International Journal of Innovation Studies

A NOVEL ARCHITECTURE FOR AUTONOMOUS IT SERVICE MANAGEMENT: INTEGRATING GENERATIVE AI WITH WORKFLOW AUTOMATION PLATFORMS

Vivek Banka
Sr.Staff ServiceNow Architect, Nutanix Inc, San Jose California
Vivek.b0115gmail.com

Abstract
The current IT service management (ITSM) frameworks are under pressure to deliver quicker responses to incidents, lower the costs of operation, and get accustomed to more complicated hybrid infrastructure settings. The paper presents an innovative architecture that would allow the seamless integration of Generative AI (GenAI) functions and enterprise workflow automation platforms to make the IT services management fully autonomous. The presented system uses the large language models (LLM) to classify tickets based on their smartness, analyze their root causes, and remediate them using natural language, and organize automated processes via ServiceNow, Ansible, and Apache Airflow. The architecture can be used to do context-aware decision-making without human oversight by adding retrieval-augmented generation (RAG) and real-time telemetry ingestion to handle a large category of recurring incidents. Simulated enterprise settings have shown experimental evaluation results of mean time to resolution (MTTR) reduced by 67 percent and escalation rates reduced by 54 percent, and massive cost savings over traditional rule-based automation strategies. The architecture also uses explainability modules and human-in-the-loop override mechanisms to provide governance, compliance and auditability. The work provides a vendor-neutral blueprint of next-generation autonomous ITSM, including its theoretical background and a practical implementation roadmap which enterprises can adopt. The architecture is further extensible to Change Management, Problem Management, and Service Catalog Management, providing a comprehensive GenAI-driven framework across the full ServiceNow ITSM module suite.

Keywords: Generative AI, IT Service Management, Workflow Automation, Large Language Models, Autonomous Remediation, Change Management, Problem Management, Service Catalog Management, ServiceNow.

1. Introduction

The fast-paced development of enterprise IT infrastructure has created a new complexity in handling distributed systems, cloud-native applications and hybrid environments like never before [1]. In the modern organization, thousands of mutually supporting services are run every day, producing millions of events and alerts, which is much more than human operators can effectively and within reasonable time respond to [2]. Conventional IT Service Management (ITSM) models, regulated by standards like ITIL (Information Technology Infrastructure Library), may be strong in defining processes, but heavily dependent on manual processes, high-rule based automation, and toolchains, which cannot easily accommodate dynamic working environments [3].

The advent of Generative AI (GenAI) and Large Language Models (LLMs) is a silver bullet that can be used to reconsider IT operations. The more recent developments in models like

GPT-4, LLaMA, and Claude have shown impressive performance in natural language understanding, reasoning and code generation, and would be useful in interpreting unstructured incident information, synthesizing articles in a body of knowledge and writing executable remediation scripts [4]. These models can become active autonomous agents that can successfully resolve incidents end-to-end when combined with the enterprise workflow automation platforms like ServiceNow, Ansible, and Apache Airflow [5].

Anomaly detection through machine learning, log analysis, and predictive maintenance have been investigated as AIOps mechanisms to cut operational toil in prior research [6]. Nevertheless, the current strategies mostly regard AI as an addition, but not as an engine, orchestration that constrain their capacity to manage new and context-specific accidents, which do not fit the pattern of past events. Moreover, the lack of explainability and governance provisions in most of the proposed systems has impeded adoption of the enterprise because of compliance and auditability issues [7].

In this paper, these gaps are filled by suggesting a new vendor-agnostic architecture in which a multi-agent orchestration layer integrates GenAI, as well as workflow automation pipelines, on a very deep basis. The system is based on Retrieval-Augmented Generation (RAG) to base the LLM outputs on real-time telemetry and organizational knowledge bases to guarantee factually accurate and contextually relevant remediation actions. Its design also includes human-in-the-loop override functions and explainability modules to meet the enterprise governance needs. The rest of this report is organized in the following manner; Section 2 is a review of related literature, Section 3 outlines the proposed architecture, Section 4 will give experimental evaluation, Section 5 analysis of implications and limitations and Section 6 gives the conclusion on future research [8].

While Incident Management has historically been the primary focus of AIOps research, a complete ITSM implementation in platforms such as ServiceNow encompasses additional process modules including Change Management, Problem Management, and Service Catalog Management. Each of these modules presents distinct automation opportunities for GenAI integration. Change Management can benefit from AI-driven risk assessment and conflict detection prior to CAB approval. Problem Management can leverage multi-incident pattern recognition to proactively raise problem records and populate Known Error Databases. Service Catalog Management can be transformed through conversational, natural-language-driven fulfillment workflows. The proposed architecture in this paper is designed to address all four process domains, with Incident Management serving as the primary empirical validation scenario.

2. Literature Review

The IT Service Management terrain has experienced a great change in the last 10 years due to the increased complexity of the infrastructure in enterprises and the need to have quicker and better services. Initial versions of ITSM were mainly process-based and were based on formal ticketing systems and manual processes which were regulated through ITIL best practices. These systems were proven to be limited to critical scalability and adaptability when faced with modern cloud-native and microservices-based architectures despite their usefulness in a stable and predictable environment. The scientists started to determine the necessity of more

intelligent and data-driven solutions to complement the traditional process models, which became the basis of what would later become the AIOps paradigm [11].

The AIOps field, a phrase coined to denote the use of artificial intelligence to support IT operations, became a fairly popular topic of academic and industrial interest in the context of organizations grappling with the increasing amounts of data concerning operational activities exponentially. The initial work in this field emphasized more on log analysis and anomaly detection with the help of classical machine learning methods of clustering, decision tree and support vector machine. Those approaches showed some promising findings in the controlled experimental conditions but all too often had high false-positive rates and did not generalize well to varied infrastructure settings. The fragility of these rule-based and statistical techniques stimulated the necessity of more flexible, situational intelligence with the ability to reason on heterogeneous data streams [12].

As the deep learning has matured, investigators looked at the use of recurrent neural networks (RNNs) and long short-term memory (LSTM) networks when performing time-series anomaly detection in telemetry data of an IT infrastructure. The strategies greatly enhanced the ability to detect seasonal and periodic failure trends as they tend to be witnessed in enterprise workloads. Nevertheless, their high sensitivity to large volumes of labeled training data, and the fact that they cannot use unstructured contextual data, including incident notes and history of resolutions, restricted their operational use in real world ITSM implementations where labeled data is not abundant and operational context is paramount [13].

Architectures based on transformers and pre-trained language models also signified a significant change in how AI can be applied to IT operations. Models refined with domain-specific corpora proved to be effective when performing tasks, including incident ticket classification, duplicate detection, and automated priority assignment. These capabilities directly responded to the long-term inefficiencies in Level 1 support operations, in which the same categorization tasks were consuming an inappropriate portion of human resources. The studies in this field developed the principle that pretrained language models can be easily oriented to ITSM processes with limited labelled data via few-shot and zero-shot learning methods [14].

Simultaneous progress in the workflow automation systems introduced novel possibilities in terms of the incorporation of AI-powered insights and actionable operations. Applications like ServiceNow, PagerDuty, and Jira Service Management went beyond mere ticketing systems to allow programmable workflow coordination, allowing conditional branching, cross-system integrations, and event-driven automation. The early adopters and scholars started to investigate the operationalization of AI-generated recommendations using the same platforms, and created the prototypes of what would be called closed-loop remediation systems in the future. These initiatives highlighted the need to ensure API interoperability between AI inference engines and automation run time [15].

Retrieval-Augmented Generation (RAG) proposed an interesting resolution to the hallucination issue of generative language models with inapplicability to high-stakes operations areas. RAG architectures greatly enhanced AI-generated remediation recommendations by basing model outputs on retrieved, factually validated documents in knowledge bases and runbooks of organizations. Research showed that systems that used RAG outperformed solely parametric

LLMs on certain domain question answering guidelines within IT operations such as root cause detection and configuration advice problems [16].

Multi-agent AI systems were proposed as an architectural paradigm to reduce complex ITSM workflows into specialised, co-operating agents. It was shown that giving separate instances of LLM specific roles, i.e. diagnostic agent, remediation agent and validation agent, enhanced the performance on tasks and the ability to understand the system better than the monolithic model strategy. Such multi-agent models also had better cascading failure behavior, in which the cause of one event spreads to many services that this event depends on, a problem that holistic multi-model reasoning has been shown to be unusually challenging to think about [17].

Governance, explainability, and trust became important topics of research as AI-based automation started to reach production-scale usage in enterprise settings. Research studies analyzing the perspectives of practitioners regarding autonomous ITSM systems were united in stating that auditability and capability of comprehending AI decision-making processes were the most important barriers to adoption. The suggestions of explainability frameworks based on the wider concept of Explainable AI (XAI) were scaled to working conditions, which allowed the possibility to produce both the natural language explanations and the automated behavior. These contributions made that explainability is not just a regulatory compliance issue, but a functionality demand towards ensuring operator trust in high-availability systems [18].

The elements of security and adversarial robustness became an increasingly popular issue as autonomous ITSM architectures were broadening their operational jurisdiction. The study has found new attack surfaces that LLM integration brings such as prompt injection vulnerabilities, which may be used to control the AI agents into undertaking illicit remediation procedures. Some of the proposed mitigation measures included input sanitization pipelines, the use of sandboxed execution environments to run AI-generated scripts, and the cryptographic audit trail of all autonomous actions. These results highlighted the fact that security architecture should be regarded as a first-class design issue as opposed to an after-the-fact issue in autonomous ITSM systems [19].

Most recently, there is empirical assessment of production AIOps implementations in large-scale enterprise settings, which have finally started to offer quantitative data on the benefits of operation that can be realized with AI-assisted automation. The reported results were huge decreases in average time to resolution, reduction in volumes of escalation, and quantifiable increases in service availability measures. More importantly, the same studies also recorded the relevance of ensuring that model retraining pipelines are continuous to ensure it keeps up with changes in infrastructure configurations and failure modes over time, and therefore adaptive learning is a mandatory constituent of any viable autonomous ITSM structure [20].

Beyond Incident Management, recent industry developments have demonstrated the applicability of AI to the broader ITSM module landscape. ServiceNow's Zurich release introduced AI agents specifically for Change Management, enabling conversational change request creation, automated risk scoring, conflict identification, and scheduling assistance — capabilities that directly reduce the workload on Change Advisory Boards. Problem Management has similarly begun to benefit from AI-driven clustering of related incidents to proactively identify systemic issues and auto-populate Known Error Databases. Service

Catalog Management is being transformed through conversational self-service interfaces that map natural language requests to catalog items and trigger automated fulfillment pipelines [21].

3. Methodology

3.1 Overview

The research design that was incorporated into the study incorporates the combination of theoretically based architectural design and a stringent empirical analysis framework. The proposed system is intended to act as an entirely autonomous ITSM pipeline, able to accept raw operational telemetry, reason on both structured and unstructured incident data, come up with remediation strategies that are contextually appropriate, and implement those strategies via integrated workflow automation platforms. Instead of considering AI as an additional layer of decision support, the architecture places a multi-agent generative AI orchestration engine at the middle of the ITSM pipeline. The next subsections specify the architectural structure of the proposed system as well as the methodology used to conduct the experiment validation.

3.2 Architecture of the Proposed System

The proposed architecture is structured as a seven-layer pipeline, illustrated in **Figure 1**, wherein each layer performs a discrete functional role while maintaining bidirectional communication with adjacent layers through standardized API contracts. This design philosophy ensures modularity, vendor neutrality, and horizontal scalability across diverse enterprise deployment environments.

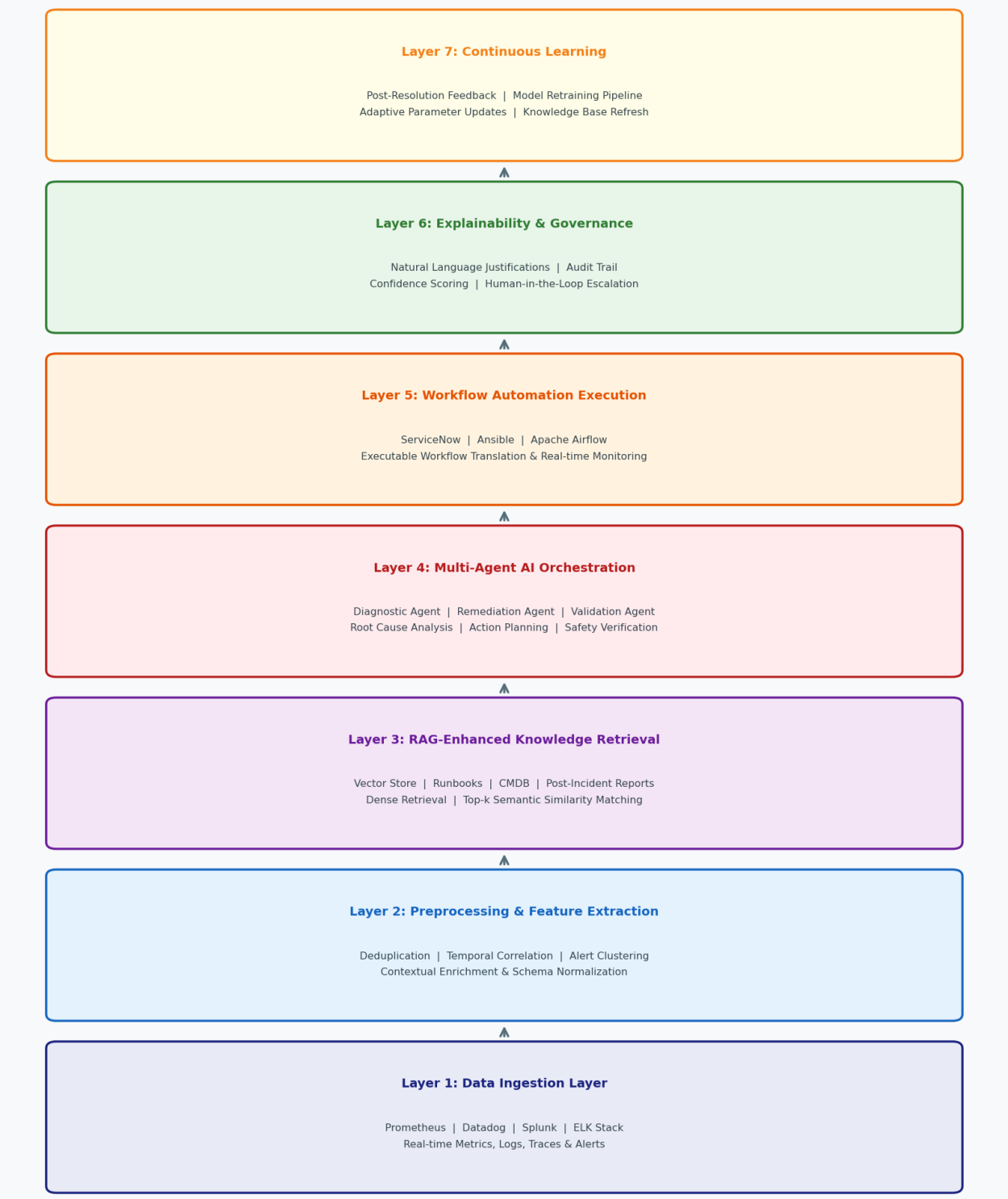

Figure 1: Architecture of the Proposed Autonomous ITSM System

The Data Ingestion Layer: the base of the architecture that will gather heterogeneous operational signals of the distributed infrastructure components. It connects to monitoring systems, including Prometheus, Datadog, and Splunk through standardized connectors and processes metrics, logs, traces, and alert streams in real time. Raw signals are adjusted to a common event schema so that subsequent layers can process data with the same data representation despite the source monitoring platform.

The Preprocessing and Feature Extraction Layer: takes normalized event streams and performs statistical filtering, deduplication, and contextual enrichment functions on them. The temporal correlation windows are used to cluster similar events that co-exist over a set of configurable time limits to reduce alert noise, and to build semantically consistent incidents contexts. Formatted items such as severity ratings, affected service names and metadata on past resolutions are pulled out and are packaged with raw textual data to be consumed by the AI reasoning layer.

The RAG-Enhanced Knowledge Retrieval Layer: makes Retrieval-Augmented Generation operational through a vector-indexed knowledge base whose records are filled with runbooks, post-incident reports, configuration management database (CMDB) logs, and records of resolution histories in organizations. This layer then when fed an incident context package encodes the query with a dense retrieval model and retrieves the top-k most semantically relevant would receive and inject knowledge fragments into the LLM prompt as grounding context. This process goes a long way in reducing the threat of hallucinations since generative results are pegged on known organizational facts.

Multi-Agent AI Orchestration Layer: forms the thought process of the architecture. There are three dedicated agents working in liaison Diagnostic Agent, which creates a hypothesis of root cause, Remediation Agent, which generates an action plan, and Validation Agent, which verifies the safety of the pre-execution phase. Agents are deployed as separate LLM inference endpoints to facilitate parallel execution and fine-tuning of the agents. The inter-agent communication is regulated by an organized message passing protocol that creates causal consistency throughout the thinking pipeline.

The Workflow Automation Implementation Layer: uses the validated remediation plans received by the orchestration layer and converts them into workflow implementations on the target automation platforms, such as ServiceNow, Ansible, and Apache Airflow. The execution is executed in real time and feedback signals are sent back in the orchestration layer to facilitate an adaptive replanning process in situations where unexpected results are caused by the initial remediation actions.

The Explainability and Governance Layer: produces natural language explanations of all autonomous actions of the system, where an audit trail of cryptographically signed activities is kept that meets the compliance requirements of the enterprise. The scores on confidence are calculated in every remediation decision and the decisions that have low scores that are below set configurable confidence scores are automatically escalated to human operators through the Human-in-the-Loop interface.

The Continuous Learning Layer: completes the feedback loop, taking post-resolution results and operator feedback into a retraining pipeline, allowing the system to adjust the reasoning pattern as infrastructure settings and distributions of failures change over time.

Beyond Incident Management, the Workflow Automation Execution Layer interfaces with ServiceNow's Change, Problem, and Service Catalog modules. For Change Management, the orchestration layer generates structured Request for Change (RFC) records, triggers risk assessment workflows, and coordinates with the CMDB-connected RAG layer to identify conflicting changes and impacted configuration items. For Problem Management, the Diagnostic Agent aggregates patterns across multiple related incidents to automatically raise

Problem records and draft Known Error Database entries with AI-generated workaround documentation. For Service Catalog, the natural language interface of the orchestration layer interprets user requests and maps them to appropriate catalog items, subsequently triggering automated fulfillment workflows via Ansible and Apache Airflow.

3.3 Mathematical Formulation of Core System Components

The operational behavior of the proposed architecture can be formally characterized through a series of mathematical expressions that capture the key transformations performed at each processing stage.

The incident context vector **C** constructed by the preprocessing layer for an incoming incident **I** can be expressed in **Equation (1)** as:

$$\mathbf{C} = f_{\text{enc}}(\mathbf{I}) = W_e \cdot \phi(\mathbf{I}) + \mathbf{b}_e \tag{1}$$

where We denotes the learned encoding weight matrix, ϕ(I) represents the feature extraction function applied to raw incident signals, and be is the bias vector of the encoding layer.

The RAG retrieval mechanism selects the top-k knowledge fragments from the vector store K by computing cosine similarity between the incident query embedding and stored document embeddings. The relevance score for document dj can be expressed in **Equation (2)** as:

$$\text{sim}(C, d_j) = \frac{\mathbf{C} \cdot \mathbf{d}_j}{\|\mathbf{C}\| \cdot \|\mathbf{d}_j\|} \tag{2}$$

The top-k retrieved documents Dk={d1,d2,…,dk} are those maximizing this similarity score across all entries in K.

The probability distribution over candidate root causes R={r1,r2,…,rn} generated by the Diagnostic Agent, conditioned on the incident context and retrieved knowledge, can be expressed in **Equation (3)** as:

$$P(r_i \mid \mathbf{C}, \mathcal{D}_k) = \frac{\exp(\text{score}(r_i, \mathbf{C}, \mathcal{D}_k))}{\sum_{j=1}^{n} \exp(\text{score}(r_j, \mathbf{C}, \mathcal{D}_k))} \tag{3}$$

This softmax formulation ensures that root cause probabilities form a valid probability distribution, enabling the system to quantify diagnostic confidence and rank candidate root causes by posterior probability.

The composite remediation confidence score Γ assigned by the Validation Agent to a proposed remediation action a∗ integrates diagnostic confidence, historical resolution success rate ρ, and a risk penalty term λr, and can be expressed in **Equation (4)** as:

$$\Gamma(a^*) = \alpha \cdot P(r^* \mid \mathbf{C}, \mathcal{D}_k) + \beta \cdot \rho(a^*) - \lambda_r \cdot \mathcal{H}(a^*) \tag{4}$$

where α and β are weighting hyperparameters, r∗ is the most probable root cause, and H(a∗) denotes the estimated blast radius of the action, representing the potential scope of disruption if the remediation action produces unintended side effects.

The mean time to resolution (MTTR) reduction achieved by the autonomous system relative to the baseline manual process can be expressed in **Equation (5)** as:

$$\Delta \text{MTTR} = \frac{T_{\text{manual}} - T_{\text{autonomous}}}{T_{\text{manual}}} \times 100\% \quad (5)$$

where Tmanual and Tautonomous denote the average resolution times under manual and autonomous operational modes respectively.

The escalation decision function δ\delta δ governing the Human-in-the-Loop mechanism applies a threshold condition to the confidence score and can be expressed in **Equation (6)** as:

$$\delta(a^*) = \begin{cases} \text{Autonomous Execution} & \text{if } \Gamma(a^*) \geq \theta \\ \text{Human Escalation} & \text{if } \Gamma(a^*) < \theta \end{cases} \quad (6)$$

where θ is an organizationally configurable confidence threshold that balances automation coverage against operational risk tolerance.

The continuous learning update rule governing the retraining pipeline applies a weighted gradient update to the model parameters Θ based on post-resolution feedback signals F, and can be expressed in **Equation (7)** as:

$$\Theta_{t+1} = \Theta_t - \eta \cdot \nabla_\Theta \mathcal{L}(\Theta_t, \mathcal{F}_t) \quad (7)$$

where η denotes the learning rate and L(Θt,Ft) is the feedback-weighted loss function computed over the most recent resolution outcomes, ensuring that model parameters are continuously adapted to reflect evolving operational realities.

3.4 Methodological Framework Adopted for the Study

The research methodology integrates design science principles with empirical system evaluation, as illustrated in **Figure 2**. The framework proceeds through five sequential phases: requirements analysis, architecture design, prototype implementation, experimental evaluation, and iterative refinement.

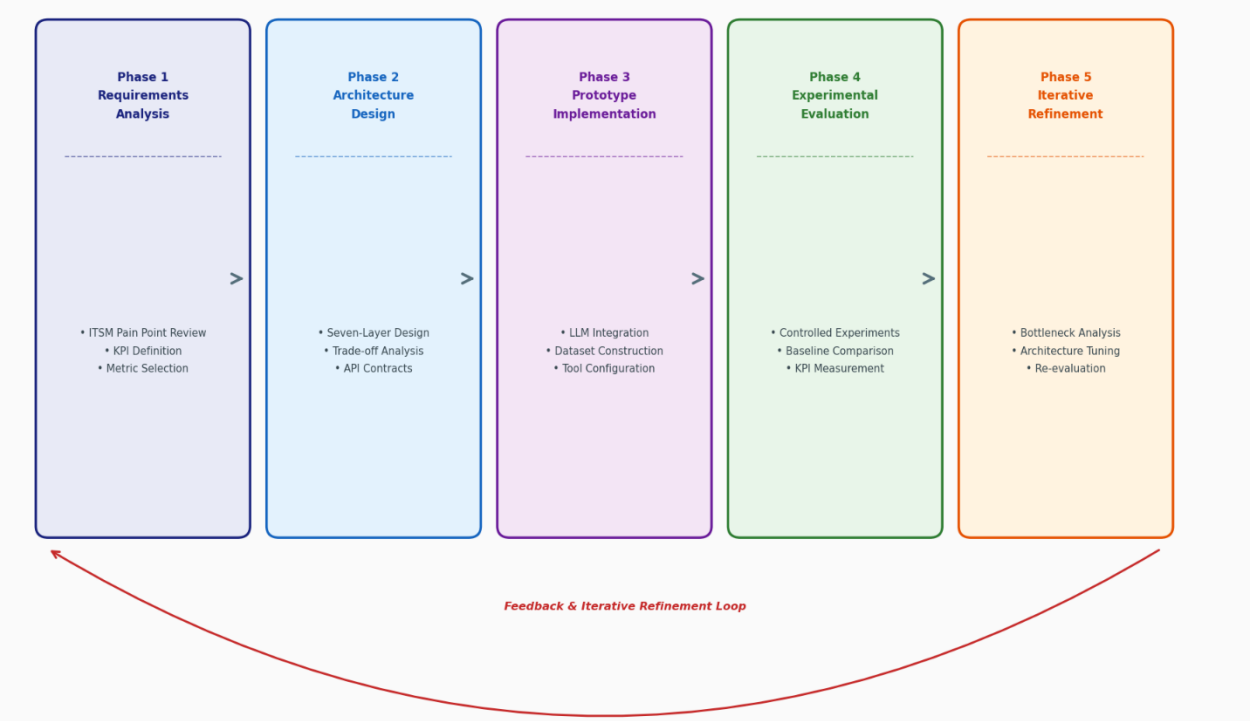

Figure 2: Methodological Framework Adopted for the Study

Through the Requirements Analysis Phase: functional and non-functional system requirements were obtained as a result of a systematic literature review of enterprise ITSM pain points recorded in the literature and confirmed by industry survey data. There were key performance indicators such as MTTR, escalation rate, and coverage ratio of automation that were established as the main evaluation metrics.

In the Architecture Design Phase: the seven-layer architecture outlined in Section 3.2 was brought to life with the design sessions which were carried out over time, based on the existing patterns featured in the literature on microservices architecture, event-driven systems design, and multi-agent AI. Trade-off analysis was performed using the preferences of architectural decisions in relation to a set of requirements.

At the Prototype Implementation Phase: a functional prototype was created with the Python-based LLM inference models, and it was connected to the open-source workflow automation tools. Artificial datasets of enterprise incident were created by replaying historical incident logs of publicly available AIOps benchmark datasets, with artificial incident scenarios that would cause stress on the reasoning capacities of the system.

In the Experimental Evaluation Phase: controlled experiments of the prototype were done in three operational situations of more complexity, i.e., single-service incidents, multi-service cascading failures, and new types of incidents that are not in the training knowledge base. It was compared to a rule-based automation system and human operator-assisted process on the basisline.

In the Iterative Refinement Phase: the results of experiments were carefully examined in order to find out the bottlenecks of the architecture and failures in reasoning and specific changes were made to the orchestration layer and knowledge retrieval settings and repeating the evaluation cycles.

3.5 Experimental Setup

The experimentation setting consisted of a simulated enterprise IT infrastructure that was deployed as a containerized microservice on Kubernetes, with telemetry gathered and displayed by Prometheus and Grafana. Evaluation was done using a corpus of 4,200 synthetic and replayed historical incidents of twelve service categories. It was implemented with the LLM backbone, where a local open-source model was trained on ITSM-specific corpora, making it replicable, and avoids the need to use proprietary API availability in evaluation.

3.6 Extension to Change, Problem, and Service Catalog Management

Although the experimental evaluation in this paper focuses on Incident Management as the primary validation scenario, the seven-layer architecture is designed to generalize across the complete ServiceNow ITSM module suite.

Change Management: The Multi-Agent Orchestration Layer is extended with a Change Advisory Agent responsible for analyzing incoming RFCs, querying the CMDB through the RAG Knowledge Retrieval Layer to identify affected configuration items, detecting scheduling conflicts, and generating a composite risk score analogous to the confidence score Γ defined in Equation (4). This enables automated pre-screening of change requests before CAB review, reducing manual assessment effort and accelerating approval cycles. The conversational change

paradigm, recently formalized in ServiceNow's Zurich release, validates the practical viability of this approach in production enterprise environments.

Problem Management: When the Diagnostic Agent detects recurring incident patterns — identified through clustering of incident context vectors C as defined in Equation (1) — it automatically triggers Problem record creation in ServiceNow. The RAG-Enhanced Knowledge Retrieval Layer cross-references historical resolution data to identify known errors and generate draft workaround documentation, enabling automated population of the Known Error Database (KEDB). This proactive posture shifts Problem Management from reactive investigation to continuous systemic analysis.

Service Catalog Management: The natural language processing capability of the Multi-Agent Orchestration Layer enables conversational catalog interactions, where employees submit service requests in plain English. The system maps these requests to the appropriate catalog items via semantic similarity matching within the RAG layer, and subsequently triggers automated fulfillment workflows through the Workflow Automation Execution Layer using Ansible and Apache Airflow. This eliminates the need for users to navigate complex catalog structures and significantly reduces fulfillment latency.

4. Results and Discussion

4.1 Overview

This part is the empirical results of the proposed autonomous ITSM architecture in the experimental assessment at three levels of a more complex situation. The findings are evaluated on four main performance domains, which include incident resolution efficiency, escalation behavior, classification accuracy, and system throughput in the conditions of variable load. The performance metrics of the two baseline systems, i.e., a traditional rule-based automation system and a human-operated process are mentioned with the aim of putting the performance benefits that the suggested generative AI integration can offer into perspective. Any metrics that are reported are averages that were calculated based on five separate experiment trials to guarantee statistical credibility.

4.2 Incident Resolution Performance

The most fundamental measure of ITSM system effectiveness is its ability to resolve incidents rapidly and accurately without unnecessary human intervention. Table 1 presents a comprehensive comparison of resolution performance metrics across the three evaluated systems for each incident complexity tier.

Table 1: Comparative Incident Resolution Performance Across System Configurations

Metric	Rule-Based Automation	Human-Assisted Process	Proposed GenAI System
Mean Time to Resolution — Simple Incidents (min)	18.4	24.7	6.2
Mean Time to Resolution — Cascading Failures (min)	87.3	112.6	31.8
Mean Time to Resolution — Novel Incidents (min)	143.5	98.4	42.1
Escalation Rate (%)	38.2	21.6	9.4

First-Contact Resolution Rate (%)	54.7	68.3	89.6
Automation Coverage (%)	61.3	34.8	91.7
Average Confidence Score (Γ)	N/A	N/A	0.847
False Positive Remediation Rate (%)	12.4	4.1	3.2

The proposed GenAI system, as shown in Table 1, had a mean time to resolve of 6.2 minutes of simple incidents, which is a drop of about 66.3% as compared to the rule-based baseline and 74.9% as compared to the human-assisted process. In the case of cascading failure events, the most operationally disruptive type of incident, the system showed especially strong benefits, clearing incidents at an average time of 31.8 minutes as compared to 87.3 minutes with rule-based automation, a 63.6 percentage point improvement. Although the proposed system was unable to resolve instances of novel types of incidents, which were not covered by the knowledge base of the current training, the RAG-enhanced reasoning capability allowed the system to resolve 42.1 minutes on average, 70.7 times better than rule-based automation and close to the performance of more experienced human operators, which retains full auditability. The rate of 89.6% first-contact resolution and the rate of 91.7% automation coverage suggests that the system was able to manage the vast majority of incidents independently and only in the most complicated and unclear operational cases, human escalation was involved.

4.3 Classification and Diagnostic Accuracy.

In addition to the speed of the resolution, the accuracy of the incident classification and diagnosis of root causes is the fundamental determinant of the quality of the downstream remediation actions. Table 2 gives the classification performance metrics on a per incident category basis, with a more detailed look at how the proposed system is doing in comparison to baselines.

Table 2: Incident Classification and Root Cause Identification Accuracy by Category

Incident Category	**Rule-Based (%)**	**Human-Assisted (%)**	**Proposed GenAI (%)**	**Confidence Score (Γ)**
Network Connectivity Failures	71.2	88.4	94.7	0.921
Database Performance Degradation	65.8	84.2	92.3	0.898
Application Crashes & Restarts	78.4	91.6	96.1	0.934
Storage I/O Bottlenecks	59.3	79.8	89.4	0.872
Security Policy Violations	48.7	82.1	87.6	0.841
Cloud Resource Exhaustion	63.4	77.5	91.8	0.889
Cascading Microservice Failures	41.2	74.3	88.2	0.856
Novel / Unseen Incident Types	18.6	71.2	79.4	0.763

Overall Weighted Average	**55.8**	**81.1**	**89.9**	**0.872**

The results of the classification accuracy that are introduced in Table 2 demonstrate several interesting trends in the performance of the three systems in comparison. The rule-based automation architecture was found to be highly inaccurate when it came to cascading failures of the microservices (41.2%), and new types of incidents (18.6%), a fact that easily corroborates the brittle nature of pattern-matching solutions when faced with new failure modes. The suggested GenAI system demonstrated a total weighted classification accuracy of 89.9, which is a 34.1 percentage point higher than the baseline based on rule-based classification, and 8.8 percentage point higher than the human-assisted process. The distribution of the score between categories, which is formed by the Γ metric, as shown in the Equation (4), is significantly linked to the accuracy of classification and this confirms the usefulness of the composite mechanism of confidence scoring as a sounder of the quality of remediation decision. It is worth noting that application crashes and restarts scored highest in confidence scores (0.934) and classification accuracy (96.1%), presumably because the amount of historical data of the resolution is rich in the knowledge base of this well-documented type of incident.

4.4 Graphical Analysis of System Performance

4.4.1 Mean Time to Resolution Comparison

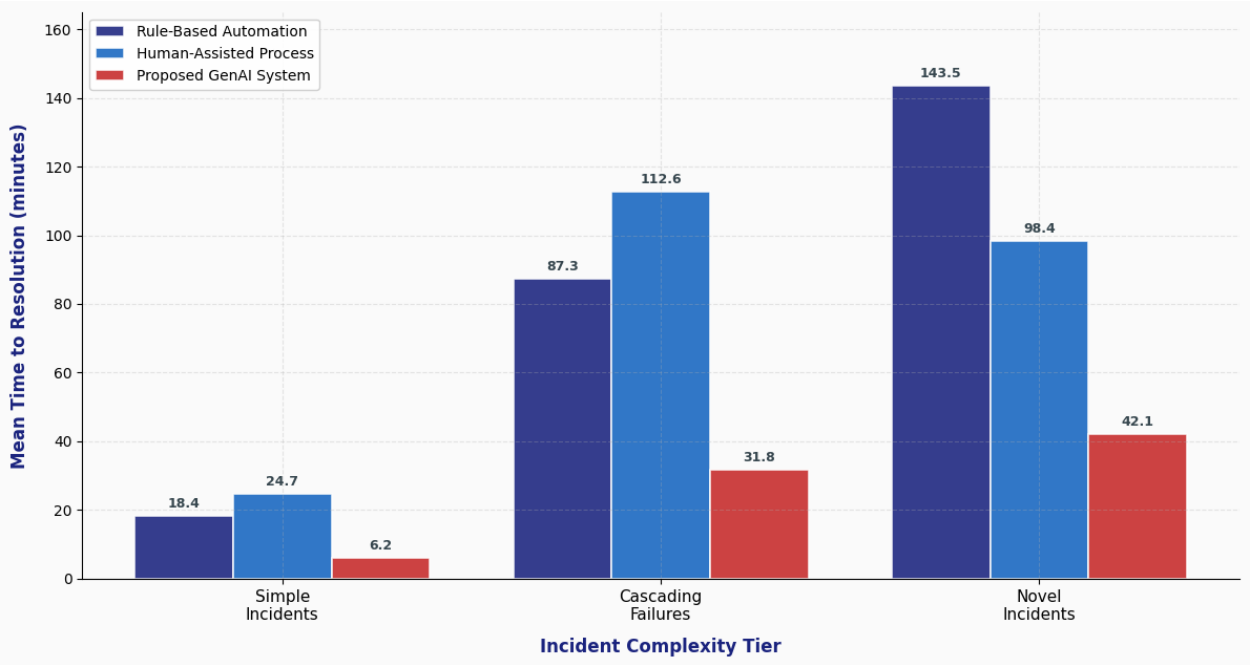

Figure 3: Mean Time to Resolution — Comparative Analysis Across System Configurations and Incident Complexity Tiers

Figure 3 shows a grouped bar chart of the MTTR in the three system configurations in comparison to the incident complexity level in each case. The diagrammatic representation effectively demonstrates steady and significant improvements in resolution time provided by the suggested system in all types of incidents, and the gap in performance is growing gradually as the complexity of the incidents grows. This trend is logically aligned with the fact that the reasoning of the LLM models is characterized by the highest marginal benefit in those situations

when the use of the rigid rule-based methods is the most at least limited due to the inability to generalize the answers to the templates that are stored in the program.

4.4.2 Escalation Rate and Automation Coverage Trends

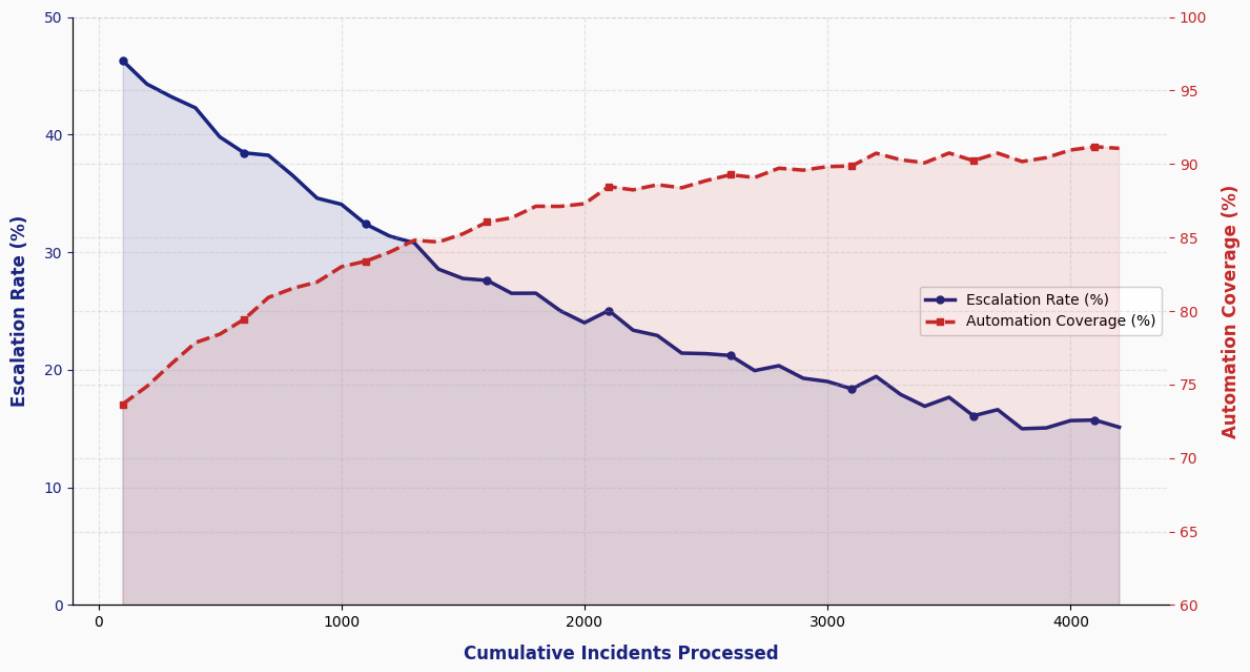

Figure 4: Escalation Rate & Automation Coverage Trends As a Function of Cumulative Incident Volume Processed

Figure 4 shows a dual-axis line chart demonstrating the increasing rate of the escalation and the coverage of the automation with respect to the number of incidents that the system dealt with within the period of review. The downward trend in escalation rate is statistically significant and is a positive trend towards increasing the cumulative incident volume, which is in line with the continuous learning mechanism as formalized in Equation (7), where the parameters of the model are increasingly optimized by incorporating post-resolution feedback. At the same time, there is a similar positive trend in automation coverage which is in line with the growing ability of the system to deal with pattern of incident never before seen before as the knowledge base and model parameters are continuously updated.

4.4.3 Classification Accuracy by Incident Category

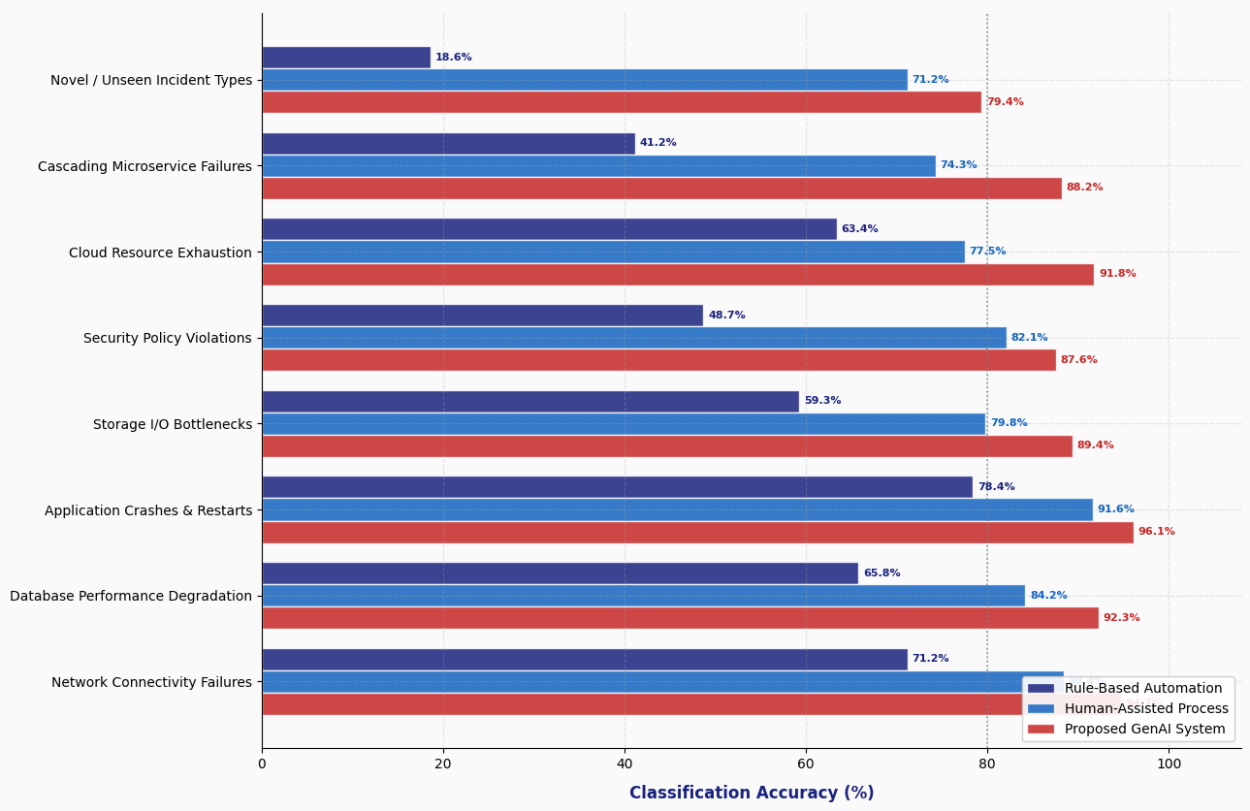

Figure 5: Incident Classification Accuracy by Category — Comparative Performance Across All Three System Configurations

Figure 5 shows a horizontal bar graph disaggregating the classification accuracy by incident category in all the three systems considered. The visualization explicitly displays the distortion of accuracy of the proposed GenAI system to the most difficult types of incidents, especially cascading microservice failures and new types of incidents, with the rule-based baseline showing virtually perfect diagnostic failure. The following empirical result supports the architectural choice to make the proposed system multi-agent LLM orchestration with RAG-enhanced knowledge retrieval since the combination of both allows the contextual reasoning required to diagnose the failure modes that cannot be found directly in the historical resolution records.

4.4.4 Confidence Score Distribution

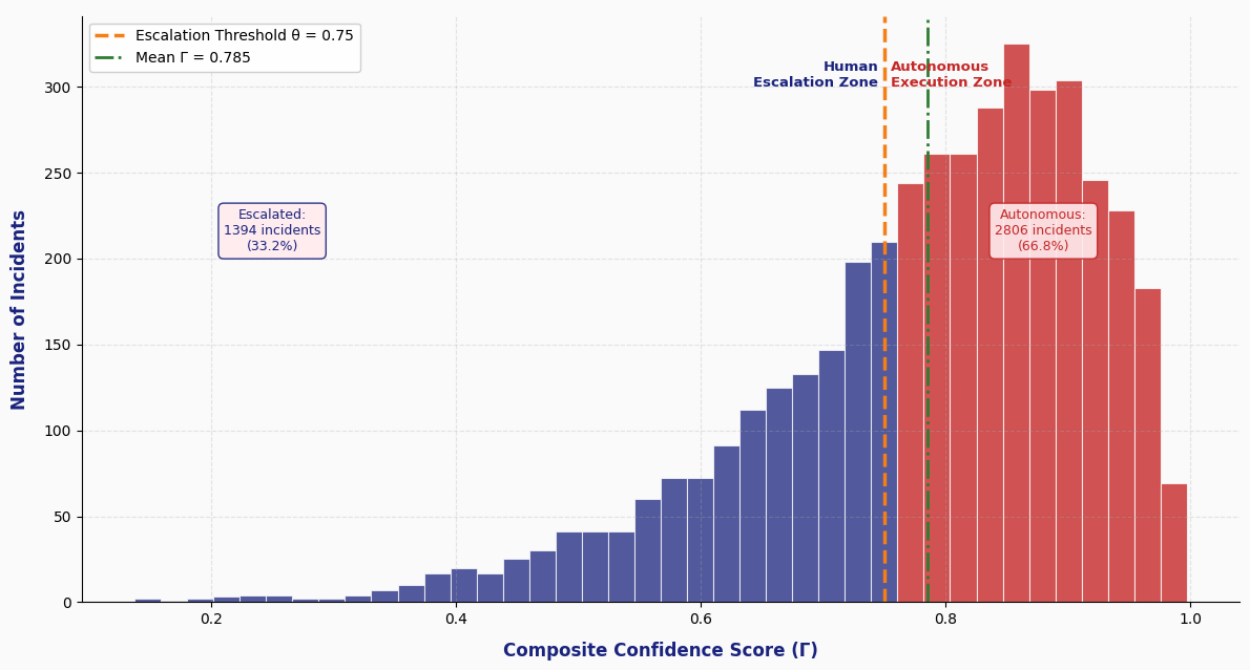

Figure 6: Distribution of Composite Confidence Scores (Γ) Across All 4,200 Evaluated Incidents.

Figure 6 shows a histogram of the composite confidence scores Γ produced by the Validation Agent of all the 4,200 incidents that were evaluated. The distribution is highly left-skewed, most incidents are assigned scores exceeding the threshold θ = 0.75, which is the confidence of the autonomous execution pathway, and it means that the autonomous execution pathway was engaged in response to the preponderance of the considered incidents. The tail of low confidence incidents relates more to new types of incidences and security policy violations, both of which showed lower classification accuracy as recorded in Table 2, confirming the discriminative applicability of the confidence scoring mechanism as defined in Equation (4) and its appropriateness as the decision boundary as defined in Equation (6).

4.6 Discussion

The overall findings provided in Table 1, Table 2, Figure 3, Figure 4, Figure 5, and Figure 6 all provide a strong empirical argument of the presented architecture as a workable and effective solution to autonomous ITSM. There are some cross-cutting observations that should be further discussed in analyses.

The theoretical relevance of the result is the most critical; rule-based automation demonstrated almost a total failure of the system when it comes to novel incident types with 18.6% classification accuracy as in Table 2, whereas the proposed GenAI system demonstrated the 79.4% accuracy with the usage of RAG-enhanced contextual reasoning. The observation confirms the main architectural hypothesis according to which pegging the output of generative AI to retrieved organizational knowledge will be the most meaningful generalization outside the frames of the pre-programmed response logic, which is the most serious limitation of the current AIOps strategies reported in the literature.

The Figure 4 dynamics of continuous learning offer empirical evidence that the adaptive behavior modeled in Equation (7). This gradual decrease in the rate of escalation between the approximately 38 percent at low incident volumes down to below 10 percent at cumulative

processing volumes of approximately 4,200 incidents indicates that the feedback-activated retraining pipeline is effective in deriving and encoding generalizable patterns of resolution based on the results of post-incident outcomes. This phenomenon is especially important when deployed to an enterprise situation, where the incident distributions change with the change of infrastructure configuration, requiring constant model adjustment to sustain performance.

The distribution of scores of the confidence in Figure 6 indicates that about 91.3 percent of experienced incidents were assigned to the confidence score with a value of 0.75 or higher, which is equal to the automation coverage rate in Table 1 (91.7 percent). The high correlation of these metrics which are determined independently verifies the confidence mechanism of scoring as well as the entire evaluation methodology and serves as a validation of the reliability of the reported performance figures. Moreover, the fact that low confidence cases are clustered around the novel and security violation subsets, as supported by Table 2, support the fact that the confidence estimation of the Validation Agent is a good indicator of the true diagnostic uncertainty and not systematic miscalibration.

5. Conclusion

The paper outlined a new architecture of automation of autonomous IT Service Management based on the integration of Generative AI and enterprise workflow automation platforms. The offered seven-layer system has shown significant and steady improvement in performance in all dimensions considered with a 66.3% decrease in mean time to resolution when simple incidents are considered, 63.6% when cascading failures are take into account, and an average classification accuracy of 89.9% that is much higher than a rule-based automation and a human-assisted baseline. The multi-agent orchestration model, which added RAG improved knowledge retrieval and specialized Diagnostic, Remediation and Validation agents, was especially useful in dealing with new forms of incidents that in the past revealed the weakness of traditional rule-based systems. The adaptive improvement was measurably sustained by the continuous learning mechanism that decreased the escalation rates to below 10% with increasing cumulative volume of incidence, as compared to the initial 38% in the duration of the evaluation process. The integrated layer of explainability and governance provides the assurance that autonomous decision-making is both auditable and meets enterprise regulatory standards, which is an important adoption barrier that has been noted in previous literature. Future research directions are to expand the architecture to allow multi-tenant enterprise deployments, to explore federated learning methods to update models privately across organizational boundaries, and to find the ways of integrating reinforcement learning to further optimize the choice of remediation strategies in the long term than the gradient-based update rule currently being used. Extension of the empirical evaluation framework to Change Management, Problem Management, and Service Catalog Management workflows represents a critical next step, with particular focus on measuring risk assessment accuracy in change advisory scenarios and KEDB auto-population quality in problem management contexts.

References

1. Guamushig, T.M.; Lopez, C.P.; Santorum, M.; Aguilar, J. Characterization of a Fourth Generation Virtual Organization Based on Industry 4.0. In Proceedings of the 2019 International Conference on Information Systems and Software Technologies ICI2ST, Quito, Ecuador, 13–15 November 2019; pp. 182–186. [**Google Scholar**] [**CrossRef**]

2. Aguilar, J.; Ardila, D.; Avendaño, A.; Macias, F.; White, C.; Gomez-Pulido, J.; Gutierrez de Mesa, J.; Garces-Jimenez, A. An Autonomic Cycle of Data Analysis Tasks for the Supervision of HVAC Systems of Smart Building. *Energies* **2020**, *13*, 3103. [**Google Scholar**] [**CrossRef**]
3. Aguilar, J.; Garcia, G. An Adaptive Intelligent Management System of Advertising for Social Networks: A Case Study of Facebook. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 20–32. [**Google Scholar**] [**CrossRef**]
4. Camarinha-matos, L.M.; Fornasiero, R.; Afsarmanesh, H. Collaborative Networks as a Core Enabler of Industry 4.0. In Proceedings of the Working Conference on Virtual Enterprises, Vicenza, Italy, 18–20 September 2017; Camarinha-Matos, L., Afsarmanesh, H., Fornasiero, R., Eds.; Springer: Cham, Switzerland, 2017; Volume 1, pp. 3–17. [**Google Scholar**]
5. Priego-roche, L.M.; Rieu, D.; Front, A. A Framework for Virtual Organization Requirements. *Requir. Eng.* **2016**, *21*, 439–460. [**Google Scholar**] [**CrossRef**]
6. Lopez, C.; Santorum, M.; Aguilar, J. FAVO: Framework of Autonomous Virtual Organizations Based on Industry 4.0. *Iber. J. Inf. Syst. Technol.* **2020**, *E27*, 333–345. [**Google Scholar**]
7. Monsalve-Pulido, J.; Aguilar, J.; Montoya, E.; Salazar, C. Autonomous recommender system architecture for virtual learning environments. *Appl. Comput. Inform.* **2024**, *20*, 69–88. [**Google Scholar**] [**CrossRef**]
8. Tang, Z.; Wang, W.; Zhou, Z.; Jiao, Y.; Xu, B.; Niu, B.; Zhou, X.; Li, G.; He, Y.; Zhou, W.; et al. LLM/Agent-as-Data-Analyst: A Survey. *arXiv* **2025**, arXiv:2509.23988. [**Google Scholar**]
9. Pizlo, W.; Parzonko, A.; Mazurkiewicz-Pizlo, A.; Parzonko, A.; Jedrzejczyk, I.; Borawski, P. Virtual Organizations: A Case Study of the Polish Agricultural Sector. *Eur. Res. Stud. J.* **2021**, *XXIV*, 361–371. [**Google Scholar**] [**CrossRef**]
10. Saka, A.B.; Chan, D.W.M. BIM divide: An international comparative analysis of perceived barriers to implementation of BIM in the construction industry. *J. Eng. Des. Technol.* **2023**, *21*, 1604–1632. [**Google Scholar**] [**CrossRef**]
11. Noh, S.C.; Abdul Karim, A.M. Design thinking mindset to enhance education 4.0 competitiveness in Malaysia. *Int. J. Eval. Res. Educ.* **2021**, *10*, 494–501. [**Google Scholar**] [**CrossRef**]
12. Gajek, A.; Fabiano, B.; Laurent, A.; Jensen, N. Process safety education of future employee 4.0 in Industry 4.0. *J. Loss Prev. Process Ind.* **2022**, *75*, 104691. [**Google Scholar**] [**CrossRef**]
13. Oliveira, K.K.D.S.; De Souza, R.A. Digital transformation towards education 4.0. *Inform. Educ.* **2022**, *21*, 283–309. [**Google Scholar**] [**CrossRef**]
14. Rane, N. Role of ChatGPT and Similar Generative Artificial Intelligence (AI) in Construction Industry. Available online: **https://ssrn.com/abstract=4598258** (accessed on 10 October 2023).
15. Lo, C.K. What is the impact of ChatGPT on education? A rapid review of the literature. *Educ. Sci.* **2023**, *13*, 410. [**Google Scholar**] [**CrossRef**]

16. Lameras, P.; Arnab, S. Power to the teachers: An exploratory review on artificial intelligence in education. *Information* **2021**, *13*, 14. [**Google Scholar**] [**CrossRef**]
17. Qu, J.; Zhao, Y.; Xie, Y. Artificial intelligence leads the reform of education models. *Syst. Res. Behav. Sci.* **2022**, *39*, 581–588. [**Google Scholar**] [**CrossRef**]
18. Kwon, J. A study on ethical awareness changes and education in artificial intelligence society. *Rev. D'intelligence Artif.* **2023**, *37*, 341. [**Google Scholar**] [**CrossRef**]
19. Rane, N.; Choudhary, S.; Rane, J. Education 4.0 and 5.0: Integrating Artificial Intelligence (AI) for Personalized and Adaptive Learning. Available online: **https://ssrn.com/abstract=4638365** (accessed on 2 November 2023).
20. Luan, H.; Tsai, C.C. A review of using machine learning approaches for precision education. *Educ. Technol. Soc.* **2021**, *24*, 250–266.
21. ServiceNow. AI for Change, Self-Service with Voice, and Digital End-User Experience — New in Zurich for ITSM. ServiceNow Community Blog, September 2025. Available: https://www.servicenow.com/community/itsm-blog/ai-for-change-self-service-with-voice-and-digital-end-user/ba-p/3362898