International Journal of Applied Science and Engineering
Citation: *IJASE*: **13**(02): 297-307, December 2025
DOI: 10.30954/2322-0465.2.2025.10

REVIEW PAPER COMPUTER SCIENCE

Agentic AI and the Future of Cloud Computing Architecture

Sumitra Das[1] and Rahul Kumar Chawda[2*]

[1]*Department of Computer Science, Assam University, Silchar, India*
[2]*Department of Computer Science, Assam University, Silchar, India*

**Corresponding author: rahul.chawda3@gmail.com*

Received: 25 Sept., 2025 **Revised:** 23 Nov., 2025 **Accepted:** 02 Dec., 2025

ABSTRACT

Cloud computing has made it easier for many industries to use computing resources whenever they need them and to scale up when demand increases. However, modern cloud systems have become very complex because they rely on containers, microservices, serverless functions, and edge devices. Managing these systems through manual effort or fixed rule-based automation is becoming increasingly difficult. Agentic Artificial Intelligence (Agentic AI) can transform how cloud systems are managed. Unlike traditional AI, which requires step-by-step human instructions, Agentic AI can observe the system, understand what is happening, plan actions, execute them, and improve based on feedback. This paper discusses how combining Agentic AI with cloud-native technologies can support cloud systems that can organize themselves, recover from failures, and scale automatically. It also examines the key design choices, security and trust concerns, real-world use cases, and future research directions. Overall, the paper presents Agentic AI as an important foundation for next-generation autonomous cloud platforms.

Keywords: Agentic AI, Cloud Computing Architecture, Cloud-Native Systems, Kubernetes Orchestration, Microservices, Serverless Computing, Edge Computing

Cloud computing underpins today's digital ecosystem, supporting large web services, data analytics pipelines, and IoT deployments by offering on-demand access to compute, storage, and networking resources[1,2]. As these platforms grow to serve millions of users and increasingly distributed microservices, day-to-day operations become far more complex. Yet, in many real-world deployments, cloud management still depends on manual configuration, reactive scaling, continuous human oversight, and fixed threshold rules, which often fall short when strong performance, high reliability, and real-time responsiveness are required at scale[3,4].

How to cite this article: Das, S. and Chawda, R.K. (2025). Agentic AI and the Future of Cloud Computing Architecture. *IJASE.*, **13**(02): 297-307.

Source of Support: None; **Conflict of Interest:** None

Over the last few years, AI for IT Operations (AIOps) has improved parts of the cloud management workflow. By applying machine learning to logs, metrics, and traces, AIOps systems can support anomaly detection, incident prediction, and capacity planning[5]. In practice, however, most AIOps deployments remain primarily advisory. They often highlight symptoms and correlations, but multi-step remediation still depends heavily on human-run runbooks and operator judgment, including actions such as coordinated traffic shifting, rollback decisions, and placement changes across tiers. This reliance can become a bottleneck in cloud environments where failures propagate through service dependencies and workload conditions change quickly.

Agentic AI offers a natural next step for such settings. In this paper, Agentic AI refers to systems that can observe their environment, reason about objectives, plan multi-step actions, execute those actions through tools or APIs, and improve using feedback[6,7]. Conventional predictive models mainly estimate future events, whereas agentic systems combine prediction with deliberation and adaptive control. These capabilities are particularly relevant for cloud platforms, where workloads fluctuate, constraints are multi-objective (latency, availability, and cost), and operational decisions often require coordinated actions across services[8]. Emerging research suggests that agentic approaches can shift cloud management from reactive monitoring to proactive, closed-loop orchestration, improving reliability and cost efficiency when deployed with appropriate constraints[9,10].

At the same time, cloud-native technologies provide the programmable base needed to operationalize such autonomy. Microservices and containers support modular deployment, while orchestration platforms such as Kubernetes provide declarative control, automated scaling, and fault recovery mechanisms[11]. Serverless platforms simplify event-driven execution by abstracting infrastructure details. Edge computing extends cloud capabilities closer to data sources to support latency-sensitive applications[12]. Together, these technologies expose both rich telemetry signals and strong actuation mechanisms, such as scaling, routing, placement, and rollout controls, that an intelligent control layer can coordinate. Table 1 below shows the operational progression from traditional control to agent-driven autonomy, illustrating how cloud systems have evolved over time.

Table 1: Cloud operations progression: from manual control to agent-driven autonomy

Capability	**Traditional Ops**	**AIOps**	**Agentic AI Direction (This Paper)**
What it mainly does	Manual monitoring and reactive actions	Detect/triage with ML	Goal-driven planning with controlled execution
Typical output	Alerts and dashboards	Anomalies, correlations, recommendations	Multi-step action plans aligned with SLOs
Execution style	Human-run procedures	Often human-triggered	Tool/API- driven, policy- constrained actions
Main limitation	Slow, operator bottleneck	Limited autonomy; weak coordination	Needs strong security, trust, and governance

Motivated by this convergence, we argue for an *Agentic Intelligence Layer* that can coordinate and automate cloud operations across the edge-cloud continuum. Such a layer would continuously monitor telemetry, forecast resource demand, detect failures, refine operational policies, and adjust system configurations to maintain service-level objectives. At the same time, because agentic control increases the range and impact of automated actions in mission-critical environments, it must be designed with strong safeguards for security, trust, and governance.

Accordingly, this paper presents a reference architecture that reflects real deployments, with heterogeneous users and devices at the edge, distributed edge nodes, a containerized microservices layer orchestrated by Kubernetes, serverless runtimes for event-driven tasks, and a cloud data center providing scalable core resources. In this architecture, the *Agentic Intelligence Layer* functions as a cross-cutting control plane that observes system state and makes decisions through cloud-native mechanisms such as routing policies, placement rules, and autoscalers. We also discuss application domains, including smart hospitals, connected healthcare, autonomous manufacturing, cyber-physical systems, smart cities, and cost-optimization platforms, where low latency, high availability, and elastic resource management make agent-driven autonomy particularly relevant.

LITERATURE REVIEW

Cloud-Native Foundations and Control Surfaces

Cloud-native systems use microservices and containers to make applications easier to develop, deploy, and scale in small parts. This flexibility comes with added operational complexity, since services depend on one another, releases happen frequently, and failures can spread across the system[2]. Orchestration platforms such as Kubernetes address part of this challenge through reconciliation-based control loops that continuously work to keep the system aligned with the declared desired state. They also provide elastic scaling and basic self-healing features[13]. In most cases, however, these controls operate at a local level, such as a single service or a single cluster, and they are not built to optimize global goals across multiple services, including end-to-end service-level objectives (SLOs), cost limits, and risk constraints[14]. Recent studies also point to continuing bottlenecks in areas such as container scheduling, storage management, and event-driven networking, which can affect reliability and efficiency at large scale.

Observability as a Prerequisite for Autonomy

Cloud observability has moved beyond basic monitoring to include richer telemetry, such as logs, metrics, and traces, which makes it easier to diagnose problems in distributed systems. This telemetry is critical for any intelligent control layer because it captures the system state needed to detect performance degradations, identify likely causes, and confirm whether recovery actions are actually working. Service meshes and distributed tracing are especially useful here, since they provide dependency-aware visibility that supports coordinated remediation in microservice architectures[11,12].

AIOps: Progress and Current Limitations

AIOps uses machine learning on operational data to improve tasks such as anomaly detection, event correlation, root-cause analysis, and forecasting. Recent survey studies suggest that the field has matured and that large language models are increasingly being used to support log interpretation, incident summarization, and interactive troubleshooting. Even so, many AIOps deployments remain largely recommendation-driven: they help teams diagnose issues and propose likely actions, but safe multi-step execution and verification are often limited by governance requirements, fragmented tooling, and the risk of unintended changes in production systems[15].

From AIOps to Agentic/AgentOps: Toward End-to-End Incident Lifecycle Automation

A notable recent direction is the shift from stand-alone AIOps functions to agent-based systems that can support a broader portion of the incident life-cycle. For example, Microsoft Research's AIOpsLab introduces a comprehensive framework for evaluating AI agents in incident management and underscores the importance of standardized interfaces between agents and cloud platforms, along with reproducible evaluation setups. At the same time, studies on LLM-driven AIOps in cloud environments suggest that language models can act as practical assistants or controllers for operational work-flows. These works also make clear that deploying such approaches in production requires careful attention to safety controls, reliable verification of actions, and tight integration with existing tools and processes[16].

Autonomic Computing and MAPE-K as a Conceptual grounding

Autonomic computing, often described through the MAPE-K loop, offers a well-established way to think about closed-loop adaptation. The idea is straightforward: a system monitors its signals, analyzes deviations from expected behavior, plans corrective steps, executes changes, and updates what it knows based on the outcome. In cloud environments, many self-managing behaviors already exist in limited forms, such as per-service autoscaling. However, applying MAPE-K at a cloud-wide level remains challenging because the knowledge base must bring together telemetry from multiple layers, capture service dependencies, and respect operational policies and constraints. Ongoing research continues to apply MAPE-K principles to cloud-edge orchestration, highlighting that feedback-driven control remains a core requirement for achieving performance and efficiency in large distributed systems[17,18].

Edge-Cloud Orchestration and AI-Driven Resource Management

Edge computing brings in constraints that are much less prominent in a single, centralized cloud, including strict latency requirements, limited bandwidth, device mobility, and highly diverse hardware. These factors make orchestration harder because decisions about where to run tasks and how to manage resources must be made across a more dynamic and heterogeneous environment. Surveys on edge-cloud collaboration and mobile edge computing (MEC) consistently point to challenges such as task offloading, resource allocation, and service migration, where decisions often need to be taken under uncertainty. Recent survey work also highlights how cloud platforms, edge intelligence, and AI for IoT are increasingly converging, reinforcing the need for smarter, coordinated control across the continuum[19,20].

Table 2: Literature positioning: Why an Agentic Intelligence Layer is needed

Area	What it improves	What typically remains missing	Key latest anchors
Cloud-native control (Kubernetes/service mesh)	Deployment automation, local scaling, and basic recovery mechanisms	Cross-service planning under joint SLO and cost constraints	Runtime scheduling limitations and operational bottlenecks[13]
AIOps	Anomaly detection, correlation analysis, and forecasting	Safe multi-step autonomy with execution and verification loops	Recent surveys on AIOps and log-based analysis[15]

Agent-based / AgentOps direction	Coverage of larger portions of the incident life-cycle	Standardized evaluation and trustworthy actuation mechanisms	Emerging agent-centric AIOps frame works and evaluation methods[16]
Edge-cloud orchestration	Latency-aware workload placement and task off loading	Unified control with integrated learning and governance	Recent advances in edge-cloud coordination architectures[17,18]

Research Gaps Motivating an Agentic Intelligence Layer

Across existing studies, three recurring gaps help explain why agentic approaches are gaining attention in cloud operations.

1. Current control mechanisms are mostly designed around local objectives and short feedback loops. They often optimize a single service or a narrow signal, while AIOps solutions frequently stop at detection and recommendation rather than producing and executing coordinated, multi-step remediation plans.
2. Autonomous action in production environments raises governance concerns. Any system that can change routing, scaling, or placement must operate within clear limits, including policy checks, least-privilege access, reliable rollback paths, and auditable decision records.
3. There is no standard evaluation approach, and the community lacks widely adopted incident suites and benchmarks to compare autonomous operations approaches fairly.
4. Together, these gaps motivate the *Agentic Intelligence Layer* proposed in this paper. The layer is intended to coordinate cloud-native actuators such as scaling, routing, placement, and rollout controls using telemetry-driven, goal-aware planning, while enforcing the security and governance safeguards required to maintain service-level objectives in dynamic edge-cloud environments.

Reference Architecture for Agentic Cloud Computing

This section presents a reference architecture for integrating Agentic AI with cloud-native platforms to support self-organizing, self-healing, and self-scaling behavior across the edge–cloud continuum. The design assumes that cloud-native systems already provide two essential ingredients: (i) continuous telemetry through logs, metrics, and traces, and (ii) operational actuators such as scaling, routing, placement, and rollout mechanisms. The proposed *Agentic Intelligence Layer* builds on these primitives to coordinate decisions under explicit objectives and constraints.

Architecture Overview

The reference architecture is organized around four deployment tiers. At the outermost tier, a diverse set of users, sensors, and IoT devices generates requests and continuous data streams. These workloads are first handled by distributed edge nodes, which provide low-latency computation close to where data is produced. Behind the edge, application logic is implemented as containerized microservices managed by Kubernetes, with service-to-service communication governed by cloud-native networking controls such as service mesh policies.

Event-driven tasks run on serverless runtimes, while cloud data centers provide elastic compute and storage for large-scale processing, long-term persistence, and global coordination. Across these tiers, the *Agentic Intelligence Layer* functions as a cross-cutting control plane. It continuously gathers telemetry, builds an operational view of system health and dependencies, and reasons over explicit objectives such as latency and availability targets, throughput requirements, and cost budgets. Based on this context, it selects safe, bounded actions using existing cloud-native mechanisms and then verifies the impact of those actions through ongoing observation to ensure that service-level objectives are maintained.

Conceptual Figure

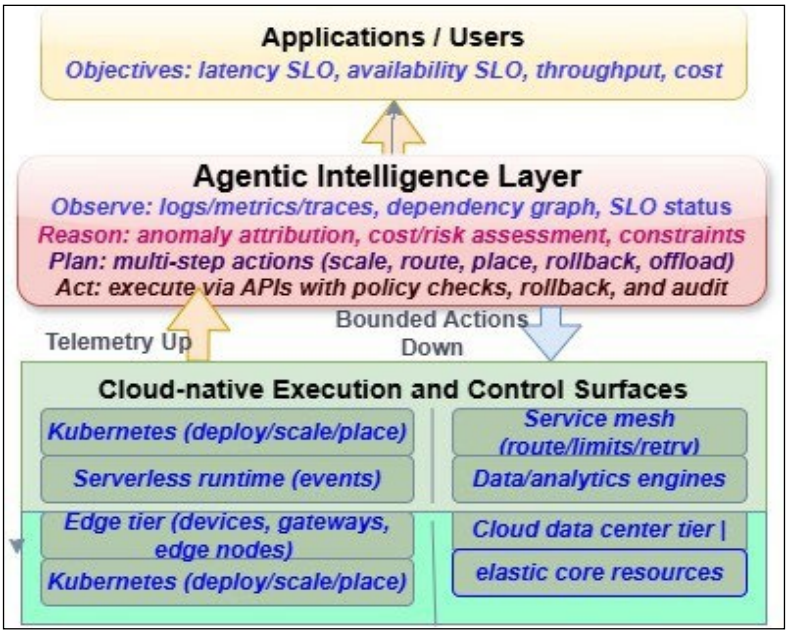

Fig. 1: Conceptual architecture with an Agentic Intelligence Layer spanning edge-cloud

Core Functions of the Agentic Intelligence Layer

The Agentic Intelligence Layer can be viewed as a practical closed-loop controller that supports four core functions:

1. First, it observes the system and constructs state by aggregating telemetry into an operational picture that includes service health signals, dependency relationships, and current SLO compliance. This matters because raw logs, metrics, and traces are typically scattered across tools and do not, on their own, provide a coordinated view.
2. Second, it reasons under explicit objectives and constraints by examining deviations from SLO targets while accounting for operational boundaries such as cost limits, approved change policies, and permissible action scopes. This helps avoid improvements in one dimension that create unacceptable trade-offs elsewhere, such as reducing latency by scaling far beyond budget.

3. Third, it plans multi-step responses when a single corrective action is insufficient. For instance, a tail-latency degradation may require a combination of traffic shifting, scaling a downstream bottleneck, or reverting a recent release, followed by checks that the system is stabilizing.
4. Finally, it executes safely and verifies outcomes by applying changes through existing cloud-native interfaces and then validating impact using ongoing telemetry. If results do not improve or regressions appear, the layer should revert changes through defined rollback procedures and retain an auditable record of decisions and actions.

The proposed architecture is not intended to replace Kubernetes or other cloud-native building blocks. Instead, it treats them as reliable execution mechanisms and brings them together through a goal-driven control loop. In practice, the Agentic Intelligence Layer interacts with the platform through a small set of well-defined operational touchpoints. At the Kubernetes layer, it can adjust replica counts, apply placement constraints, tune resource allocation policies within approved limits, and manage rollouts. At the networking layer, service mesh controls enable traffic splitting, rate limiting, and resilience policies such as retries, timeouts, and circuit breaking. For event-driven workloads, serverless platforms expose controls such as concurrency limits and scaling parameters. Finally, in edge-cloud deployments, the layer can decide where particular pipeline stages should execute, choosing between edge nodes and the core cloud based on latency sensitivity and bandwidth constraints.

Security, Trust, and Governance Considerations

An Agentic Intelligence Layer increases the degree of automation in cloud operations by making and executing decisions that affect scaling, routing, placement, and rollouts. While this can improve responsiveness, it also expands the potential impact of incorrect actions or compromised control paths. For this reason, security and trust must be treated as core design requirements for agent-driven cloud management, not as an afterthought[9].

A practical starting point is least-privilege actuation. The agent should not run with broad administrative access. Instead, it should operate through narrowly scoped service accounts and RBAC roles that permit only the actions required for its approved responsibilities, such as bounded scaling or limited routing changes[21]. Kubernetes documentation explicitly recommends minimizing RBAC privileges and using application-specific service accounts rather than broad grants[13].

Second, autonomy must be constrained by policy guardrails that enforce organizational rules before any change is applied. Policy-as-code mechanisms can validate requests at admission time and prevent unsafe configurations or actions from entering the system. This approach fits well with the architecture in Section 3, because the agent can propose actions while admission controls enforce constraints such as maximum scaling steps, forbidden operations, namespace boundaries, and budget-related limits. Open Policy Agent (OPA) and Gatekeeper are widely used for Kubernetes admission control and policy enforcement[22].

Third, changes must be executed in a way that supports verification and rollback. In production operations, safe change management emphasizes reducing risk through incremental rollout and continuous monitoring of service objectives. For agentic systems, this means the agent should validate outcomes after each step using telemetry, and it should revert actions if signals degrade[17].

Finally, trust requires auditability and accountability. Actions should be recorded with what the agent observed, what decision it made, what changes were applied, and what effect followed. This supports post-incident review and compliance, and it helps operators understand and improve the system's behavior over time. Kubernetes security guidance also emphasizes monitoring, logging, and security best practices for cluster environments[22].

Applications and Use Cases

Agent-driven cloud management is most valuable in domains where workloads change rapidly, systems are distributed across cloud and edge, and service-level objectives (SLOs) must be maintained under strict latency and availability constraints[14]. In such environments, human-in-the-loop operations and static automation rules can become slow and difficult to scale, especially when incidents propagate across service dependencies. This section highlights representative application areas where an Agentic Intelligence Layer can provide practical impact by coordinating scaling, routing, placement, and recovery actions through cloud-native mechanisms.

Smart Hospitals and Connected Healthcare

Healthcare systems increasingly rely on cloud-hosted services for patient monitoring, medical imaging workflows, electronic health records, and telemedicine. These workloads are often bursty and time-sensitive, and they require high availability with low-latency response[1]. In a connected healthcare setting, an agentic control layer can help maintain SLOs by forecasting demand spikes (for example, during peak outpatient hours), scaling critical services ahead of time, and redistributing workloads across cloud and edge resources to reduce latency near hospital networks. During service degradations, the layer can support faster recovery by coordinating multi-step mitigation actions such as traffic shifting, selective scaling of bottleneck components, and rollback of unstable releases[3].

Autonomous Manufacturing and Industrial IoT

Modern manufacturing environments use sensor streams, robotics, and cyber-physical control systems that often demand real-time decision-making. These systems typically combine edge processing for low-latency control with cloud processing for fleet-wide analytics and long-term optimization. In such deployments, an Agentic Intelligence Layer can decide where tasks should execute edge versus cloud based on latency, bandwidth, and resource availability. It can also adaptively allocate compute for streaming analytics pipelines, manage failures in edge nodes, and maintain stable operation when workloads shift due to changes in production schedules[5,15].

Smart Cities and Large-Scale Urban IoT

Smart city platforms integrate data from cameras, traffic sensors, public transport systems, and environmental monitoring. The data sources are widely distributed, and traffic patterns can vary sharply by time of day, events, or emergencies. Agent-driven orchestration can help by dynamically placing and scaling microservices across distributed edge nodes to meet latency needs while controlling network costs [5]. When abnormal events occur (for example, sudden traffic congestion or sensor outages), the layer can correlate telemetry across services and apply coordinated mitigation steps, such as rerouting requests, scaling selected analytics services, or shifting computation closer to affected zones[13].

Cloud Cost-Optimization and Resource Efficiency Platforms

Cloud cost control has become a major operational requirement for many organizations. In large deployments, aggressive scaling improves performance but can inflate costs, while conservative scaling may violate SLOs[14]. An Agentic Intelligence Layer can support budget-aware orchestration by balancing cost and performance objectives[16]. For example, it can forecast demand, scale services within pre-defined cost ceilings, and select placement strategies that reduce resource waste. This is particularly useful for multi-tenant environments where resource contention and variable workloads can otherwise lead to inefficiencies.

FUTURE RESEARCH DIRECTIONS

While cloud-native platforms and AIOps methods provide a strong foundation for automation, several challenges remain open for building dependable agent-driven cloud systems across the edge-cloud continuum.

1. First, there is a need for clearer and more standardized interfaces that allow agents to interact safely with cloud-native actuators such as Kubernetes controllers, service-mesh routing policies, and serverless scaling mechanisms. Without consistent abstractions, it is difficult to build agents that are portable across deployments or reliable under changing platform configurations[13].
2. Second, safe autonomy requires stronger operational safeguards. Agentic control expands the range and impact of automated actions, so future work should focus on defining practical governance constraints for actions such as scaling, traffic shifting, and workload placement, and on ensuring that every change can be validated and reversed when outcomes are unfavorable[8]. In addition, decision processes should remain transparent and auditable so operators can review why a particular action was taken under a particular system state.
3. Third, robust operation depends on trustworthy telemetry. Cloud environments often produce noisy, incomplete, or delayed monitoring signals, especially in multi-tenant and edge settings. Future research should improve how agentic controllers handle uncertainty in telemetry, including cross-checking multiple signals, detecting misleading patterns, and avoiding unstable feedback loops that can cause oscillatory behavior[17,18].
4. Finally, evaluation remains a major gap. The community still lacks widely adopted benchmarks for assessing autonomous cloud management in a reproducible way. Shared incident suites, workload traces, and common metrics such as SLO violation duration, recovery time, cost overhead, and stability would make it possible to compare approaches fairly and to demonstrate progress beyond individual case studies[14].

Taken together, these directions point toward the practical steps needed to move from this modern era of advisory and reactive operations toward cloud platforms that can learn, adapt, and maintain service objectives with reduced human intervention.

CONCLUSION

Cloud computing has become the foundation for modern digital services, yet operating cloud platforms at scale is increasingly difficult as deployments move toward microservices, containers, serverless

runtimes, and distributed edge resources. Although AIOps has improved visibility and supports tasks such as anomaly detection, forecasting, and incident triage, many environments still depend on humans for coordinated, multi-step remediation and for decisions that balance performance, reliability, and cost.

This paper presented an architecture-focused perspective on how agentic AI can support the next stage of cloud management. We motivated the need for an Agentic Intelligence Layer that continuously observes telemetry, reasons over service-level objectives and workload conditions, and coordinates actions through cloud-native mechanisms such as scaling, routing, and placement across the edge-cloud continuum. We also outlined representative application domains where such capabilities can be especially valuable, including connected healthcare, industrial IoT, smart cities, and cost-aware cloud operations.

Overall, the proposed reference architecture provides a structured basis for designing more adaptive and self-managed cloud platforms. The future directions discussed in this paper highlight the practical work needed to realize this vision, including safer operational interfaces, stronger governance and trust mechanisms, improved robustness to uncertain telemetry, and reproducible evaluation methods to assess autonomous cloud management under realistic workloads and incidents.

REFERENCES

1. Sunyaev, A. 2024. *Cloud computing.* In *Internet computing: Principles of distributed systems and emerging internet-based technologies* (pp. 165–209). Springer.

2. Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R. and Safina, L. 2017. Microservices: Yesterday, today, and tomorrow. *Present and ulterior software engineering*, pp. 195–216.

3. Gaddam, R.R. 2026. Cloud-native intelligent computing platforms for secure, scalable, and automated infrastructure. *International Journal of AI, Big Data, Computational and Management Studies*, pp. 118–128.

4. Kang, H., Le, M. and Tao, S. 2016. Container and microservice driven design for cloud infrastructure devops. *2016 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 202–211.

5. Abbas, S.I. and Garg, A. 2024. Aiops in devops: Leveraging artificial intelligence for operations and monitoring. *2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, pp. 64–70.

6. Kostopoulos, G., Gkamas, V., Rigou, M. and Kotsiantis, S. 2025. Agentic AI in education: State of the art and future directions. *IEEE Access.*

7. Abou Ali, M., Dornaika, F. and Charafeddine, J. 2025. Agentic ai: A comprehensive survey of architectures, applications, and future directions. *Artificial Intelligence Review,* **59**(1): 11.

8. Acharya, D.B., Kuppan, K. and Divya, B. 2025. Agentic AI: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, **13**: 18912–18936.

9. Banala, S. 2025. Agentic AI in the cloud: A framework for self-directed decision-making and optimization. *2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, pp. 534–541.

10. Thota, M.R. 2025. Autonomous policy-driven cloud platforms: Integrating declarative governance, distributed data systems, and AI-driven control loops for intelligent enterprise modernization. *International Journal of Research and Applied Innovations*, **8**(4): 12642–12657.

11. Ugwueze, V.U. 2024. Cloud native application development: Best practices and challenges. *International Journal of Research Publication and Reviews*, **5**(12): 2399–2412.

12. Aslanpour, M.S., Toosi, A.N., Cicconetti, C., Javadi, B., Sbarski, P., Taibi, D., Assuncao, M., Gill, S.S., Gaire, R. and Dustdar, S. 2021. Serverless edge computing: Vision and challenges. *Proceedings of the 2021 Australasian Computer Science Week Multiconference*, pp. 1-10.

13. Aruna, K. and Gurunathan, P. 2024. Enhancing edge environment scalability: Leveraging kubernetes for container orchestration and optimization. *Concurrency and Computation: Practice and Experience*, **36**(28): e8303.

14. Elhabbash, A., Jumagaliyev, A., Blair, G.S. and Elkhatib, Y. 2019. Sloml: A language for service level objective modelling in multi-cloud applications. *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, pp. 241–250.

15. Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., Wu, Z., Hu, X., Yu, P. and Li, Y. 2025. A survey of aiops in the era of large language models. *ACM Computing Surveys*, **58**(2): 1–35.

16. Chen, Y., Shetty, M., Somashekar, G., Ma, M., Simmhan, Y., Mace, J., Bansal, C., Wang, R. and Rajmohan, S. 2025. Aiopslab: A holistic framework to evaluate ai agents for enabling autonomous clouds. *Proceedings of Machine Learning and Systems*, **7**.

17. Alang, K. 2026. Ai-powered multi-cloud and hybrid cloud strategies. In *Revolutionizing the cloud: Generative AI, security, and sustainability* (pp. 107–125). Springer.

18. Syed, N., Anwar, A., Baig, Z. and Zeadally, S. 2025. Artificial intelligence as a service (AIAAS) for cloud, fog and the edge: State-of-the-art practices. *ACM Computing Surveys*, **57**(8): 1–36.

19. Singh, R., Sukapuram, R. and Chakraborty, S. 2023. A survey of mobility-aware multi-access edge computing: Challenges, use cases and future directions. *Ad Hoc Networks*, **140**: 103044.

20. Prangon, N.F. and Wu, J. 2024. AI and computing horizons: Cloud and edge in the modern era. *Journal of Sensor and Actuator Networks*, **13**(4): 44.

21. Cruz, J.P., Kaji, Y. and Yanai, N. 2018. Rbac-sc: Role-based access control using smart contract. *IEEE Access*, **6**, 12240–12251.

22. Sissodiya, A., Chiquito, E., Bodin, U. and Kristiansson, J. 2025. Formal verification for preventing misconfigured access policies in kubernetes clusters. *IEEE Access.*