International Journal of Advanced Engineering Science and Information Technology (IJAESIT)

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Multi-Agent AI Frameworks for Autonomous Incident Management in IT Service Platforms

Dileep Valiki

Independent Researcher, India

valiki.dileep.researcher@gmail.com

ABSTRACT: Multi-Agent AI frameworks for autonomous lifecycle management of IT Service incidents are explored. Although considerable effort has been invested in making real-time operations more efficient, operational workloads continue to increase. Multi-Agent AI infrastructure offers a means to address the growing burden. The multi-agent AI ecosystem is composed of independent, autonomous, problem-oriented agents that collectively undertake responsibility for the lifecycle management of their area of expertise, thereby reducing the operational workload.

Operations Analytics, Service Telemetry and Observability, Policy, and Governance objectives are classified into a four-layer framework for incident lifecycle management. Each layer defines a collection of multi-agent AI objectives that are necessary for autonomous management of incidents across large-scale IT service operations. The objectives are expressed as a mix of high-level and granular activities. The results can guide AI-Actor development or integration into enterprise service platforms.

KEYWORDS: Multi-agent systems; autonomous systems; artificial intelligence; incident response; cyber-physical systems; IT service operation; cloud computing; federated learning; service management automation.

I. INTRODUCTION

Information technology (IT) incident management involves a variety of IT support services that identify, diagnose, and remediate incidents and problems in an IT ecosystem. Such ecosystems include applications, systems and services, hardware devices, and networks. These IT support services can be located inside an organization as well as outsourced, and incident detection can be either automated or manual. An incident can be further classified as either routine, emergency, or security, based on its characteristics. An incident is routine when it has an acceptable impact on service quality and is inside the predefined thresholds of IT services for the business. An incident is classified as emergency when it does not follow the business-as-usual conditions but requires some action from the IT support. An incident is classified as a security incident when there is a data breach.

Technological advancements in areas such as IT operations management (ITOM), site reliability engineering (SRE), observability, action-oriented dashboards, and natural language processing (NLP) have enabled emerging capabilities such as predictive maintenance, intelligent incident routing, incident prioritization, recommendation systems, noise reduction, self-healing, and automating later-stage activities. However, incident detection, diagnosis, and remediation are still predominantly performed via human effort in a reactive manner. The previous capabilities focus mainly on detecting quality degradation and streamlining support. The technical research and literature on incident management and security incident response continues to focus mainly on scaling people, accelerating response time, and improving skill levels via training.

1.1. Scope and Objective

Incidents affecting IT service platforms can be of varying sizes and complexity and can occur with different frequencies. The service-response time, be it manual or automatic, is critical and can be governed by current policy. The ultimate goal is to reduce the load on the manual action in carrying out incident response for IT services with the complete circulation of the closed-loop control and make it highly autonomous. AI with dedicated agents acts actively during the moment of incident detection, strives for a quick solution without human action, and can failover between services. Different agents in various roles are defined in the incident response and orchestration process.

The solution is aimed to achieve data ingestion and thermal temperature monitoring, telemetry on service utilization/responsiveness, security, policy adherence, incident response using tools like dynamic programmable networks and cloud patching, data compliance, governance assurance, and observability into a common platform supported by multiple agents co-existing in the ecosystem. Similar to autonomous vehicles, services for IT operations

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

can be built more resilient with minimal human intervention with voyeur agents taking control on behalf of man on monitoring data, service-level agreements and responding to incidents when the services degrade in performance.

Fig 1: Best multi agent ai systems for enterprise deployment

1.2. Background and Significance

Cyber incidents are on the rise. Cloud service providers are experiencing significant surges in demand, allowing for aggressive investment from the large technology enterprises enabling these platforms. Still, many individual service platforms remain susceptible to security incidents, outages, degrading incidents, and major outages, resulting in the need for substantial mitigation effort. Major incidents require the effective mobilization and coordination of thousands of skilled resources across many organizations and geographic locations. The complexity and scale of major incidents, combined with their relatively infrequent occurrence, usually limit the effectiveness of the mobilization and orchestration processes.

Multi-agent systems have the potential to enable autonomous incident management by directly supervising the complete incident lifecycle. They do so by orchestrating and managing optimal resource conditions for incident and anomaly handling efforts. They support the oversight and management of policy, compliance, and governance considerations, as well as the ingestion of new data sources. Multi-Agent AI frameworks utilize a combination of formally captured IT Service Management (ITSM) operational processes, human and machine expertise-aware automated orchestration, and execution of data ingestion, telemetry, observability, and control processes to achieve high levels of autonomy and reliability.

Equation 1: Incident response utility

Let:

- $a \in \mathcal{A}$: candidate action
- s: current incident state
- $U(a \mid s)$: utility of taking action $a$in state s

A standard decomposition is:

$$U(a \mid s) = B(a \mid s) - C(a \mid s) - R(a \mid s)$$

Derivation

Step 1: The article frames autonomous incident handling as selecting actions that improve service restoration and reduce burden.

So utility should increase with **benefit**:

$$U \propto B$$

Step 2: Actions consume resources and time. That introduces **cost**:

$$U \propto B - C$$

Step 3: The paper repeatedly emphasizes policy, governance, compliance, and risk. So an action with high risk or policy exposure should score lower:

$$U \propto B - C - R$$

Step 4: Converting proportional reasoning into an explicit scalar objective:

$$\boxed{U(a \mid s) = B(a \mid s) - C(a \mid s) - R(a \mid s)}$$

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Step 5: If we want weighted terms:

$$\boxed{U(a \mid s) = w_b B(a \mid s) - w_c C(a \mid s) - w_r R(a \mid s)}$$

This becomes the agent's decision rule:

$$a^* = \arg\max_{a \in \mathcal{A}} U(a \mid s)$$

II. FOUNDATIONS OF MULTI-AGENT SYSTEMS IN IT OPERATIONS

Multi-agent systems (MAS) offer a research and practical framework for developing IT operations that can adapt, coordinate, and act without human intervention in all aspects, including incident and problem management. These research avenues comprise MAS roles defined in the incident lifecycle, data ingestion and telemetry for observability, and policy, governance, and compliance areas. The autonomy levels and operational integration of agents into IT service platforms are assessed using simulated, animated, testbed, and production data.

A MAS is defined as a purposeful collection of software agents whose interactions lead to the realization of global goals. Each agent is an autonomous entity with its own knowledge base, goals, and internal tasks. Outsourcing is established from agents to partners, each specializing in an area of operations. A counterpart of an outsourcing agent is known as a service consumer. The global goal is realized through a coordinated sequence of inter-agent interactions—some agents delegate their goals and tasks to other agents and rely on delivery partners. The coordination can be synergy, negotiation, cooperation, or competition.

2.1. Definitions and Concepts

A multi-agent system (MAS) consists of multiple interacting agents working together to achieve a collective goal. Each agent is situated in a dynamic environment, equipped with sensors and actuators to monitor and interact with the surroundings. According to Wooldridge (2009), the agents have the capacity for autonomous action in their environment, exhibit a degree of social ability, and are designed or programmed to take the actions that will maximize their expected utility. While agents are located in the same environment and collaboratively pursue a shared objective, they are capable of performing individual, potentially conflicting actions without supervision. Mas however are not necessarily intelligent and can comprise finite-state machine agents. A multi-agent system can also feature a leader that coordinates the actions of other agents that lack the intelligence, knowledge, or competence to reach a solution by themselves. Agent autoencoders can be considered a special case of multi-agent systems in which the agents are required to produce an output that diverges from the initial input before re-encoding and recreating the data sample (Wang et al., 2018).

The term autonomous agent has a much narrower meaning, and is attributed to agents that act in a way similar to a human or robot. These agents exist in a dynamic environment, have their own goals and interests, represent their surroundings, and possess a reasoning capability that allows them to select the best action to perform at each moment (Rao et al., 1992; Hatzilygeroudis et al., 2008). Rao et al. further define autonomy as the capacity to achieve objectives without external intervention. An autonomous agent can operate independently to pursue a long-term goal. Currently, most systems for managing IT services in cloud environments operate in an assisted mode with human supervision.

2.2. Architectural Styles for Autonomous Agents

The development of architectures for autonomous agents can be categorized according to three distinct styles: reactive, deliberative, and hybrid. Reactive agents are simple systems that generate real-time responses to environmental stimuli via direct stimulus–response mappings. Autonomy emerges from the agents' ability to manage their internal states and collections of responses, while the environment is responsible for agent coordination. Natural and artificial systems alike can exemplify this approach—by definition, legless and eyeless predators are reactive agents, and in simple multi-agent settings, robots exhibiting only reactive behavior can perform swarm-like tasks.

Deliberative systems are capable of reasoning about the external world, including considering the effect of their own actions through internal simulations. The capacity to foresee consequences produces a plan that typically serves to coordinate action across a collection of agents. This deliberative activity is generally based on a formal model of the world, encompassing knowledge of how the world evolves on its own and how the agent affects it. Such agents are typically complex but—because actions are planned ahead of execution—often can be implemented in a simpler manner (e.g., visual sensors instead of sonar) in exchange for higher computational latency.

Hybrid architectures combine both capabilities, with a deliberative agent primarily functioning in planning mode while retaining a simpler architecture for obstacle avoidance during execution. A completely separate reactive layer fed by

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

percepts can also override the deliberative layer, opting to ignore plans in favor of immediate reactions when those vary significantly from the intended course of action. The positive signs of this strategy are evident in a variety of artificial and natural systems, as well as in current research.

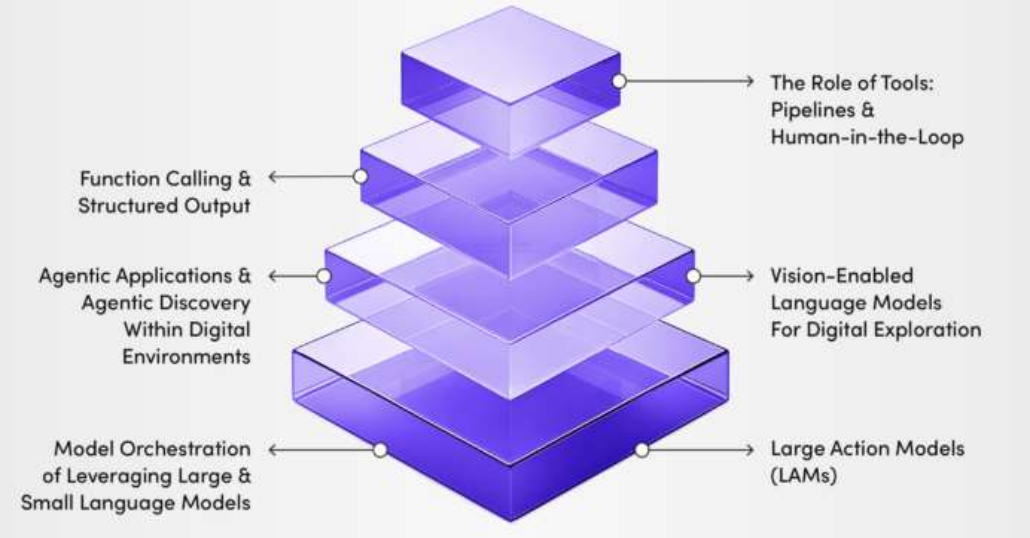


Fig 2: AI Agent Architecture

2.3. Inter-Agent Communication and Coordination

Two types of agent communications are needed—direct communication and broadcast for specific events—together with information exchange, or knowledge sharing, for collective decision making. Tasked agents create knowledge that is sent to the Agent Coordinator, which considers what, how, and when to send back the information. For rapid operational responses, a shortened version of the Agent Proposal Decision Making approach implements internal negotiation and then informs the Tasked agents what to do next from the set of proposals they have submitted. The approach minimizes the risk of duplicate work, for example resolving the same incident.

Agent coordination adds an experimental challenge, in that it requires the agents to negotiate both with themselves and with the other agents at the same time. Current implementations address only the supervision aspects after an incident has occurred, but future research is expected to develop the coordination agents further to take on more reflexive roles, supporting the incoming incident requests and data ingestion, and facilitating negotiation, coordination, and execution of combined agents workflows, together with real–time resolution before deployable changes are made.

Equation 2: Nash bargaining for agent negotiation

- $u_1(x), u_2(x)$: utilities of agents 1 and 2
- d_1, d_2: disagreement utilities

Then the Nash solution is:

$$\boxed{x^* = \arg\max_{x \in \mathcal{F}} (u_1(x) - d_1)(u_2(x) - d_2)}$$

Derivation

Step 1: Bargaining should reward outcomes that are **better than disagreement**:

$$u_i(x) - d_i$$

Step 2: For both agents to benefit, each surplus must be positive:

$$u_1(x) - d_1 > 0, u_2(x) - d_2 > 0$$

Step 3: Nash's idea is to choose the feasible agreement maximizing the **product of surpluses**:

$$\max_{x \in \mathcal{F}} (u_1(x) - d_1)(u_2(x) - d_2)$$

Step 4: Therefore:

$$\boxed{x^* = \arg\max_{x \in \mathcal{F}} (u_1(x) - d_1)(u_2(x) - d_2)}$$

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Log-form derivation
Sometimes it is easier to derive by taking logs:

$$\max_x \log((u_1 - d_1)(u_2 - d_2))$$

Using log rules:

$$\max_x [\log(u_1 - d_1) + \log(u_2 - d_2)]$$

So Nash bargaining is equivalent to maximizing the sum of log-surpluses.
For $n$agents:

$$x^* = \arg\max_{x \in \mathcal{F}} \prod_{i=1}^{n} (u_i(x) - d_i)$$

III. RESEARCH SUMMARY

Sources of autonomous multi-agent systems for incident and problem management in IT service platforms are investigated. The incident lifecycle is outlined, along with the major roles involved, together with key requirements encompassing data ingestion, telemetry, observability, policy, governance, and compliance. Concepts such as the service management lifecycle, IT operation command, federated and distributed data repositories, and risk management form additional building blocks. The use of a dedicated network link to a volatile data space, dedicated accelerator resources for training and learning, plus a reduced-order service graph improve performance. Complementary and enhance were simulated in hybrid environments to confirm main aspects of the model and learn lessons for future deployments.

Research in autonomous multi-agent systems for incident response in large-scale IT service platforms is analysed. The incidence lifecycle is defined and the important agent roles identified. The necessary conditions for effective operation encompass data ingestion, telemetry, observability, policy, governance and compliance. Further building blocks include the service management lifecycle, IT operation command, federated and distributed data repositories, risk management, a dedicated network link to a volatile data space, a reduced-order service graph for improved performance and dedicated accelerator resources for training and learning. Concepts complement and enhance each other, and simulation in hybrid environments confirms principal aspects of the model and provides lessons for future deployments.

3.1. Incident Lifecycle and Roles

All incidents in IT service platforms can be represented as an evolving series of states within a lifecycle model, from the initial creation of the incident, through its mitigation, to closure. Each incident traverses the complete set of agent roles, which can be depicted as a hypergraph whose vertices represent individual roles and edges determine the incident sequence through the roles of interest and their order of execution. The hypergraph can be transformed into an acyclic directed graph by collapsing each hyperedge into a single node, thus depicting the directed sequence of roles that are traversed during an incident.

Each incident will interact directly with one or more data sources according to the policies that are set for those sources. These sources can either detect incidents and be a trigger for the incident lifecycle or provide information in response to a specific request that can help in mitigating the incident when it is crossed. These direct interactions are often referred to as data ingestion. Observability and telemetry are higher-level constructs that deal with a set of sources that are monitoring some aspects of the service environment and provide information regarding the incident when asked, although the response may be delayed.

In modern incident management systems, each incident typically interacts directly with one or more data sources, guided by predefined policies that determine how and when these interactions occur. These data sources play a dual role: they can actively detect anomalies or failures and serve as triggers that initiate the incident lifecycle, or they can act as responsive repositories that provide critical information when queried during incident investigation and mitigation. This direct exchange of information between incidents and data sources is commonly referred to as data ingestion, a process that ensures relevant data is continuously collected, processed, and made available for analysis. Beyond this foundational layer, higher-level constructs such as observability and telemetry come into play, offering a more holistic view of system behavior. Observability encompasses the ability to infer the internal state of a system based on its outputs, while telemetry involves the automated collection and transmission of performance and usage data from various components within the service environment. Together, these constructs aggregate inputs from multiple monitoring sources—such as logs, metrics, and traces—to provide contextual insights into system health. Although

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

these sources are invaluable for diagnosing and mitigating incidents, their responses may not always be instantaneous, introducing potential delays that must be accounted for in incident response strategies. Consequently, effective incident management relies on a seamless integration of real-time data ingestion with broader observability frameworks, enabling teams to detect, analyze, and resolve issues with greater accuracy and efficiency.

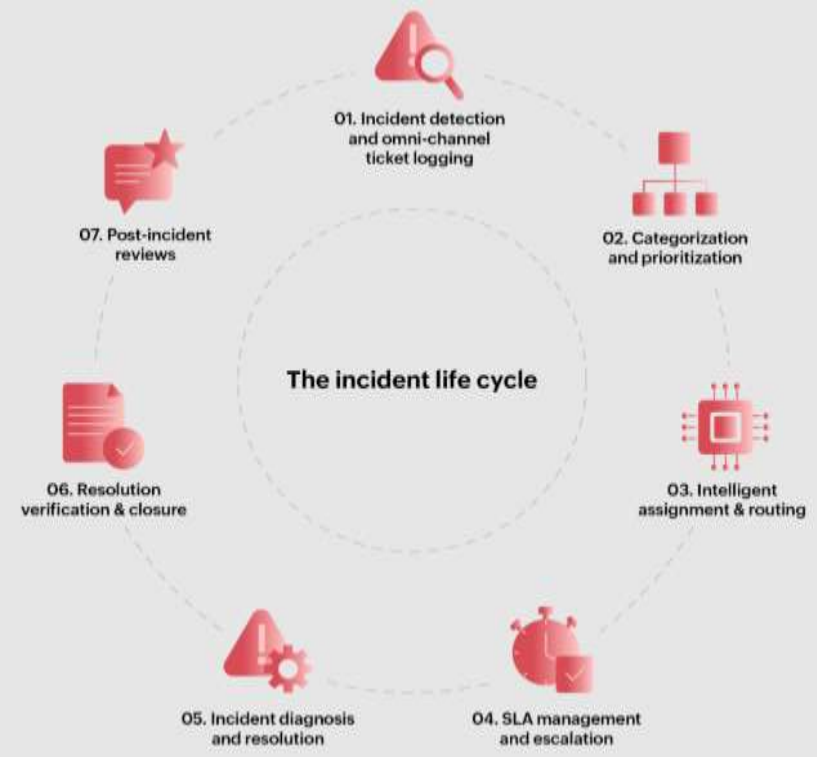


Fig 3: The incident life cycle

3.2. Data Ingestion, Telemetry, and Observability

Agents operating in a multi-agent framework ingest data from a variety of IT service platform telemetry sources, including conventional infrastructure and application performance monitoring systems, internal business monitoring services, and external feed integration platforms. The volume of telemetry is significant, supporting automated and/or manual incident identification and classification, also bringing telemetry-in-time to enhance incident response. Telemetry augmentation is done with internal and external AI-based services that infer missing information augmenting incident response analysis. Primary and secondary sources supply relevant and trustworthy data used during incident resolution. Multi-agent systems enable these agents to balance incident response data quality with an increasing data quantity, a key challenge in AI.

A framework built upon proprietary, open, and academic research implements a security, governace and compliance monitoring agent. Policy engines ingest definition, observability data, templates, and alerts, enabling the execution of policy lookup and verification of asset policy compliance status. Policy-compliant data processing and policy engine design fulfillment enable a policy agent to execute operations in an IT service platform. The agents are designed and built to integrate with AI safety and their security, governance and compliance monitoring roles alignment is key to successfully deploy their use.

3.3. Policy, Governance, and Compliance

Policies define the acceptable state of systems, applications, and services. Non-compliance and deviations are considered incidents, for which an autonomous agent acts as a custodian. A checkpoint agent manages the checks, creating justification records for small deviations and incidents for changes beyond tolerance ranges. If multiple policies conflict, the agent resolves the dispute using priority levels assigned by a governance agent. State change approval requires consensus among custodians and the governance agent. When a policy dispute involves an incident, the implicit lead agent retains the incident's priority.

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Policy agents are responsible for defining and publishing policies. A set of policies along with the authorizing signature is stored in an immutable ledger. Configuring decision points in systems, applications, or services involves a change request and approval from custodians and the governance agent. Automated compliance checks and successful change requests signal that the decision point is operating within policy tolerance level. If monitored values fall outside tolerance, the policy agent investigates. Minor deviations require justification. Major deviations trigger an incident with the compliance agent as implicit lead. The compliance agent also provides the approval signature in cases where two or more policy custodians disagree.

Equation 3: Federated consensus for incident severity prediction

Let:

- x: telemetry feature vector
- Agent $i$predicts $\hat{y}_i = f_i(x)$
- w_i: trust/expertise weight, with $\sum_i w_i = 1$

Consensus prediction:

$$\boxed{\hat{y} = \sum_{i=1}^{n} w_i \hat{y}_i}$$

Derivation

Step 1: Each expert agent gives its own severity estimate:

$$\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$$

Step 2: Not all agents are equally reliable, so use a weighted average:

$$\hat{y} = \frac{\sum_{i=1}^{n} w_i \hat{y}_i}{\sum_{i=1}^{n} w_i}$$

Step 3: If weights are normalized so that $\sum_i w_i = 1$, the denominator disappears:

$$\boxed{\hat{y} = \sum_{i=1}^{n} w_i \hat{y}_i}$$

Deriving weights from expertise

If expertise score of agent $i$is e_i, normalize it:

$$w_i = \frac{e_i}{\sum_{j=1}^{n} e_j}$$

Substitute into consensus:

$$\hat{y} = \sum_{i=1}^{n} \frac{e_i}{\sum_{j=1}^{n} e_j} \hat{y}_i$$

$$\boxed{\hat{y} = \frac{\sum_{i=1}^{n} e_i \hat{y}_i}{\sum_{i=1}^{n} e_i}}$$

Variance-aware version

If each agent has uncertainty σ_i^2, inverse-variance weighting gives:

$$w_i = \frac{1/\sigma_i^2}{\sum_{j=1}^{n} 1/\sigma_j^2}$$

Hence:

$$\boxed{\hat{y} = \frac{\sum_{i=1}^{n} \hat{y}_i /\sigma_i^2}{\sum_{i=1}^{n} 1/\sigma_i^2}}$$

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

IV. OBJECTIVE OF THE STUDY

A multi-agent architecture for autonomous management of security incidents in IT service platforms is postulated. Security incidents are formalized as a distributed, asynchronous, multi-stage process involving multiple roles including data acquisition, analysis, response, and remediation. Integration with modern technologies such as observability and real-time telemetry pipelines is explored. Furthermore, the implications for Policy, Governance, and Compliance (PGC) are examined. A catalogue of autonomous agent roles required to handle security incidents is identified.

The automation of cloud infrastructure management expands the attack surface available to adversaries. An entire ecosystem of processes and technologies exists to minimize the number and impact of security incidents. Research focuses on the feasibility of autonomous agent-based architectures for managing some or all of them, based on underlying technology capabilities and the essential requirements of the private sector. Modern observability solutions enable near real-time analysis of data, while PGC matrices define policies and procedures to ensure compliance with laws and regulations. Mapping the security incident life cycle and interaction of the various roles and responsibilities forms a first step in exploration, followed by a catalogue of incident-related software agents. Various preparatory and kinesthetic workflows are adduced with reference to recent implementations of network defence mechanisms in hybrid environments.

4.1. Agent Roles for Incident Response

A multi-agent framework for autonomous incident incident management in IT Service Platforms can be conceptualized through the lens of the lifecycle of an incident and the roles of the agents involved as defined by the ITIL organization. The data ingestion, telemetry, and observability workstreams, these three agent role groups also cover the aspects of policy, governance, and compliance, including service, security, and SLO/SLA management. These serve as the basis for the articulation and evaluation of data sources, pipelines, and services, which can be fiscal and physical in their nature. Three further agent roles, namely orc- help- and info-master, draw from the orchestration layer found in traditional Enterprise Service Buses. They provide workflow management integration services, coordination of unbudgeted requests, and high-level information and queries to external entities. The preparation of the review data is articulated through the pioneering work of Condition-Resource-Policy-Data.

Agents tasked with incident response are located at the center of the multi-agent model. The role group covers omniscient autonomic and cognitive management systems as the epitomés of an operational-technology demarcation. Conflict resolution among the two incident-resolution agents—hirer and triage—parallels scheduling and budget-consumption considerations among service-management agents. Communication graphics depict the operation of machine-learning-based training-and-learning stages that provide archives, and rules; and the non-functional properties of the incident-response agents emphasize the fit-for-purpose exploration of non-functional properties.

4.2. Orchestration and Workflow Management

Agent orchestration takes high-level decisions about autonomous incident-management agents and their interactions, including when to enable/disable certain roles and specify priorities for concurrent actions. Score functions balancing autonomy, collaboration, and risk desirability guide this process, providing input for scheduling and negotiation. Workflow management focuses on executing an agent cooperation plan, keeping other agents informed and retracting roles when actions succeed or reach deadlocks.

Well-known orchestration models, such as the Event-Condition-Action or Business Process Execution Language pattern, together with adaptive Agent-Orchestration Architecture, can be adopted or tailored to the specific context. The Multi-Agent-Based Orchestration Framework for Security Assessment can also be repurposed. The detailed planning of interaction sequences can sometimes be avoided by using established frameworks coupled with negotiation capabilities. Even without such a global plan, autonomous agents can self-organize and propose collaboration at run time. Testing part of these models with player-controlled agents in a simulation fully supported collaborative behaviour.

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

Fig 4: AI Agent Orchestration Frameworks

4.3. Learning, Adaptation, and Feedback Loops

Autonomous agents are expected to learn from the experience and adapt their decision-making based on feedback from the environment or objective entities. Learning, adaptation and feedback can occur directly through training of models (e.g., Machine Learning) or indirectly through the exploration of new actions and collecting their outcomes. In a fully autonomous setting, agents undertake actions or sequences of actions (policies) that yield the most beneficial outcome according to a reward function. For such agents, feedback is built into the policy learning framework; feedback from multiple incidents improves future responses to similar incidents.

More common today, in semi-autonomous settings, the agent is aided by a user-in-the-loop setup where the agent provides situation awareness through monitoring or observability and provides suggestions or recommendations that are confirmed, vetoed or modified by the operator. Feedback can be enforced by auditing operations and enforcing compliance with alerts issued by an operation decision-support system or by a process compliance tool. Reward assignment may be formal through training of promoted helper agent models or informal through lessons learned discourses in communities of practice.

Equation 4: Orchestration score for action selection

Let:

- A_k: autonomy score of plan k
- C_k: collaboration score
- R_k: risk penalty
- P_k: policy-compliance score

Then a natural orchestration objective is:

$$\boxed{S_k = \alpha A_k + \beta C_k + \gamma P_k - \delta R_k}$$

and choose

$$\boxed{k^* = \arg\max_k S_k}$$

Derivation

Step 1: The article says good orchestration should favor autonomy:

$$S_k \propto A_k$$

Step 2: It also values coordinated multi-agent behavior:

$$S_k \propto A_k + C_k$$

Step 3: Since policy/governance matter, compliant plans should score higher:

$$S_k \propto A_k + C_k + P_k$$

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Step 4: Risk should reduce desirability:

$$S_k \propto A_k + C_k + P_k - R_k$$

Step 5: Introduce tunable importance weights:

$$\boxed{S_k = \alpha A_k + \beta C_k + \gamma P_k - \delta R_k}$$

Constraint form

Sometimes policy is hard-constrained rather than rewarded:

$$\max_k \ \alpha A_k + \beta C_k - \delta R_k$$

subject to

$$P_k \geq P_{\min}$$

V. METHODOLOGY

Negotiation and consensus among agents play an essential role in timely and proper resolution of dynamic incidents. Incidents that are harmless and self-heal can be ignored with minimal damage. Conflicting resolutions for the same incident should be gracefully avoided by establishing agent priorities. Human-AI interaction should allow humans to remain in charge of the resolution process while preventing cognitive overload.

A well-defined architecture helps streamline multiagent deployment across on-premises and cloud infrastructures. The cloud-native model leverages auto-scaling and orchestration capabilities of public cloud providers. In a hybrid setup, the primary service is hosted in the public cloud while securing sensitive customer data in a private cloud or on-premises. Scalability and low-latency access to resources are important factors for deploying a fully automated agent ecosystem in the public cloud. An on-premises architecture is preferred for autonomous learning in finance and law enforcement sectors, where research on sensitive data may be needed.

5.1. Negotiation and Consensus Mechanisms

The design of mechanisms for negotiation and consensus among autonomous agents responding to incidents follows the general principles for multi-agent systems. Negotiation provides a structured means for forming agreements among agents while appropriately distributing benefits and costs. Conditions for reaching a stable outcome are formally captured as the Nash Bargaining Solution, and solution mechanisms are provided for the simpler case of resource allocation among self-interested agents.

Several consensus mechanisms based on Federated Learning are described for non-self-interested agents facing the task of jointly predicting the severity of an incident. The prediction is based on shared information, such as telemetry data, that has been appropriately labelled by agents holding expert knowledge. Consensus among the agents is essential for project completion at a minimum deadline with minimal cost while ignoring the interest of individual agents, as required by agents with mutually displaced priorities. It is shown that accurate severity prediction is possible when the number of agents with expertise grows linearly with the number of considered telemetry sources.

Equation 5: Learning and feedback via Bellman / Q-update

For a state s, action a, reward r, next state s', the Q-learning update is:

$$\boxed{Q_{t+1}(s,a) = Q_t(s,a) + \eta \left[r + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a) \right]}$$

Derivation

Step 1: Define long-term value of taking action $a$in state s:

$$Q(s,a) = \text{expected discounted future reward}$$

Step 2: Bellman optimality says optimal value equals immediate reward plus discounted best future value:

$$Q^*(s,a) = \mathbb{E}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a \right]$$

Step 3: In practice, the true expectation is unknown, so use the observed sample target:

$$\text{Target} = r + \gamma \max_{a'} Q_t(s',a')$$

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

Step 4: Update old estimate toward target with learning rate η:

$$Q_{t+1}(s,a) = Q_t(s,a) + \eta(\text{Target} - Q_t(s,a))$$

Step 5: Substitute the target:

$$\boxed{Q_{t+1}(s,a) = Q_t(s,a) + \eta\left[r + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)\right]}$$

Policy extraction

Once $Q$is learned:

$$\boxed{\pi^*(s) = \arg\max_a Q(s,a)}$$

5.2. Conflict Resolution and Priority Handling

An autonomous agent or agent group may be required to attend to multiple requests generated either by other agents or by human operators. Such cases can arise when individual agents lack sufficient expertise, resources, or capabilities to perform a specific task, or when task delegation out of personal motivation is not possible or desired. In such instances, agents must prioritize among pending requests, negotiate with each other for task acquisition, and ascertain whether the task is warranted in the presence of system governance policies. The absence of task requests does not negate ongoing processing. Auxiliary agents proactively pull data from the environment to reconcile the respective domains and act based on determined trigger conditions. Trigger conditions define when the agent should perform a task in the absence of any request for assistance or supervision. For example, agents monitor for data volume attained within the environment, and if reached, participate in executing a data archiving task.

In multi-agent systems, conflict resolution is necessary during resource acquisition and negotiation, which may use formalized protocols such as contract nets, delegations, and bargaining approaches. The underlying rationale for the negotiation process is based on the following conditions. First, the agent capable of handling the request may not have sufficient resources or lacks personal motivation to attend to the task. In such scenarios, other agents—interested in acquiring more resources or motivated by user policy to assist the request-originating agent—formulate and send requests. Second, multiple agents having the means to perform the task may compete for it based on expertise or precedence. In addressing the conflict, the agent of highest priority performs the task, based on precedence of both intention and ownership of resources. Priority is determined using discretionary filters, such as dependency on trigger management or data ownership.

5.3. Human–Agent Interaction and Oversight

To ensure accountability, autonomy alone is insufficient. Agents require a mechanism for invoking human assistance in situations beyond their competence. Such scenarios may arise when agents have low confidence in a decision or prediction, encounter unforeseen difficulties in executing actions, or face urgent decisions that threaten data privacy or service integrity. Human checks should not be overly frequent, as this undermines the advantages of autonomous operation. A well-designed threshold, perhaps based on predicted risk, can make such checks effective.

The extent to which human intervention follows the normal or emergency modality depends on factors such as risk and speed. In the normal case, a human operator can inject real-time comments, suggest corrective actions, and verify the execution log for reliability. In the emergency scenario, rapid decision-making with delegation of execution to the agents is critical for success. Since grasping the overall picture of a complex incident is inherently difficult for a single operator, it is proposed that agent-based data visualization techniques assist operators by extracting reasoning results from the various agents involved. Specialized agents may also provide suggestions for available actions or directions.

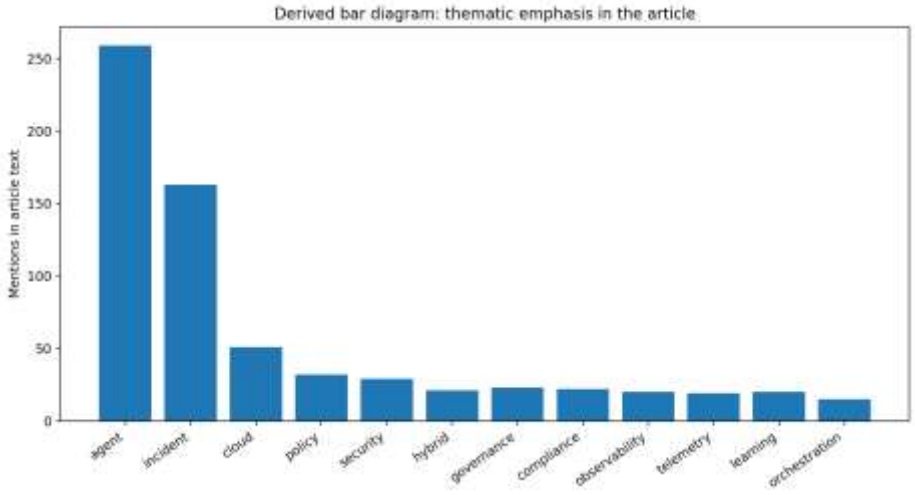
VI. RESULT

The growing importance of provenance knowledge within IT infrastructures is the main driver behind the collection and management of provenance information in almost all aspects of incident response and process automation. Provenance information must be created, stored, updated, and made available for archiving and forensic operations. Lineage data of data mutations and transformations, decisions taken, and implication of those decisions in the form of data subsets excluded from the operations for different reasons, should be made available for review to all involved agents during the incident lifecycle to support better informed decision making.

The security, access control, and ethical aspects of the agent operations become an important topic for consideration when dealing with autonomous systems. The roles used for agent decision making and execution, need to be well defined and restricted according to need. The information available in an incident can be a sensitive and confidential

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

information and general mechanisms of access control based on data classification may not be sufficient. Autonomous agents in enterprise systems do not understand risks and ethical implications of their operations. The feedback provided by the world, which guides their behavior in the long run, is based on the outcome of their actions and is not aware of the harm or other implications it may have on the external world. Decisions that take into consideration ethical aspects, minimizing damage to people, environment, and world in general, need to be explicitly included in the design of automated systems capable of learning from its operations.



6.1. Data Provenance and Lineage

Maintaining provenance and lineage is an important requirement for operational readiness, particularly in highly regulated industries such as financial services and healthcare. It is crucial to leverage data to ensure compliance with relevant policies, laws, and regulations. Although many data sources providing off-the-shelf capabilities are available, they are often not sufficiently traceable to support governance needs.

Data provenance in any data sharing cooperation establishes where shared data originates (data source) and where shared data goes (data destination). Data sharing systems maintain data provenance metadata. Suppose that the data source or destination corresponds to one of the partners in a data sharing cooperation. In that case, the authentication and authorization capabilities of a trusted third party can be employed to determine whether a data creator is authorized to create data or whether a data destination can be reached.

6.2. Security and Access Control

A multi-agent framework for autonomous incident management involves the consolidation of multiple agents within predefined functional areas. The combination of different agents allows for negotiation and consensus determination, as well as oversight capabilities. The framework addresses security and access control, including how policies defined within organizations can be assured in an automated manner with minimal human intervention.

Security represents a significant worry in any system, particularly in a system that is partially autonomous during an incident. The agents would have access to almost all aspects of the infrastructure, and if a malicious actor was capable of interfering with the agent's own channels, a major incident could be created. Such circumstances would be unfavourable for the organization, and placing detection of such manipulation in the agent's purview and supervision would assist the organization in those cases.

6.3. Ethics and Responsible AI

As AI systems assume autonomous decision-making capabilities, ethical policies must govern their behaviour. A classical approach is for systems to only act and communicate in accordance with human-defined policies. However, ethical decision-making in complex environments generally cannot be completely codified: the possible actions of an intelligent agent are rarely fully enumerable, and the consequences of actions may differ from expectations. Behaviour that exceeds defined policies may be ethically defensible in response to a genuine emergency created by other agents. Consequently, the lack of suitable policy warrants the ability to exercise judgement in exceptional circumstances.

Extensions to existing theoretical models allow agents to negotiate and improve policies through multi-issue, multi-agent negotiation. In this context, an ethical value set allows agents to evaluate the moral difference between two

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

policies, enabling agents to avoid neglecting sensitive ethical issues while being mindful of implementation costs. In cases where stakes are higher than the cost of negotiations, agents are motivated to collaboratively improve ethical policy that better reflect community values by shifting limited resources from everyday operations to collective discussions.

VII. EVALUATION AND VALIDATION FRAMEWORKS

Traditional benchmarks and prototypes in dynamic and adaptive environments may not provide sufficient validation for autonomous AI systems. New evaluation frameworks, metrics, and techniques are required, incorporating methodological guidance and simulation environments.

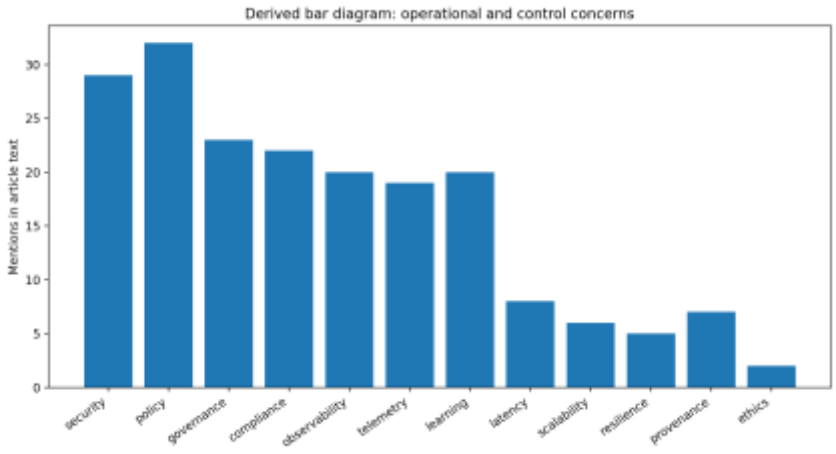
Agility, generality, and resource scarcity may push autonomous agents into situations where their behaviour cannot be straightforwardly deduced from their manifest policies. Reliable validation in such situations requires a dedicated framework that can accommodate requirements emerging from diverse domains and application areas, combining rigorous test-driven arguments with experimental, modelling, and simulation techniques. Such requirements include validation of the agents' behaviours at different levels of granularity and abstraction, and the behaviour of the deployed agent system as a whole. A closely related concern is a practical evaluation metric for autonomy that captures the system resources consumed for a given task.

Multiple testbeds support the examination of autonomous agent behaviour across paradigms, including robot soccer, robotic trading, and predator–prey simulations. The ongoing exploration of negotiation in heterogeneous domains includes two scenarios that exemplify autonomy at different levels of granularity: data centre automata exchanging offers for data storage, and security agent car dealers negotiating contractual obligations with human players. Other simulation frameworks have been built and used for learning and adaptation in complex, dynamic, multi-player settings.

7.1. Evaluation Metrics for Autonomy

Care must be taken when analysing the degree of autonomy of the agents, especially when third-party and self-trained models are used to fulfil their tasks and provide decisions. This is important for several reasons, including: a) service or business decisions can not be assigned without a proper understand of the ability of the agents, particularly in relation to Human Trust; b) ensuring that the autonomy level is calibrated to the real capabilities of the agents, taking bort into account external considerations like the area's maturity at using AI-based services; c) providing to business owners a better understanding of the impact of the Single Point of Failure (SPoF) in terms of company risks of significant issues in the situation involved when using the agent.

The most logical reasoning behind these considerations is that agents should behave without the need of human intervention. Following this line of thought the Work Group established the following for the agent's autonomous capability: "Learn from historical data, both in the agents' training stage and in providing on-going feedback loops during operation. Based on this learning agents should decide and execute actions especially when a sense of urgent, critical or importance is present". This therefore does not consider the usage of a Generic Decision Making Model or any other approach since it is based on the autonomous capability development.



|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

7.2. Simulation Environments and Testbeds

A taxonomy of simulation environments supporting the autonomous management of incidents in IT services is defined at different levels of abstraction. The architectures of environments and testbeds developed in the domain are reviewed through the lens of the taxonomy. Environments enable the evaluation of autonomous agents for their non-functional properties, whereas testbeds assess the role of negotiation, consensus-building, and conflict-resolution mechanisms and the performance of dedicated agent teams. Hypervisors provide an abstraction layer that creates an environment for simulating new scenarios.

Cyber-physical systems serve as physical testbeds for experimental research in AI and machine learning, involving real robots navigating real terrains, detecting presence and movements, interacting with physical actuators, and communicating with exposed APIs. Cloud gaming environments remove the challenge of building scalable cloud-hosting infrastructures, providing enough game sessions for experimentation in a collaborative cloud-gaming platform. An experimental agent organization currently being designed provides the mechanism for explaining to end-users mission status and results and for achieving user acceptance of agents in charge of incident response.

7.3. Benchmarking and Real-World Validation

From an application perspective, use-case-specific multi-agent frameworks can give rise to clusters of interrelated agent roles across multiple execution styles. These frameworks can be realised using a combination of modelling, simulation, and experimental evaluation pipelines that study behaviour at different abstraction levels. Acting as a virtual testbed, a modelling environment can initially examine relationships in agent interactions, their pros and cons for incident response within an IT-service context, and how these contribute toward the overall incident-management lifecycle. Responses to metrics that quantify the degree of autonomy for the incident-management process can help identify areas for further refinement.

However, a comprehensive assessment requires the integration of a fully featured agent ecosystem, increasingly populated by domain-specific concepts. Supporting an autonomous incident management service at scale entails maturing and testing every operational layer—from the readiness of individual autonomous agent behaviours, through clusters of agent roles for focused incident lifecycle phases, to complex adaptability and learning curriculums that can change how the complete service operates over time. These mix an array of formal modelling techniques with interaction-oriented evaluation, spanning artificial life, reinforcement learning, and multi-agent method definition.

At an extreme level of detail, large parts of a contemporary IT service platform can be emulated using simulation and virtualisation, enabling the operational integrity of an incident-management service to be scrutinised under realistic conditions across a continual cycle of proofs, fixes, and reproofs, before deployment within the production service itself. Post-emulation, an operational incident-management service moves into formal production support, where the focus shifts from technical correctness to service quality, and from automated delivery to high-availability production demand across all operational processes.

VIII. DEPLOYMENT ARCHITECTURES AND OPERATIONAL CONSIDERATIONS

Opportunity for implementing a framework for research on multi-agent coordination Mechanisms and implementation experiments on a cloud-native multi-agent platform for incident response.

Successfully deploying multi-agent frameworks in cloud-native, hybrid, or on-premises IT cloud environments will provide empirical evidence for the feasibility of autonomous incident management and auxiliary incident service management actions across domains, such as public, private, or hybrid clouds, as well as multi-cloud combinations of different service vendors. The association of autonomous agents with a professed goal, assigned capabilities, resources, and budget is mapped to the monitored data supplied to data-agnostic service agents driving the incident operation. Latency, scalability, and resilience are first-order design attributes of the agent ecosystem supporting the ITOps process goal of rapid service restoration, not service completion. A cloud-native deployment environment closes the cloud management stack.

Business-server agents, incident agents, orchestration agents, policy governance agents, vendor, service provider and service resolvers, security and ethical agents, predictive infrastructure resource agents and network connection agents, policy requirements, name resolution and storage-binding services will be supported by reservoir- or queuing-based service agent telemetry observation and data-storage-binding support of traditional telemetry for move-gather or move-place-move agents. Scalable isolation for data-privacy-compliance security and risk agents of trained networks and data-bound service agents is ensured by the cloud business, data, and network isolation boundaries. Visibility and

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|
DOI: 10.15662/IJAESIT.2024.0706006

observability of service-device and service-data-level reflection and rules for agent behaviour are supplied by the agent policy-specification and-effect-enforcement functions.

Layer	Primary objectives	Representative agents or functions	Key outputs
Governance	Resolve policy conflicts, approve state changes, maintain oversight	Governance agent, custodians, ledger controls	Priorities, approvals, signed policies
Policy & Compliance	Check tolerance breaches, verify adherence, trigger compliance incidents	Policy agent, checkpoint agent, compliance agent	Compliance status, incidents, justifications
Telemetry & Observability	Ingest signals, enrich alerts, provide monitoring context	Telemetry agents, observability services, external feeds	Context-rich alerts, quality data, situational awareness
Incident Operations	Detect, classify, route, diagnose, remediate, verify, and close incidents	Triage agent, resolver agents, orchestration agents	Mitigated incidents, closure records, review inputs

Table : Four-layer autonomous incident management framework

8.1. Cloud-Native and On-Premises Integrations

Cloud-native incident response platforms, such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure, provide built-in capabilities to automatically respond to service disruptions and security threats via incident management functionality. Multi-Agent Systems (MAS) can play a vital role in orchestrating incident response in such cloud-native environments, and these environments can offer support for the agent-based architectures described previously, thereby making cloud-native incident response platforms attractive candidates for autonomously managing incidents.

While the above-mentioned platforms have their own in-house capabilities for managing service disruptions and security threats, hybrid IT service environments that encompass other clouds and on-premise data centres will need collaborative agent-based integration with these cloud-native platforms as well as their corresponding incident response capabilities when a significant incident occurs. Multi-Agent Systems (MAS) are suited to orchestrate such incident resources spanning multiple hybrid IT service platforms. An important contradiction to consider in hybrid environments is latency in data transmission between on-premise datacenters and the three main cloud platforms, AWS, Azure and GCP. While there is an inherent risk of a data breach, augmenting these traditional mechanisms that rely on detection rules can help manage service disruptions, security threats and compliance checks.

8.2. Scalability, Latency, and Resilience

The performance and service level of an IT service platform is determined, among others, by its scalability, latency, and resilience. Scalability is the capacity of a system to accommodate increasing workload without significant performance degradation. Each agent running on the platform must be able to handle the workload for which it was designed. Further, in a typical incident lifecycle, the capacity required by each agent may increase and decrease over time as the incident evolves, such as when more queries are being processed. For cloud-native agent platforms, this elasticity can be gracefully handled by the built-in scalability properties of cloud-native solutions. For on-premises installations, the agents can be implemented in load-balanced clusters that can scale in and out based on the load.
Latency is the time required to detect, respond to, and resolve an incident. Each agent tackles a specific aspect of the entire incident response, and therefore, it must perform its specific task in a timely manner to ensure that the overall delay remains acceptable. During an incident response, multiple agents collaborate at various stages. They can co-locate and run on the same or adjacent servers to minimize the network latency. Resilience is the ability to continue operating despite failures. Each agent must be tolerant of any failures. For example, if an agent that processes incoming queries fails, the service should continue to serve incoming queries with a significant increase in processing time until the fault is either repaired or mitigated by a substitute.

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

8.3. Observability and Monitoring of Agent Ecosystems

Moreover, given the inherently probabilistic nature of multi-agent systems, the observability and monitoring of collectively-navigated processes remain challenging. To address this challenge, it is recommended to consider the introduction of dedicated agents assigned to such tasks. These agents may collaborate to model and replicate the ecosystem's behavior through monitoring, event logging, and metadata construction. An approach that engages these agents in deriving maps of the ecosystem under numerous possible scenarios could be especially insightful. The deployment of such an agent monitoring ecosystem could occur as part of user-administered governance, possibly with the initial iteration limited to observing and measuring a specific process rather than the entire set of autonomously operated processes.

Although observability is vital for maintaining user trust in the independence of an AI ecosystem, observability alone is not sufficient for ensuring the ethical and responsible application of multi-agent AI/Digital-Twin environments. Moreover, dedicated multi-agent agents can also provide assistance with respect to ethical and responsible AI concerns and thereby help improve both user and ecosystem interactions.

IX. CASE STUDIES AND EMPIRICAL EVIDENCE

Multi-Agent AI Frameworks for Autonomous Incident Management in IT Service Platforms
Faculty of Computer Science, University of Constance, GermanyDistributed and Hybrid Cloud-Native IT Service Platforms, Entropy-Driven Processes, and Major Incident ResolutionRecent years have seen the rise of large-scale distributed cloud-native IT service platforms that combine public and private cloud services using several integration technologies. Their operation relies on custom-developed, distributed microservices complemented by COTS services from multiple providers that are mainly operated on a Public Cloud Infrastructure.

Adopting these frameworks typically promises substantial business benefits, making them desirable. However, despite the attraction and extensive feasibility analyses, evidence of successful adoption remains scarce. Worldwide major incidents have been reported on these platforms, and the inability to execute an automated resolution during the subsequent service downtime raises questions. Incident Management must adapt to these Cloud-Native IT Service Platforms, acknowledging that the service delivered by these platforms is a temporary point in time and primarily driven by business entropy. During a major incident, the resolution process shifts from “the service is down but there is no impact” to “the service is down and there is an impact,” signifying a major incident.

The major incident process is interpreted as a set of roles and their responsibilities throughout the incident lifecycle. Large-scale incident resolution requires informed and qualified decision-making, which, in the absence of other resolutions, becomes a guessing game. Acceptance of inappropriate compromise speeds the resolution, but tests the service. Multi-Agent Systems provide an architectural style for implementing intelligence, and agents acting on behalf of incident process roles support these systems. Research thus embeds known IT operations concepts from the ITIL framework and its supporting literature within Multi-Agent Systems to enable intelligent agents that operate autonomously and with a high degree of sophistication. Silver and bronze agent roles support the Incident Management process during major incidents.

9.1. Large-Scale IT Service Platforms

Despite pre-emptive strategies to avoid interruptions in IT services, incidents still occur. The majority cannot be resolved without substantial human intervention, requiring costly resources and possibly resulting in serious reputational or financial damage. Multi-agent frameworks aim to manage incidents in an autonomous manner akin to self-driving systems, where AI and autonomous regulations are combined to reduce service disruptions and minimize business costs.

The bodies of funding agencies and enterprise consortia are supporting the development of large-scale IT service platforms capable of self-governance. These are not purely connected infrastructure clouds but consist of IaaS, PaaS, SaaS, and XaaS environments that can interact with people, systems, and mobile electronic devices. The totality can provide services for incident management in digitally reproducible hybrid environments, where enterprise workloads are distributed across enterprise clouds, telecommunication clouds, National Research and Education Networks, external multi-clouds, or edge resources. Multi-agent frameworks are designed for incident self-management within large cloud-native service platforms, with a federated development model.

9.2. Incident Response in Hybrid Environments

For IT systems, deployed at very large scales, in hybrid environments, comprising both cloud-scale service components and on-premise systems, lab-based incident-response testbeds were integrated with a realistic, high-fidelity, cloud-scale

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

service platform setup in service-integration testbeds where incidents and outages could be served from an actual SECoE. The service-integration testbeds, however, reflected human-in-the-loop interactions in the incident response. The incident-response capability was further expanded within a real-world, fully on-premise IT setup at the IT service stage of a major Asian university, addressing security and surveillance, resource management, and application-support incident response. Earlier research work also investigated testing and calibrating autonomous incident-response capability in a SECoE on small-scale cloud-scale IT service platforms in cloud-native and services mode before actual deployment on larger-scale platforms. Lessons learned from the work highlighted the importance of a good combination of human oversight and multi-agent systems capability for improving alert handling and achieving faster incident resolution.

From a hybrid-organisational perspective, multiple department security with central-department surveillance, across departments, was emphasised, together with the need for a place for human-agent collaboration in the incident-response ecosystem. From a hybrid-service perspective, incident categorisation of incident-response actions into those exclusively agents would handle, those needing some human intervention but completing with the agent assistance for quicker resolution, and those that agents would not attempt analysing the telemetry or logs was found necessary for operational efficiency. A hybrid incident-response on-premise SECoE with the capabilities and necessary governance and monitoring as-a-service for components, other than servers with redundancy, was proposed.

9.3. Lessons Learned and Best Practices

An overarching architectural framework for an agent-based approach to IT incident management has been delineated. Since incidents can arise from multiple reasons and impact multiple services, the research effort focuses on the incident lifecycle from inception to closure and on various agent roles that collectively perform these lifecycle functions. Each role is not necessarily realized by a dedicated, separate agent; multiple roles can be assumed by a single agent and vice versa based on policy requirements. The research has moved to a new phase focusing on the orchestration of agents and their workflows. Agent-based systems operate in supervised mode; for certain tasks, agent actions need to be validated and approved by human managers. Policy and governance checks are embedded into the orchestration. Each incident is assigned to a lead agent, which, based on insights derived from the ongoing incident as well as historical data, coordinates the actions of other discrete or supervisory agents and triggers incident closure.

Work is in progress to design a generic simulation environment for benchmarking and evaluating the agent-assist capabilities in large-scale production or hybrid platforms or those under cyberattack. Two distinct scenarios are being pursued: managing incidents in a large IT service platform with a cloud-native launcher of integrated microservices; and managing incidents through collaborative MLOps in a hybrid infrastructure with on-premises data and learning models. These scenarios serve to identify agent roles relevant to incident response, their action patterns and interactions, and candidate learning policies and feedback mechanisms for subsequent operational sessions. Best practices for autonomous incident management are being derived from real-world implementations along the dimensions of observability, security, ethics, and responsible AI.

X. CONCLUSION

Distributed Multi-Agent Architectures (MAA) for Autonomous Governance, Management, and Operation of Multidisciplinary, Multienterprise IT Service Infrastructure Dedicated to Servicing Non-Business-Critical or Business-Critical Digital Service Platforms Support Full Support for Support Issue Lifecycle Management and Non-Business-Critical or Business-Critical Incident Lifecycles Based on the Continuous Adaptation of Policies and Workflows with Guaranteed Support of Business Continuity with Maximum Security and Minimum Cost Delivery of Digital Services. The Two-layered Multi-agent Architecture Realizes the Administration and Management with Theorizing-based Governance of Multiple Multi-Control Capacity Such as Policy, Governance, Compliance, and Risk Management While the Multi-agent Architecture Non-Business-Critical or Business-Critical Incident Management of the Continuous Agentizing of the Incident Management of Multiple Non-business-critical or Business-Critical Digital Services Driven by Data Processing and Machine Learning Resilience and Continuous Adaptation Through Orchestration and Operations with Micro Workflows to Guarantee the Security and Manage the Resources during the Continuous Change in Business Volume and Related Services Hosting with Performance Affecting the Business Continuity of a Hybrid Combination of Self-Service Dynamic Compute Capacity Environment Supported by Self-Service Load Balancing in Public Cloud and Non Data-residency Requirements Public Cloud and Private Cloud Deployment Models and Solutions.

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

REFERENCES

[1] Nagabhyru, K. C. (2023). From Data Silos to Knowledge Graphs: Architecting CrossEnterprise AI Solutions for Scalability and Trust. Available at SSRN 5697663.
[2] Meda, R. (2024). Predictive Maintenance of Spray Equipment Using Machine Learning in Paint Application Services. European Data Science Journal (EDSJ) p-ISSN, 3050-9572.
[3] Sheelam, G. K., & Koppolu, H. K. R. (2024). From Transistors to Intelligence: Semiconductor Architectures Empowering Agentic AI in 5G and Beyond. Journal of Computational Analy- sis and Applications(JoCAAA), 33(08), 4518-4537.
[4] Bandi, V. D. V. K. (2024). Automated Feature Engineering Systems in Large-Scale Healthcare Data Environments. Journal of Neonatal Surgery, 13.
[5] Mangalampalli, B. M. (2024). AI-Enhanced Data Governance: Automating Compliance In Healthcare Analytics Platforms. The Review of Diabetic Studies, 191-204.
[6] Nagabhyru, K. C. (2023). Accelerating Digital Transformation with AI Driven Data Engineering: Industry Case Studies from Cloud and IoT Domains. Educational Administration: Theory and Practice, 29(4), 5898-5910.
[7] Bandi, V. D. V. K. (2024). AI-Driven Predictive Risk Modeling Architectures for Financial Systems. International Journal Of Finance, 37(3), 54-7.
[8] Kolla, S. K. (2023). Big Data–Driven Machine Learning Frameworks for Clinical Risk Prediction. International Journal of Medical Toxicology and Legal Medicine, 26(3), 44-59.
[9] Pamisetty, V. (2024). Transforming taxation systems through predictive analytics and AI-driven compliance monitoring tools. Am Data Sci J Adv Comput, 3, 55-68.
[10]Garapati, R. S. (2022). AI-Augmented Virtual Health Assistant: A Web-Based Solution for Personalized Medication Management and Patient Engagement. Available at SSRN 5639650.
[11]Nagabhyru, K. C. (2024). Data Engineering in the Age of Large Language Models: Transforming Data Access, Curation, and Enterprise Interpretation. Computer Fraud and Security.
[12] Meda, R. (2024). Enhancing Paint Formula Innovation Using Generative AI and Historical Data Analytics. American Advanced Journal for Emerging Disciplinaries (AAJED) ISSN, 3067-4190.
[13] Aitha, A. R. (2023). CloudBased Microservices Architecture for Seamless Insurance Policy Administration. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 607-632.
[14]Sheelam, G. K. (2024). Towards autonomic wireless systems: integrating agentic AI with advanced semiconductor technologies in telecommunications. Am. Online J. Sci. Eng., 3(4), 234-256.
[15] Yandamuri, U. S. (2022). Big Data Pipelines for Cross-Domain Decision Support: A Cloud-Centric Approach. International Journal of Scientific Research and Modern Technology (IJSRMT).
[16] Amistapuram, K. (2024). Federated Learning for Cross-Carrier Insurance Fraud Detection: Secure Multi-Institutional Collaboration. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 6727-6738.
[17] Reddy Segireddy, A. (2024). Federated Cloud Approaches for Multi-Regional Payment Messaging Systems. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(2), 442-450.
[18] Aitha, A. R. (2023). Cloud-Native Big Data AI/ML Framework for Risk Intelligence and Fraud Control in Banking and Insurance Ecosystems. Available at SSRN 6157967.
[19] Kolla, T. (2023). Predictive ETL Failure Detection in Healthcare Data Pipelines Using Anomaly Detection Algorithms. International Journal of Medical Toxicology & Legal Medicine.
[20] Gottimukkala, V. R. R. (2022). Licensing Innovation in the Financial Messaging Ecosystem: Business Models and Global Compliance Impact. International Journal of Scientific Research and Modern Technology, 1(12), 177-186.
[21] Pamisetty, V. (2024). AI-Driven Decision Support for Taxation and Unclaimed Property Management: Enhancing Efficiency through Big Data and Cloud Integration. Available at SSRN 5250776.
[22] Kolla, S. H. (2023). Deep Learning–Driven Retrieval-Augmented Generation for Enterprise ITSM Automation: A Governance-Aligned Large Language Model Architecture. Journal of Computational Analysis and Applications, 31(4).
[23] Aitha, A. R. (2022). Cloud Native ETL Pipelines for Real Time Claims Processing in Large Scale Insurers. Available at SSRN 5532601.
[24] Meda, R. (2023). Data Engineering Architectures for Scalable AI in Paint Manufacturing Operations. European data science journal.
[25] Gottimukkala, V. R. R. (2023). Privacy-Preserving Machine Learning Models for Transaction Monitoring in Global Banking Networks. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 633-652.
[26]Davuluri, P. N. Integrating Artificial Intelligence into Event-Driven Financial Crime Compliance Platforms.
[27] Kolla, S. H. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10, 495-506.
[28] Mahesh Recharla, "Integrated Genomic and Neurobiological Pathway Mapping for Early Detection of Alzheimer's Disease," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), DOI: 10.17148/IJARCCE.2023.12122.

[29] Bandi, V. D. V. K. (2024). Intelligent Data Platforms For Personalized Retail Analytics At Scale. Metallurgical and Materials Engineering, 30 (4), 1011–1027.
[30] Mangala, N. (2022). Implementing Databricks Unity Catalog For Centralized Data Governance In Multi-Business-Unitenterprises. Journal of International Crisis and Risk Communication Research , 101–122. https://doi.org/10.63278/jicrcr.vi.3738.
[31] Davuluri, P. N. AI-Augmented Sanctions Screening: Enhancing Accuracy and Latency in Real Time Compliance Systems.
[32] Meda, R. (2024). Agentic AI in Multi-Tiered Paint Supply Chains: A Case Study on Efficiency and Responsiveness. Journal of Compu-tational Analysis and Applications (JoCAAA), 33(08), 3994-4015.
[33] Inala, R., & Somu, B. (2024). Agentic AI in Retail Banking: Redefining Customer Service and Financial Decision-Making. Journal of Artificial Intelligence and Big Data Disciplines, 1(1).
[34]Recharla, M. (2024). Advances in Therapeutic Strategies for Alzheimer's Disease: Bridging Basic Research and Clinical Applications. American Online Journal of Science and Engineering (AOJSE)(ISSN: 3067-1140), 2(1).
[35] Segireddy, A. R. (2024). Machine Learning-Driven Anomaly Detection in CI/CD Pipelines for Financial Applications. Journal of Computational Analysis and Applications, 33(8).
[36] Mangalampalli, B. M. Intelligent Data Profiling for Healthcare Data Lakes Using AI-Enhanced Analytics.
[37] Amistapuram, K. (2024). Smart Decision Support Systems For Dynamic Tax Policy Optimization Using Reinforcement Learning. Available at SSRN 6143426
[38]Kolla, S. K. (2024). Federated Machine Learning On Big Healthcare Data For Privacy-Preserving Analytics. The Review of Diabetic Studies, 175-190.
[39] Singireddy, J. (2024). AI-Driven Payroll Systems: Ensuring Compliance and Reducing Human Error. American Data Science Journal for Advanced Computations (ADSJAC) ISSN, 3067-4166.
[40] Yandamuri, U. S. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. International Journal Of Finance, 36(6), 682-706.
[41] Pamisetty, A. (2024). Leveraging Agentic AI and Cloud Infrastructure for Predictive Logistics in National Food Supply Chains. Available at SSRN 5262994.
[42] Deep Learning-Driven Optimization of ISO 20022 Protocol Stacks for Secure Cross-Border Messaging. (2024). MSW Management Journal, 34(2), 1545-1554.
[43] Pamisetty, A. (2024). Leveraging Big Data Engineering for Predictive Analytics in Wholesale Product Logistics. Available at SSRN 5231473.
[44] Kolla, S. K. (2023). Explainable AI and ML Models for Transparent Clinical Decision Support. Journal for ReAttach Therapy and Developmental Diversities, 6, 2444-2460.
[45] Singireddy, J. (2023). Finance 4.0: Predictive analytics for financial risk management using AI. European Journal of Analytics and Artificial Intelligence (EJAAI) p-ISSN, 3050-9556.
[46] Mangala, N. (2021). Optimizing Large-Scale ETL Pipelines Using Medallion Architecture on Azure Data Lake. Journal of Artificial Intelligence and Big Data, 1(1), 1-20. https://doi.org/10.31586/jaibd.2021.1361
[47] Valiki, D., & Segireddy, A. R. (2023). Deep Learning Architectures Deployed on Cloud Platforms for Dynamic Financial Risk Evaluation and Market Prediction. American International Journal of Computer Science and Technology, 5(5), 12-24.
[48] Nandan, B. P. (2024). Semiconductor Process Innovation: Leveraging Big Data for Real-Time Decision-Making. Journal of Computational Analysis and Applications (JoCAAA), 33(08), 4038-4053.
[49] Singireddy, J. (2024). Deep Learning Architectures for Automated Fraud Detection in Payroll and Financial Management Services: Towards Safer Small Business Transactions. Journal of Artificial Intelligence and Big Data Disciplines, 1(1), 75-85.
[50] Yandamuri, U. S. AI-Driven Decision Support Systems for Operational Optimization in Hospitality Technology
[51] Sheelam, G. K. (2024). Deep Learning-Based Protocol Stack Optimization in High-Density 5G Environments. European Advanced Journal for Science & Engineering (EAJSE)-p-ISSN, 3050-9696.
[52] Pamisetty, A., Adusupalli, B., Mashetty, S., & Singreddy, S. (2024). Redefining Financial Risk Strategies: The Integration of Smart Automation, Secure Access Systems, and Predictive Intelligence in Insurance, Lending, and Asset Management. Sneha, Redefining Financial Risk Strategies: The Integration of Smart Automation, Secure Access Systems, and Predictive Intelligence in Insurance, Lending, and Asset Management (December 05, 2024).
[53] Kolla, S. H. (2024). RETRIEVAL-AUGMENTED GENERATION WITH SMALL LLMS FOR KNOWLEDGE-DRIVEN DECISION AUTOMATION IN ENTERPRISE SERVICE PLATFORMS. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(3), 476-486.
[54] Singreddy, S. (2024). Applying deep learning to mobile home and flood insurance risk evaluation. Available at SSRN 5238946.
[55] Garapati, R. S. (2023). Optimizing Energy Consumption in Smart Build-ings Through Web-Integrated AI and Cloud-Driven Control Systems.

|ISSN 2349-3216| Volume 7, Issue 6, November-December 2024| Bimonthly, Peer Reviewed and Scholarly Indexed Journal|

DOI: 10.15662/IJAESIT.2024.0706006

[56] Inala, R. (2022). Engineering Data Products for Investment Analytics: The Role of Product Master Data and Scalable Big Data Solutions. International Journal of Scientific Research and Modern Technology, 155-171.
[57] Nagubandi, A. R. (2023). Advanced Multi-Agent AI Systems for Autonomous Reconciliation Across Enterprise Multi-Counterparty Derivatives, Collateral, and Accounting Platforms. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 653-674.
[58] Singireddy, S. (2024). The Integration of AI and Machine Learning in Transforming Underwriting and Risk Assessment Across Personal and Commercial Insurance Lines. Journal of Computational Analy- sis and Applications(JoCAAA), 33(08), 3966-3991.
[59] Kummari, D. N. (2023). AI-powered demand forecasting for automotive components: A multi-supplier data fusion approach. European Advanced Journal for Emerging Technologies (EAJET)-p-ISSN, 3050-9734.
[60] Mangalampalli, B. M. Generative AI Applications In Healthcare Data Mart Design And Optimization.
[61] Kalisetty, S., & Singireddy, J. (2023). Optimizing Tax Preparation and Filing Services: A Comparative Study of Traditional Methods and AI Augmented Tax Compliance Frameworks. Available at SSRN 5206185.
[62] Mangala, N. (2022). Real-Time Data Quality Monitoring and Gating Frameworks in Cloud-Based Data Pipelines. International Journal of Research and Applied Innovations, 5(6), 8197-8219.
[63] Inala, R. (2023). Big Data Architectures for Modernizing Customer Master Systems in Group Insurance and Retirement Planning. Educational Administration: Theory and Practice, 29 (4), 5493–5505
[64] Nandan, B. P. (2024). Revolutionizing Semiconductor Chip Design through Generative AI and Reinforcement Learning: A Novel Approach to Mask Patterning and Resolution Enhancement. International Journal of Medical Toxicology and Legal Medicine, 27(5), 759-772.
[65] Kolla, T. (2024). AI-Powered Data Catalog Systems For Healthcare Data Discovery And Governance. South Eastern European Journal of Public Health, 2296–2311. https://doi.org/10.70135/seejph.vi.7077
[66] Kumar, V., & Singh, R. (2024). AI-driven service management frameworks for automated incident resolution in cloud platforms.
[67] Kummari, D. N., & Burugulla, J. K. R. (2023). Decision Support Systems for Government Auditing: The Role of AI in Ensuring Transparency and Compliance. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 493-532.
[68] Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.