation:** existing scripts and runbooks automate individual tasks but are not connected to diagnosis or to the wider state of the environment.
- **Trust and risk:** allowing AI to change production systems raises concerns around incorrect actions, cascading failures, compliance and accountability.
- **Operating model fit:** autonomy must work within established incident, problem and change management processes and contractual service levels, particularly in managed services arrangements.

4. Methodology/ Approach

The proposed approach combines a layered reference architecture, a set of specialised operational agents, a maturity model and a governance framework. The architecture was derived from common patterns observed in enterprise infrastructure operations and from the building blocks identified in the literature survey.

4.1 Layered Reference Architecture

The reference architecture is organised into five layers, shown in Figure 1. Each layer has a distinct responsibility, so that capabilities can be introduced and replaced independently.

Layer	Key components
L5 Experience and Governance	Natural-language interface for engineers, approval workflows, audit trail, explainability, policy engine
L4 Agentic Orchestration	Detection, Correlation, Diagnosis, Remediation Planning, Execution and Validation agents; LLM reasoning; orchestrator
L3 Automation and Execution	Runbook automation, configuration management, ITSM workflows, cloud and platform APIs, rollback
L2 Operational Data and Knowledge	Unified operational data layer, service topology, CMDB, historical incidents, knowledge base and runbooks
L1 Infrastructure and Telemetry	Data centre, cloud, network, end-user devices; metrics, logs, traces, events and tickets

Figure 1: Layered reference architecture for self-healing IT operations

Layer 1 covers the infrastructure estate and the telemetry it produces. Layer 2 is the foundation of the architecture: a unified operational data layer that brings together metrics, logs, events, tickets, configuration and topology data, and links them through a common service model. Historical incidents, known errors, knowledge articles and runbooks are indexed so that agents can retrieve relevant context. Without this layer, agents would reason on incomplete information, which is the most common reason AI-driven operations fail in practice.

Layer 3 provides governed execution. All changes to the environment are made through existing automation tools, ITSM workflows and platform APIs rather than through direct, unrestricted access by a model. Each automated action is defined with its preconditions, expected outcome and rollback step. Layer 4 contains the agents and the orchestrator, described in Section 4.2. Layer 5 provides the interface through which engineers interact with the system, review and approve actions, and inspect the reasoning behind decisions.

4.2 Specialised operational agents

Rather than relying on a single general-purpose agent, the architecture uses a set of specialised agents coordinated by an orchestrator. This keeps each agent's scope narrow, makes behaviour easier to test and allows different techniques to be used where they work best. Table 1 summarises the agents.

Table 1: Operational agents and their responsibilities

Agent	Responsibility	Primary techniques
Detection	Identify anomalies and degradations before or as they affect users	Statistical baselines, ML anomaly detection
Correlation	Group related events into a single probable incident	Topology-aware correlation, clustering
Diagnosis	Identify the most likely root cause with supporting evidence	LLM reasoning, RAG over past incidents, dependency analysis
Remediation Planning	Select or compose a remediation plan and assess its risk	Runbook retrieval, policy rules, LLM planning
Execution	Carry out approved actions through governed automation	Automation tools, ITSM and platform APIs
Validation	Confirm recovery, trigger rollback if needed, update records	Health checks, service KPIs, ticket updates

In a typical flow, the Detection Agent raises a signal, the Correlation Agent links it to related events and affected services, and the Diagnosis Agent forms a root cause hypothesis using topology, recent changes and similar historical incidents. The Remediation Planning Agent then selects a plan and classifies its risk. Depending on that classification, the plan is either executed automatically or routed for human approval. The Validation Agent confirms that the service has recovered, updates the incident record and feeds the outcome back into the knowledge layer so that future decisions improve.

LLMs are used mainly for reasoning, summarisation, runbook interpretation and natural-language interaction with engineers. They are not given open-ended control over infrastructure. Every action they propose must map to a pre-approved, parameterised automation, which limits the impact of incorrect model output.

4.3 Maturity model

Enterprises rarely move to autonomous operations in a single step. The proposed maturity model, shown in Table 2, defines four stages that can be adopted progressively, service by service.

Table 2: Maturity model for autonomous IT operations

Stage	Characteristics	Role of people
1. Reactive	Alert-driven monitoring, manual triage and resolution	Engineers detect, diagnose and fix
2. Predictive	AIOps correlation, noise reduction and early warning	Engineers act on AI insights
3. Preventive	Automated diagnosis and recommended or approved remediation	Engineers approve and supervise
4. Autonomous	Closed-loop resolution of known issues within policy limits	Engineers set policy and handle exceptions

4.4 Governance framework

Autonomy is only useful if it can be trusted. The governance framework defines the controls that allow autonomy to grow in a measured way:

- **Risk-based approval thresholds:** each action is classified by impact and confidence; low-risk, high-confidence actions run automatically, while others require approval.
- **Human-in-the-loop checkpoints:** engineers review proposed actions for critical services and can override decisions at any point.
- **Change management integration:** automated actions are logged as standard or pre-approved changes and respect change freezes and maintenance windows.
- **Rollback and blast-radius control:** every action has a defined rollback, and actions are limited in scope to avoid cascading failures.
- **Audit and explainability:** the evidence, reasoning and outcome of each decision are recorded and can be reviewed after the fact.

5. Results & Discussion

This paper presents an architectural framework rather than a controlled experiment, so the discussion is based on applying the architecture to typical scenarios found in enterprise infrastructure operations. Table 3 illustrates how the approach applies across the main infrastructure towers.

Table 3: Illustrative application across infrastructure towers

Tower	Example scenario	Autonomous action
Service Desk	Repeated password lockouts and access requests	Verify identity, reset or unlock, update ticket
End User Computing	Device performance degradation or failed patch	Clear cache, repair agent, redeploy patch
Data Centre	Disk or file system nearing capacity	Archive logs, extend volume within policy
Network	Link flapping or interface errors	Isolate path, fail over, raise vendor case
NOC	Service health check failure after a change	Correlate with change, roll back, notify owner

Compared with a conventional AIOps deployment, the main difference is that the loop is closed. Diagnosis and remediation are no longer separate manual steps, and the knowledge gained from each incident is captured systematically. The expected benefits include lower mean time to resolve for known issues, fewer tickets reaching engineers, more consistent resolution quality and better use of skilled staff on complex problems. In managed services contexts, these benefits translate into improved service levels and a more sustainable cost model. [Author note: insert measured results here, for example reduction in MTTR or percentage of incidents auto-resolved, from an anonymised engagement or pilot, if available.]

The discussion also highlights limitations. The quality of outcomes depends heavily on the operational data layer; inaccurate configuration data or poor topology mapping will directly reduce diagnostic accuracy. LLM reasoning can be convincing even when wrong, which is why the architecture constrains models to pre-approved actions and requires validation after every change. Finally, organisational

readiness matters as much as technology: teams need to trust the system, and that trust is best built by starting with low-risk, high-volume scenarios and expanding scope as results are proven.

6. Conclusion

This paper has proposed a vendor-neutral reference architecture for moving enterprise IT operations from AIOps-assisted monitoring to autonomous, self-healing operations using agentic AI. The architecture brings together a unified operational data layer, specialised agents for detection, correlation, diagnosis, planning, execution and validation, and a governed execution layer. A four-stage maturity model and a governance framework provide a practical path for adopting autonomy safely. The central argument is that self-healing infrastructure is achievable today, but only when AI reasoning is combined with strong operational data, constrained automation and clear human oversight.

7. Future Scope

Several areas merit further work. First, empirical evaluation of the architecture in production environments, measuring resolution times, auto-resolution rates and error rates, would strengthen the findings. Second, multi-agent coordination across organisational boundaries, such as between enterprises, cloud providers and managed service providers, raises open questions about trust and accountability. Third, techniques for quantifying agent confidence and calibrating approval thresholds automatically would allow autonomy to scale more safely. Finally, extending the architecture to proactive capacity planning and sustainability goals, such as energy-aware workload placement, is a promising direction.

References

[1] Kephart JO, Chess DM. The vision of autonomic computing. Computer. 2003;36(1):41-50.
[2] Dang Y, Lin Q, Huang P. AIOps: real-world challenges and research innovations. In: Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE; 2019. p. 4-5.
[3] Notaro P, Cardoso J, Gerndt M. A survey of AIOps methods for failure management. ACM Transactions on Intelligent Systems and Technology. 2021;12(6):1-45.
[4] Chen Z, Kang Y, Li L, Zhang X, Zhang H, Xu H, et al. Towards intelligent incident management: why we need it and how we make it. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM; 2020. p. 1487-97.
[5] Beyer B, Jones C, Petoff J, Murphy NR, editors. Site reliability engineering: how Google runs production systems. Sebastopol (CA): O'Reilly Media; 2016.
[6] AXELOS. ITIL foundation: ITIL 4 edition. London: TSO; 2019.
[7] Ahmed T, Ghosh S, Bansal C, Zimmermann T, Zhang X, Rajmohan S. Recommending root-cause and mitigation steps for cloud incidents using large language models. In: Proceedings of the 45th International Conference on Software Engineering (ICSE). IEEE; 2023. p. 1737-49.
[8] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems. 2020; 33: 9459-74.
[9] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K, et al. ReAct: synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR); 2023.
[10] Wang L, Ma C, Feng X, Zhang Z, Yang H, Zhang J, et al. A survey on large language model based autonomous agents. Frontiers of Computer Science. 2024;18(6):186345.
[11] National Institute of Standards and Technology. Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1. Gaithersburg (MD): NIST; 2023.

Author Profile

Mihir Patel leads Cloud Presales and Solutions at Tata Consultancy Services (TCS). He has over 28 years of experience in the IT industry, more than 25 of them with TCS, where he has led many strategic customer engagements across the globe and shaped key solutions for client needs. His areas of specialisation are enterprise architecture and cloud technologies.