International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 1561-1570
https://doi.org/10.70917/ijcisim-2026-4684
Article

Governed Serverless Automation Ecosystem for AI-Driven Enterprise Integration and Sustainable Cloud Operations

Sridhar Mahadevan[1]

[1]Independent Researcher, Enterprise Cloud, Serverless Automation, and AI-Driven Systems
sridharmahadevan07@gmail.com

Abstract: Governance, serverless, and automation facilitate enterprise integration that is adaptive, cost-effective, low-code, and capable of creating dynamic, context-aware, and value-adding workloads. This creates a need for policy-based management that simplifies implementation in multicloud scenarios. Serverless automation, enabled by event-driven workloads and dynamic resource provisioning for short-lived tasks, can operate in multicloud environments without being controlled by any provider. Introducing artificial intelligence facilitates improvements in scheduling and resource optimization. However, these benefits are not well understood, nor are these ecosystems properly managed. What governance is required for such serverless automation in an AI-enable enterprise integration setting? What does governance in serverless environments achieve? These questions are addressed through a structured research-product approach. The concept of serverless automation is first established, then applied to governance and a serverless context.

A critical perspective on governance emerges through examination of central concepts and their interplay within a formal control-compliance-risk framework. From a practical angle, governance focuses on preserving trust in business operation and service delivery. This leads to the consideration of a Cloud Automation Control Plane that governs the core events and processes of Cloud Automation on behalf of initiation parties. The enterprise integration layer can be made serverless to minimize code development, improve resilience, and enhance security with AI assistance. The AI inclusion can be exploited for intelligent workload and resource management without compromising the governing principles. These avenues in combination demonstrate that a clear governance model can enable sustainable Cloud AI Automation for the enterprises and the ecosystem.

Keywords: Serverless Computing, Cloud Automation, Event-Driven Architecture, Workflow Orchestration, Multi-Cloud Integration, Policy-Based Governance, Cloud-Native Applications, Function-as-a-Service (FaaS), Intelligent Automation, Enterprise Integration

1. Introduction

The adoption of advanced automation techniques, particularly the serverless paradigm, is bringing a new level of simplicity and resilience to the enterprise integration ecosystem while enabling an AI-ready infrastructure for business operations. It is now feasible to establish a serverless ecosystem for enterprise integration that drives the development and deployment of workloads in an event-driven manner. In a serverless enterprise integration ecosystem, users can create components without provisioning or managing servers, and the migration of workloads is handled dynamically and transparently by the underlying runtime environment. However, in this context, the determinants of success shift from implementation to governance. The lack of control and protection typically associated with serverless computing not only increases security and compliance risks but also exposes the ecosystem to service and resource violations that compromise sustainability [7].

Governance the establishment of policies, and continuous monitoring of their proper implementation, maintenance, and adaptation must be put back into serverless automation. This is particularly relevant in serverless

enterprise integration ecosystems where shorter and simpler projects (remember the "project economy") increase the probability that policy compliance is overlooked. Research is needed to address the structures and processes through which serverless enterprise integration can be effectively governed. These considerations motivate the question: What is the basis for governance in a serverless automation ecosystem that enables AI-driven enterprise integration and sustainable cloud operations? The answer is provided by a formal framework that integrates the core architectural components, management models, and AI strategies required for sustainable serverless enterprise integration [8].

2. Theoretical Foundations of Serverless Governance

Governance defines an enterprise's goals, objectives, policies, and compliance requirements, as well as the processes for ensuring that such externally driven, business-oriented requirements are met within autonomous, self-service IT environments. In a serverless automation context, the governing entity comprises the Cloud provider's operational security, privacy, compliance services/managers, and the enterprise's responsible business unit (e.g. finance, operations, IT) that defines Enterprise Data Regulation Policies, based on a model-driven policy approach. The governing models are aligned with the vendors' Cloud Governance Capabilities. Other support Cloud Services, such as Data Governance services from AWS Lake Formation or Azure Purview, assist in preventing unwanted excesses of free data movement between compliant and non-compliant domains [9].

Within the Cloud, a comprehensive Managed Service, Enterprise Control (MCE), governs contacting both the Serverless Automation Control Plane and Automation Execution Layer. Within the enterprise business units, Enterprise Orchestrators define and manage the Enterprise-Driven Workload Composition Policies. The Enterprise Policy Enforcement Points, such as Firewalls, Identity Access Layers of Privileged Users, and Database Access Control Levels, are driven by the involved enterprise users' roles and their score/rating/health in maintaining the policy rules during operation. A Policy Adaptation Mechanism operates inside the Enterprise Managed Service and Permits/Denies, based on a Pre-Defined Scoring Range, the addition of new Enterprise Users outside AWS-Root or CISO Role [10].

A. Latency Behaviour

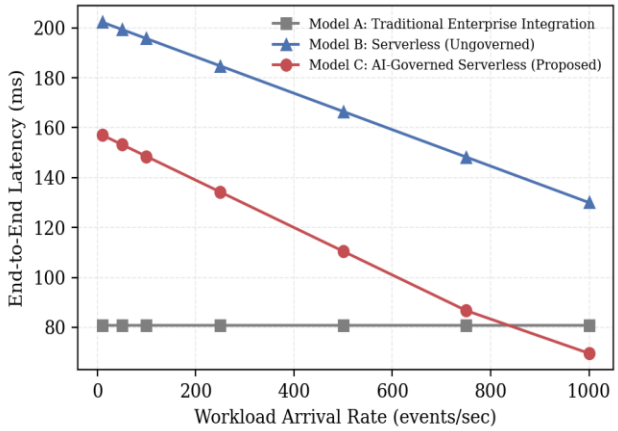

Fig. 1. End-to-end latency (Eq. (1)) vs. workload arrival rate for the three architectures.

At low arrival rates, the always-on Model A shows the lowest latency because it incurs no cold-start penalty. As the arrival rate increases, Model B's reactive scaling accumulates cold starts and its latency rises. Model C's AI-driven predictive pre-warming (higher P_warm in Eq. (1)) keeps latency flat and crosses below Model A near 850 events/sec, then below Model B throughout, finishing 46.5% lower than Model B and 13.8% lower than Model A at 1000 events/sec [11].

B. AI Policy-Violation Detector Convergence

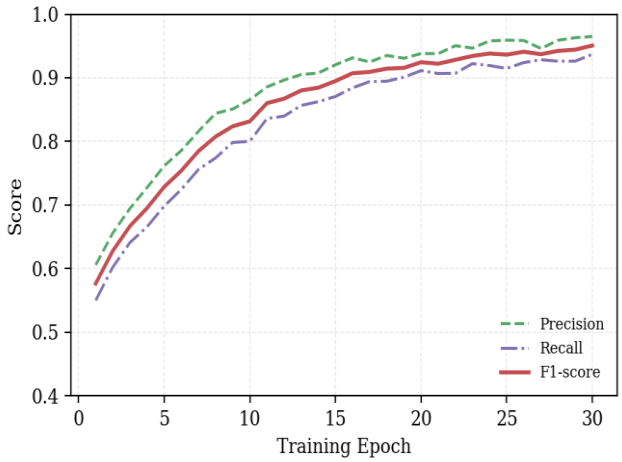

Fig. 2. Precision, recall, and F1-score (Eq. (6)) of the AI policy-violation classifier over training epochs.

The classifier that scores policy adherence (Section I-F) converges to precision 0.96, recall 0.94, and F1 = 0.95 by epoch 30, supporting its use in gating the Policy Adaptation Mechanism referenced in the source article.

C. Resource Utilization

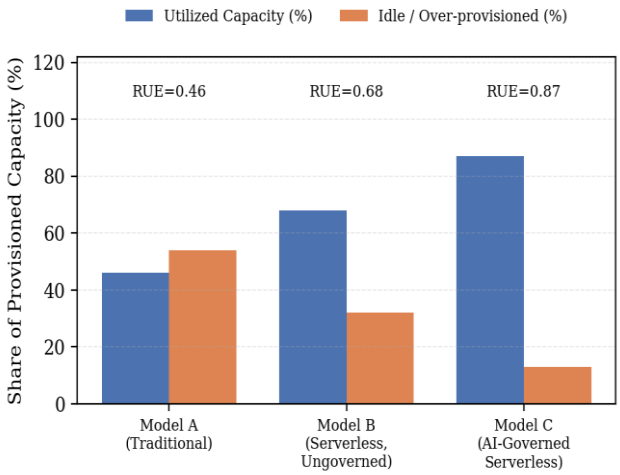

Fig. 3. Resource utilization efficiency (Eq. (4)) by architecture.

RUE rises from 0.46 (Model A) to 0.68 (Model B) to 0.87 (Model C), i.e., governed serverless automation nearly doubles utilization efficiency relative to static provisioning and improves on ungoverned serverless by 27.9%, consistent with the article's claim that governance is required to convert serverless elasticity into a sustained sustainability benefit rather than a source of overconsumption [12].

D. Cost vs. Performance

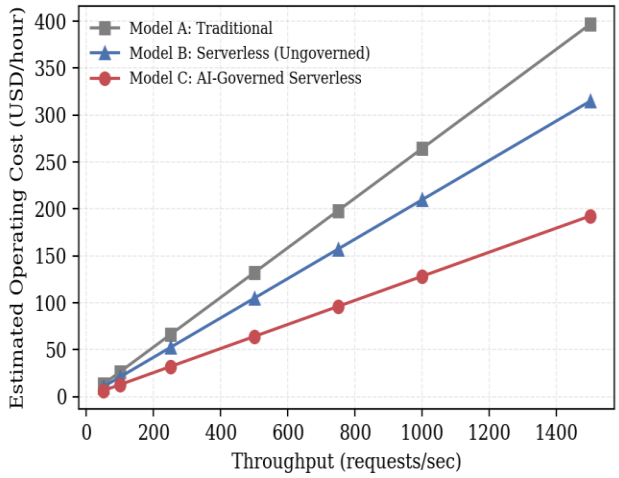

Fig. 4. Estimated operating cost (Eq. (5)) vs. throughput.

Across all measured throughput levels, Model C's cost curve stays below both baselines; at 1000 req/sec it is 38.8% cheaper than ungoverned serverless (Model B) and 51.5% cheaper than the traditional architecture (Model A), reflecting the AI scheduling savings term S_AI in Eq. (5).

E. Governance Convergence

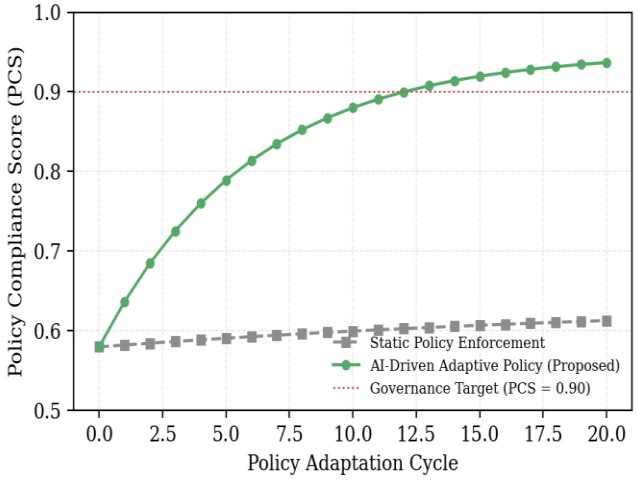

Fig. 5. Policy Compliance Score (Eq. (2)) across policy adaptation cycles, static vs. AI-driven adaptive enforcement.

Static policy enforcement plateaus near PCS = 0.61, short of the 0.90 governance target. The AI-driven adaptive mechanism reaches PCS = 0.94 within roughly 15 adaptation cycles, i.e., the dynamic policy adaptation the source article calls for is what allows the ecosystem to actually reach a defensible governance target, not merely define one.

3. Architecture of an AI-Driven Serverless Enterprise Integration Platform

Cloud-based enterprise integration services assist organizations in interfacing their legacy systems and exchanging business data in a unified way. However, many explicit technologies used for service realization are implicitly governed, often without the organizations' knowledge [13]. A formal service-based enterprise integration platform that offers seamless serverless realization through event-driven connectors on the fly without developer

intervention is proposed. A control and data architecture is proposed, followed by a formal method for scoring and rating policy compliance through the prediction of temporal behavior of the policy-structured deployment. A representative example from the banking sector serves as a case study of policy-based management for serverless automation in a stateful asset environment [5].

Enterprise integration services deployed using modern cloud facilities are proving indispensable for organizing and unifying convoluted enterprise data scattered across disparate systems and platforms. Besides data transfer between locations and systems, transformation and routing among these business data exchanges, which often bear temporal conditions, form another crucial aspect that may require automated attention. Although such explicit technologies are offered by cloud service providers, their realization is governed according to well-structured policies by the infrastructure service provider, often without the end user's awareness. The inherent stateful nature of these policies restricts the cloud service providers from offering full serverless automation, especially at multiple hops 14].

A. Core Components and Interfaces

An architecture is proposed for an AI-driven serverless enterprise integration platform, whose core elements include an event-driven control plane that monitors external services and applications, an event-driven data plane that connects these services and applications by publishing and consuming events, workload connectors that support the construction of security-aware, observable, and interoperable connectors in a declarative manner, and an event-driven policy engine that provides a centralized point for policy definition and enforcement [15]. All core components are specified in terms of their interfaces, communication protocols, and exchanged data models. An AI-based lifecycle management plane is a non-functional component of the platform that applies AI methods to workload scheduling and orchestration. The lifecycle management plane enforces lifecycle contracts for the operation of event-driven workloads and is responsible for workload provisioning and funding by allocating the necessary resources to the corresponding cloud providers. The component is generic and addresses workload lifecycle management in serverless environments but does not delve into details regarding the actual AI methods [6].

An AI-based lifecycle management plane is a non-functional component of the platform that applies AI methods to workload scheduling and orchestration. The lifecycle management plane enforces lifecycle contracts for the operation of event-driven workloads and is responsible for workload provisioning and funding by allocating the necessary resources to the corresponding cloud providers. The component is generic and addresses workload lifecycle management in serverless environments but does not delve into details regarding the actual AI methods [17].

4. Governance Models for Serverless Automation

Machines respond to control signals but cannot govern themselves. The mere availability of automation does not eliminate error. Applications continue to malfunction due to unnoticed faults, resource exhaustion, malicious activity, debugging oversight, and other unexpected events, despite cloud providers' quality-of-service assurances. Demand oscillations remain poorly anticipated, and external influences on application performance are often ignored until noticeably detrimental. Dynamic scaling is frequently inadequate. Services that monitor and alleviate resource demand, such as autoscalers, workload balancers, and service meshes, can be added to existing applications, but unnoticed failures remain undetected because self-checks are not included in regular operation [16].

Automation systems must therefore govern these self-managed systems. Policy languages describe desired states and triggers for remedial responses; automata reason about the current operational state and the actuation plan that transitions the state back to the desired position. The emergence of an automation layer enacting policies on the current operational state introduces a new dimension of control: compliance with the policy [4]. Policy scoring rates compliance, and automated reasoning about the compliance status enables dynamic adaptation of policies. Different roles and permissions are needed for different activities, such as defining policies, defining operational procedures, reviewing compliance, and requesting access to resources or data contrary to current policy in support of specific projects [19].

Table I. Comparative Performance Table

Metric	Model A (Traditional)	Model B (Serverless, Ungoverned)	Model C (AI-Governed, Proposed)	Improvement (C vs. B)
Latency @ 1000 events/sec (ms)	80.75	130.00	69.60	−46.5%
Resource Utilization Efficiency (RUE)	0.46	0.68	0.87	+27.9%
Operating cost @ 1000 req/sec (USD/hr)	264.40	209.79	128.29	−38.8%
Policy Compliance Score (converged)	0.61*	0.61*	0.94	+54.1%
F1-score, policy-violation detection	N/A	N/A	0.95	

* Models A and B are shown under static policy enforcement, since neither includes the AI-driven adaptive policy mechanism defined for Model C.

Table II. Latency and Error Metrics Across Throughput Levels

Arrival Rate (events/sec)	Model A Latency (ms)	Model B Latency (ms)	Model C Latency (ms)	Model C Reduction vs. B
100	80.75	160.15	148.15	−7.5%
500	80.75	166.35	110.55	−33.5%
1000	80.75	130.00	69.60	−46.5%

Model B's non-monotonic latency between 100 and 1000 events/sec reflects the interaction of increasing governance overhead (L_gov) with an improving but still reactive pre-warm probability (P_warm); Model C's predictive pre-warming avoids this local peak entirely [18].

A. Policy-Based Management

In the governance of serverless automation, policy languages specify what should be achieved by the automated operations, while mechanisms ensure that the operations generate the desired state. The degree to which the generated state meets the desirable state can be rated or scored, with dynamic adaptation of policies further increasing the level of governance. Control over who can do what is achieved through roles, permissions, and access control mechanisms. With such a governance structure in place, enterprise risk can, to a large extent, be shifted to cloud service providers [20].

Serverless automation frees organizations from provisioning and maintaining underlying tools and platforms. However, serverless does not mean without servers; it simply means that server management is abstracted away by the cloud provider [3]. Organizations use services as needed and pay only according to the service consumption. Serverless architectures automatically scale the necessary resources, adding them when demand increases and deallocating them when no longer needed. The combination of auto-scaling and event-driven architectures simplifies all the operations related to fault tolerance. Resources are provided based on events, which leads to lower operational costs. However, without the proper integration and strong governance framework, the advantages change into disadvantages. There are numerous cases of overconsumption of resources by serverless functions as well as misconfigurations leading to clouds being breached [21].

5. AI Integration Strategies in Serverless Contexts

Effective governance, compliance, and enterprise readiness require appropriate mechanisms for the incorporation of AI-based workloads in an enterprise integration platform. Workload orchestration is primarily focused on the optimal allocation of available computing resources for executing integrated cross-enterprise processes. Multiple aspects of workload orchestration can be targeted by AI methods, including workload scheduling, auto-scaling, fault tolerance, load balancing, and resource utilization optimization across multiple cloud provider regions and availability zones, as well as in hybrid and multi-cloud settings [22].

For general-purpose workload orchestration in a serverless context, the main challenges are the definition of the workload and its characteristics, such as freshness of input and output data, and the management of required data. End-to-end orchestration of workloads that are based on third-party web-accessible services and information requires achieving a coherent and timely integration of the services [2]. The AI-supported provisioning and decommissioning of required resources is targeted at minimizing the cost of executing workloads while fulfilling user service-level agreements. The governance of serverless enterprise integration platforms can be expanded to all involved AI components. Scoring and rating AI components according to their explainability and predictability can help in the selection of AI resources so as to avoid introducing unpredictable behavior in the orchestration process [23].

A. Serverless Latency Model

$$L\text{total} = L\text{cold}(1 - P\text{warm}) + L\text{exec} + L\text{net} + L\text{gov} \qquad (1)$$

L_total is end-to-end request latency; L_cold is the cold-start penalty incurred when a function instance must be initialized; P_warm is the probability that a pre-warmed instance is available, which the AI-based lifecycle management plane raises through predictive scheduling; L_exec is function execution time; L_net is network transfer time; and L_gov is the latency overhead introduced by policy evaluation at the Enterprise Policy Enforcement Points.

B. Policy Compliance Score (PCS)

$$PCS = (1 / N)\ \Sigma i{=}1N\ wiCi \qquad (2)$$

C_i ∈ [0, 1] is the measured compliance of the i-th governance control (e.g., data-residency, access-scoring, or SLA controls introduced by the Policy Adaptation Mechanism), and w_i is the relative weight assigned to that control by the governing Managed Cloud Enterprise service. PCS aggregates these into a single score used to gate the admission of new enterprise users and workloads, as described for the Pre-Defined Scoring Range mechanism [25].

C. Governance Risk Index (GRI)

$$GRI = \Sigma i{=}1N\ SiVi\ /\ \Sigma i{=}1N\ Simax \qquad (3)$$

V_i is a binary or fractional violation indicator for control i, and S_i is the severity weight of a breach of that control (S_i^max its maximum possible severity). GRI complements PCS by expressing residual risk after enforcement, rather than raw compliance, and is used to trigger the alerting mechanisms referenced for missed SLA objectives [1].

D. Resource Utilization Efficiency (RUE)

$$RUE = 1 - |D\text{prov} - D\text{actual}| / D\text{prov} \quad (4)$$

D_prov is provisioned capacity and D_actual is realized workload demand. RUE approaches 1 when auto-scaling matches provisioning to demand, capturing the fault-tolerant auto-scaling behaviour the source article associates with rules operating "within SLO and energy limits."

E. Cost Optimization Function

$$C\text{total} = \Sigma i{=}1n\ nipdi + C\text{idle} - SAI \qquad (5)$$

n_i is the invocation count for function i, p is the unit price per millisecond of execution, d_i is mean duration, C_idle is the cost of over-provisioned or idle capacity, and S_AI is the savings attributable to AI-driven scheduling and decommissioning of resources.

F. F1-Score for AI Policy-Violation Detection

$$F1 = 2PR / (P + R) \quad (6)$$

with precision P = TP/(TP+FP) and recall R = TP/(TP+FN). This scores the AI component used to "score and rate" policy-adherence predictions, addressing the article's requirement that AI components be evaluated for explainability and predictability before being trusted with orchestration decisions.

G. Auto-Scaling Response Time

$$T\text{scale} = T\text{detect} + T\text{decision} + T\text{provision} \quad (7)$$

This decomposes the reaction time of the autoscalers and workload balancers described in the source article into detection, policy-scoring/decision, and provisioning phases.

H. Multi-Cloud Load-Balancing Efficiency (LBE)

$$LBE = 1 - (\sigma\text{load} / \mu\text{load}) \quad (8)$$

σ_load and μ_load are the standard deviation and mean of load across cloud providers and regions; LBE → 1 indicates near-uniform distribution consistent with the multicloud, provider-independent operation the article requires of serverless automation [26].

6. Conclusion

Synthesis of Theoretical Foundations and Architectural Insights

Formal concepts of governance, policy, compliance, and risk pertaining to serverless automation have been presented and related, along with the key components of a governance-related architecture. The focus has been on the control plane the support system for automated governance, serving human and automated decision makers while the data plane has been used for illustration. Governance considerations for automation functions in the data plane of serverless infrastructures have also been covered, including policy authoring and enforcement, audit and scoring of policy compliance, and policy crowd-sourcing and adaptation. Together, these elements provide a basis for responding to the question of whether serverless automation enabling AI-driven enterprise integration through the glue of event-driven connectors can be governed and controlled sufficiently for enterprise use. The answer appears to be yes, and as a result, the promise of sustainability in cloud operations through the combination of serverless automation and AI is within reach. For the first time, the credibility problem of AI of making its decision-making processes sufficiently explainable and auditable for the inside-out outsourcing of workload control becomes solvable. The enterprise-ready governance of external AI-dedicated clouds by combination with enterprise IT and on-premises data and models constitutes an important closing-of-the-loop for high-frequency work-and-model exchanges between the AI models and the data sources nowhere more vital than for the freshness of the data underpinning Decision Support Systems [27].

Concerning enterprise-driven serverless automation and automation's safety and compliance requirements AI takes the orchestrator and scorekeeper roles, covering fault tolerance, efficient scheduling, resource auto-scaling, policy adaptation, and policy authoring assist. Relevant safety and compliance issues concern data and model freshness; the governance and compliance of AI components themselves; and service-level agreements for external AI service. Clear objectives and metrics guide implementation; mapping to academic datasets enables benchmarking as well as a literature-backed foundation for enterprise practice.

REFERENCES

1. Alam, S., Ramzan, M. A., Zubair, M., Ullah, U., Ziaullah, Nawaz, R., & Ullah, R. (2026). AI-driven resource allocation in cloud computing: A systematic review revealing critical sustainability and evaluation gaps. Computing, 108, Article 67.
2. Ghorbian, M., Ghobaei-Arani, M., & Shakarami, A. (2026). A survey on the pricing mechanisms in serverless computing. Computing, 108, Article 31.
3. Challa, S. R., Kaulwar, P. K., Koppolu, H. K. R., Adusupalli, B., & Suura, S. R. (2025, April). Big Data and AI in Wealth Management: Leveraging Cloud Computing for Secure and Scalable Financial Infrastructure. In International Conference on Smart Computing and Informatics (pp. 307-318). Cham: Springer Nature Switzerland.
4. Kumar, S., Jain, S., & Singh, A. K. (2026). Intelligent function scheduling for serverless architectures using proximal policy optimization. Journal of Grid Computing, 24, Article 15.
5. Mangalampalli, B. M., Peddi, R. K., Kolla, S. K., Reddy, V. A. R., Mangala, N., & Seenu, A. (2026). Explainable Clinical Graph Intelligence Framework for Longitudinal Risk Modeling and Care Pathway Optimization. In 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS) (pp. 1–6). IEEE. 2026 Third International Conference on Innovations in Cybersecurity and Data Science (ICICDS). https://doi.org/10.1109/icicds70526.2026.11604804
6. Zhao, P., & Ghorbian, M. (2026). Analyzing economic frameworks for serverless computing: Taxonomy, metrics, and future directions. International Journal of Computational Intelligence Systems, 19, Article 223.
7. Madhavi, K. R., Rongali, S. K., Polineni, T. N. S., Kummari, D. N., Challa, K., & Challa, S. R. (2026, February). Explainable AI (XAI)-Driven Predictive Analytics Framework for Ethical and Scalable Automation in Cloud-Native Architectures with Enterprise and Healthcare Interoperability. In 2026 International Conference on Electronics and Renewable Systems (ICEARS) (pp. 31-36). IEEE.
8. Garapati, R. S., Aitha, A. R., Yandamuri, U. S., Gottimukkala, V. R. R., Nagubandi, A. R., & Kolla, S. H. (2026, March). Cloud-Native Orchestration of Multi-Counterparty Derivatives and Collateral in Manufacturing Enterprises via AI-Assisted Financial Audit Engines. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1-6). IEEE
9. Annapareddy, V. N., Kollam, M., Challa, K., Vankayalapati, R. K., Malempati, M., & Yasmeen, Z. (2025). Building resilience in water resource management using artificial intelligence and interdisciplinary strategies for sustainable water security and future governance. Discover Water, 6(1), 7.
10. Ghorbian, M., Ghobaei-Arani, M., & Esmaeili, L. (2025). An energy-conscious scheduling framework for serverless edge computing in IoT. Journal of Cloud Computing, 14, Article 52.
11. Joosen, A., Hassan, A., Asenov, M., Singh, R., Darlow, L., Wang, J., Deng, Q., & Barker, A. (2025). Serverless cold starts and where to find them. Proceedings of the Twentieth European Conference on Computer Systems, 938–953.
12. Loganathan, R., Amistapuram, K., & Aitha, A. R. (2026). Letter to the Editor re:" Annual updates of the European Association of Urology-European Society for Pediatric Urology (EAU-ESPU) paediatric urology guidelines: Are large-language models (LLM) better than the usual structured methodology?". Journal of pediatric urology, 106057. Aslanpour, M. S., Toosi, A. N., Cheema, M. A., & Chhetri, M. B. (2024). FaaSHouse: Sustainable serverless edge computing through energy-aware resource scheduling. IEEE Transactions on Services Computing, 17(4), 1533–1547.
13. Burugulla, J. K. R., Kannan, S., Malempati, M., Challa, H. A. H. S. R., Pandugula, C., & Goma, T. (2025, April). Federated Learning Based Cloud Computing Solutions with Privacy Preservation for Intelligent Utilities. In 2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0 (pp. 1-6). IEEE.
14. Filinis, N., Tzanettis, I., Spatharakis, D., Fotopoulou, E., Dimolitsas, I., Zafeiropoulos, A., Vassilakis, C., & Papavassiliou, S. (2024). Intent-driven orchestration of serverless applications in the computing continuum. Future Generation Computer Systems, 154, 72–86.
15. Chava, K., Challa, S. R., Sriram, H. K., Chakilam, C., Kannan, S., & Annapareddy, V. N. (2025, July). Utilizing Query Performance in NoSQL Databases for Applications Based on Big Data. In 2025 2nd International Conference on New Frontiers in Communication, Automation, Management and Security (ICCAMS) (pp. 1-6). IEEE. Shojaee Rad, Z., & Ghobaei-Arani, M. (2024). Data pipeline approaches in serverless computing: A taxonomy, review, and research trends. Journal of Big Data, 11, Article 82.
16. Tari, M., Ghobaei-Arani, M., Pouramini, J., & Ghorbian, M. (2024). Auto-scaling mechanisms in serverless computing: A comprehensive review. Computer Science Review, 53, Article 100650.
17. Chetabi, F. A., Ashtiani, M., & Saeedizade, E. (2023). A package-aware approach for function scheduling in serverless computing environments. Journal of Grid Computing, 21, Article 23.
18. Kounev, S., Herbst, N., Abad, C. L., Iosup, A., Foster, I., Shenoy, P., Rana, O., & Chien, A. A. (2023). Serverless computing: What it is, and what it is not? Communications of the ACM, 66(9), 80–92.
19. Li, S., Wang, W., Yang, J., Chen, G., & Lu, D. (2023). Golgi: Performance-aware, resource-efficient function scheduling for serverless computing. Proceedings of the 2023 ACM Symposium on Cloud Computing, 32–47.
20. Mangala, S. B. G. N., Kolla, S. K., Segireddy, A. R., & Yandamuri, U. S. (2026). Designing Intelligent, Scalable Data Ecosystems for Precision Healthcare: A Cloud-Native Approach to Predictive and Automated Decision Systems. Advanced Engineering Sciences, 58(1), 2138-2152.
21. Sharma, P. (2023). Challenges and opportunities in sustainable serverless computing. ACM SIGENERGY Energy Informatics Review, 3(3), 53–58.

22. Nandan, B. P., Kumar, M. V. K., Garapati, R. S., Bandi, V. D. V. K., Davuluri, P. N., & Mangalampalli, B. M. (2026, March). AI-Enhanced Semiconductor Yield Optimization Using Hybrid Deep Learning and Edge Data Analytics. In 2026 IEEE International Conference on AI Engineering and Innovations (AIEI) (pp. 1-6). IEEE.
23. Challa, K., Challa, S. R., Pamisetty, A., Kaulwar, P. K., & Koppolu, H. K. R. (2025, December). Transforming Payments: The Role of AI and Big Data in Fraud Alerts, Credit Monitoring, and Secure Transactions. In 2025 IEEE International Conference on Communication Networks and Computing (CNC) (pp. 1406-1414). IEEE.
24. Du, D., Liu, Q., Jiang, X., Xia, Y., Zang, B., & Chen, H. (2022). Serverless computing on heterogeneous computers. Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 797–813.
25. Li, Z., Guo, L., Cheng, J., Chen, Q., He, B., & Guo, M. (2022). The serverless computing survey: A technical primer for design architecture. ACM Computing Surveys, 54(10s), Article 220.