International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

OBSERVABILITY AND AUDITABILITY IN SAAS MONETIZATION SYSTEMS

Siddhartha Kantipudi

Northwest Missouri State University

Abstract

The development of Software-as-a-Service (SaaS) platforms has, thereby, created more complicated monetization processes, especially in multi-tenant, compliance-subject interactions. The need to have observability and auditability, in ensuring financial transparency, enabling regulatory compliance, and resilience in these systems, has emerged as a critical element in these systems. This overview examines the technical foundations, technology, and governance that is behind observability and auditability in the monetization of SaaS. The most important themes are about the use of metrics, logs, and traces to provide end-to-end visibility of systems in real-time, the use of policy-as-code to enforce monetization, and the application of tamper-evident audit trails to assist system compliance and dispute resolution. Other new challenges that can be noted in the review include policy drift, high-cardinality telemetry, and changing regulatory environments. With observability and auditability of the monetization stack, SaaS now has increased support for transparent pricing, secure entitlements, and revenue models that are accountable. The future trends point to expanded significance of AI, explainable telemetry, and the unification of compliance models in this field.

Keywords: SaaS Monetization, Observability, Auditability, Compliance, Multi-Tenant Systems, Billing Workflows, Policy-as-Code, Revenue Assurance, Cloud Governance, Instrumentation, Distributed Tracing, Regulatory Compliance.

1. Introduction

The blistering development of the Software-as-a-Service (SaaS) business models has considerably changed the process of providing digital products, charging for them, and controlling them. Along with the more frequent utilization of automated invoicing engines, dynamic pricing systems, as well as multi-tenant architecture, the monetization layer of the SaaS platforms is currently a vital operational and compliance point of emphasis. Observability and auditability have been adopted in this changing world to act as a keystone in providing system health and integrity, accurate financial reporting, compliance across regulations, and reliable user trust.

The term observability in SaaS monetization describes the ability to know in real-time what the billing systems are doing, including how they are used, how well they are meeting a set of metrics, traces, and policy decisions merely as inputs. Observability, in contrast to traditional monitoring, has traditionally concentrated on the availability of systems and adds support to instrumenting the billing pipelines to monitor behavioral anomalies, bursts of usage, regulatory breaches, and rogue use of monetized capabilities [1]. On the other

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

hand, this is called auditability because we are able to track and review all actions, events, and decisions for the efficacy of billing and entitlements. It reinforces internal control, the enforcement of contracts, and defense in the law courts. With the increasing tightening of global data privacy and financial regulations (i.e., GDPR, SOC 2, HIPAA, and IFRS 15), tamper-resistant, yet easy-to-use audit logs are no longer a nice-to-have in monetization infrastructure [2].

When we bundle the concept of a monetization system with SaaS, a substantial universe of events is handled by monetization systems: user logins, API requests, resource provisioning, usage limits, feature toggles, authorization constraints, etc. Every incident can cause billing activities, alteration of plans, or access. To ensure such events can be visible, traced, and verified, there is a need to integrate observability and auditability in the platform architecture, and not as an afterthought as a secondary on logging. Additional layers of complexity associated with the intersection of multi-tenancy and region-aware monetization lead to complexity. Tenants can work at different rates or pricing bands, data localisation policies, contract agreements, or tax codes. Incorrectly assigning/applying usage, or applying billing policies of inconsistent definition, can result in loss of revenue, customer conflicts, or even government fines. Systems in use in observability should hence grant fine-grained telemetry (e.g., down to the tenant, feature, and geographic level), but audit trails should show correctness of policy logic over time [3].

Modern SaaS systems are beginning to use distributed tracing, policy as code enforcement, automatic compliance verification, and non-mutating audit pipelines, and are incorporating them with systems like OpenTelemetry, Jaeger, Prometheus, and commercial tools such as Datadog, Splunk, and New Relic as a means of managing this complexity. Such integrations enable the root-cause analysis, system forensics, and proactive alerts on anomalies in monetization pipelines [4]. The paper focuses on the structure and application of the concept of observability and auditability in SaaS monetization systems and their regulatory importance. It details the technologies, protocols, and governance models that will allow secure and transparent, and legally sound billing functions in multi-tenant SaaS systems. Amongst other things studied are the principles of observability, auditing systems, instrumentation strategies, traceability of policies, integration of regulations, and outlooks in the future.

Foundations of Observability in SaaS Monetization

The aspect of observability of the SaaS monetization systems is beyond the usual application performance monitoring: It observes the internal rationalization and the external manifestations of the financial operations, usage tracking, entitlement enforcement, and policy decision-making. In monetization architectures, observability should ensure that all interactions of services concerning the price, feature access, or resource consumption are traceable, visualizable, and interpretable in almost real-time. Multi-data Observability is founded on three main sources of data: metrics, logs, and traces. All these pillars provide access to another layer of insight into the functioning of monetization processes, their abuse, or possible violation.

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2.1. Metrics in Monetization

The measurements are known as metrics, quantitative time-series data carrying summary information on the state of systems and their operational health. In a monetization engine, the metrics may focus on how many billable API requests have been issued, how many times a feature has been used per tenant, the quota limits that have been hit, and pricing computation errors or latency when generating an invoice [5].

All these metrics are usually gathered through services such as Prometheus or other cloud-native services like Amazon CloudWatch or Azure Monitor. They are used to inform how to define Service Level Indicators (SLIs) and Service Level Objectives (SLOs) particularly when it comes to monetization performance indicators, like as an example of invoice issuance in 5 seconds of billing cycle completion or real-time sync of the usage events across subsystems of the billing system [6].

2.2. Logging and Monetization Events

SaaS monetization systems should have logs that verify not only infrastructure diagnostics but also billing-related information, policy calculations, and user-triggered events that affect financial and business decisions. A well-designed logging pipeline logs supporting information about these events, such as user identity, tenant metadata, location, type of resource, and the policy outcome. As an example, observer systems should track not only the change request but also the policy that applies to the change, authorization of the user who requested the change, pre- and post-change entitlement, and any change that needs billing when a tenant user transitions from a basic plan to a premium plan. These logs should be organized into hierarchical order, time-stamped, read only and cross-correlated/distributed systems to facilitate traceability and compliance [7].

2.3. Tracing Across Distributed Monetization Components

Distributed tracing is mandatory in the monetization process, specifically when the announcement of billing decisions is done in a compartmentalized style along various microservices or outside systems. It allows for following an individual monetization transaction in the context of the original usage to subsequent policy ascertaining, metering, cost calculation, and invoice creation throughout a fully distributed system.

As an example, a single billing trace may start with a user invoking a cloud function, then go to a metering engine and call a policy evaluation service (e.g., Open Policy Agent), initiate entitlement validation, and finally end up as a billing line item. This journey can then be visualized using tools like Jaeger, OpenTelemetry, or Zipkin in order for the platform engineer to identify the bottlenecks, failures, or inconsistencies [8].

2.4. High Cardinality and Tenant-Scoped Observability

Monetization telemetry should be able to track high cardinality-by-tenant, feature, geography, or compliance-level usage, and policy results. It is not only needed to correctly bill, but it is also required to isolate problems, maximize pricing strategy, and prove compliance during an audit. Lacking the ability to scope observations to tenants, the

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

misuse of resources, cross-tenant data leakage, or a violation of the policy can remain hidden until they cause problems of a financial and legal nature [9]. Observability systems that are multi-tenant have a trade-off between granularity and scale. Adding tenant identifiers, categories of service, and the context of usage (e.g., during an alert) via tagging telemetry is necessary to facilitate dashboards, alert limits, and forensics analysis. But granularity may be too fine-grained and ruin storage and compute resources, particularly at scale.

2.5. Alerting and Monetization Health Signals

Effective observability pipelines also support real-time alerting for anomalies in monetization behavior. Alerts can be triggered by:

- Unexpected billing volume surges.
- Policy evaluation errors.
- Declined payment rates.
- Discrepancies between metered usage and expected entitlements.
- High rates of failed plan upgrades or downgrades.

These alerts allow proactive mitigation of customer-impacting issues, such as double billing or unintended access revocations, which can damage trust and increase support overhead.

2.6. Observability and Policy-as-Code

The new SaaS systems bring in observability on the policy evaluation. Policy-as-code monetization (e.g., OPA, HashiCorp Sentinel) allows logging the made policy decisions, the parameters passed to it as an input, and when policy evaluation was actually made. It applies specifically to the case of the compliance-sensitive monetization in which adherence to regulations may necessitate the due consideration of regional restrictions or the classification of data, or services eligibility [10].

During observability pipelines, policies become visible, and this fact makes it possible to debug in case of an incorrectly applied pricing rule or feature access policies, or tune policies in terms of performance and fairness.

2.7. Integration with CI/CD and Feature Flags

The deployment lifecycle of monetization features should also use observability. Observability systems need to monitor the effects of these changes when new pricing models, feature gates, or discount logic are being added using CI/CD pipelines. The feature flags that are implemented to make monetized features available or off-limits have to be visible, as one has to find out: Which were accessed by whom, at what time, and which of the configurations. In brief, observability in SaaS monetization systems is a multidimensional discipline that implies gathering, correlating, and displaying different types of data metrics, logs, traces, and policy-checking. It helps in identifying anomalies, internalizing entitlements, maximizing revenue, and making all financial transactions

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

made within the system legally defensible.

2. Role of Auditability in Compliance and Risk Management

The principle of auditability in SaaS monetization systems implies that all financial activities, entitlement choices, and policy-related actions may be tracked, validated, and justified. It is essential to use platforms whose regulations are severe, like GDPR, SOC 2, PCI-DSS, and HIPAA, to provide a clear processing of user data and finances [11].

Unlike observability, which focuses on system state and performance, auditability emphasizes immutability, traceability, and governance. This involves generating detailed event logs for:

- User-initiated actions (e.g., subscription changes)
- Policy execution (e.g., geo-restricted pricing enforcement)
- Billing system outputs (e.g., invoice generation, usage reports)

Such logs are usually maintained in append-only or blockchain-based tamper-evident systems and indexed under the tenant and timestamp, and action type to facilitate forensics analysis and regulatory review [12, 13]. Authorization decisions, such as who authorised access to monetized features and under what terms and within a given time, must also be captured in the audit trails. This is highly essential in enterprise SaaS as the usage right is limited by negotiated agreements or entitlements that are dependent on compliance [13, 14].

Rules tools improve auditability through the enforcement of pricing model versions, entitlement rules, and discount configurations. People become fully responsible as any adjustment in the monetization policies is recorded, reviewed, and associated with stakeholders. Regular audit simulations or compliance drills are increasingly used to validate that audit logs align with legal and operational expectations. Platforms also expose these audit logs to customers through billing reports or admin dashboards to build trust and transparency. In summary, auditability forms the compliance backbone of monetization systems, enabling risk mitigation, contract enforcement, and regulatory conformance through persistent, structured, and verifiable event records.

Received: August 06, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

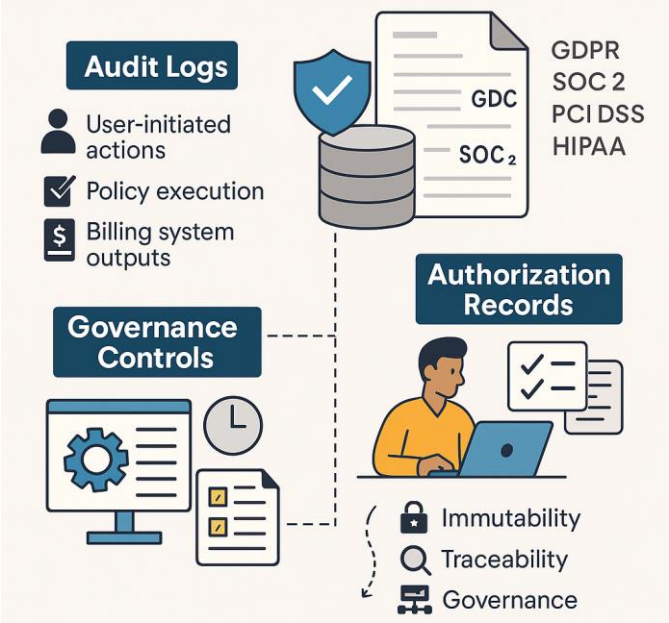

Figure 1: Auditability Architecture in SaaS Monetization Systems

This diagram illustrates the integration of audit logs, authorization records, and governance controls to ensure compliance with standards such as GDPR, SOC 2, PCI DSS, and HIPAA. Key components include traceable user actions, policy executions, billing outputs, and immutable governance records.

3. Instrumentation and Data Collection Techniques

Observability and audit-readiness in terms of SaaS monetization rely on the process of instrumentation, whereby receivers of tracking points within systems can be made so that metrics can be gathered with meaning. In instrumentation, in the area of billing and entitlement engines, usage events, the lines of decision, and operations involved in the revenues can be observed in real time.

Modern monetization platforms utilize automated telemetry pipelines, collecting structured data through:

● **API gateways:** Logging each monetized API call with tenant identifiers, timestamps, and rate limits.

● **Service meshes:** Tracking inter-service communication relevant to metering and policy decisions.

● **Custom SDKs:** Injected into application components to record user actions triggering billing changes [14, 15, 16].

Prometheus, Fluent Bit, or OpenTelemetry data collectors perform the process of streaming logs, metrics, and traces to an aggregation system and are then pointed at an

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

aggregation destination, which could be Prometheus, Elastic Stack, or other cloud-native services like Azure Monitor or AWS CloudWatch.

The instrumentation points cover the policy inputs (e.g., user plan, region) and decision (e.g., allow/deny), and the latency of processing to facilitate evaluation of policy and attempt to trace entitlement. They are crucial in debugging problematic pricing logic as well as ensuring legal conformity in multi-region settings [16-18]. High cardinality is also a necessary instrumentation to be able to split the data by tenant, location, or service. To avoid performance overheads, the platforms introduce sampling, rotation of logs, and real-time data compression. In conclusion, it can be said that the smart instrumentation by which it is implemented to deliver SaaS resources is able to provide a smart view of fine-grained and actionable data surrounding monetization behavior that is needed in the face of transparency, e.g., or troubleshooting and anytime-on compliance assurances.

4. Policy Enforcement and Traceability in Billing Workflows

As far as compliance-oriented SaaS installations are concerned, monetization is shifting towards policy-oriented use rights access, headcount price, and quotas per use. These policies can be developed as rules that can be read by a computer (e.g., formatted in Open Policy Agent) and determine when and how to charge a tenant for some action or service.

Application of the policies allows the bills to only be activated by the authorized usage, as well as the proper consumption of the entitlements. To give an example, a policy can render some premium feature unavailable to jurisdictions that restrict the export of data or impose region-specific tax policies when creating invoices [19, 20]. Platforms provide policy traceability within the billing pipelines with an aim to ensure transparency and accountability. A report of the policy applied and conditions considered in it, and the result of the same, should be provided in every action of billing so that they can be analyzed in case of audits or where a dispute arises [21]. Also, it should be significant that policies must be versioned. Additional terms to monetization may be required in case of alterations in the tax law or contract terms. One can trace the history of the versions of the policies, the date of the decision, and decisions made related to it, and this helps in tracing the adherence to the law and correctness in operation [22]. Lastly, the platforms lay a combination of policy decision logs and observability tools which can be visually followed to easily understand why a specific monetization action occurred in real-time, allowing faster debugging as well as creating more confidence among the stakeholders.

6. Integration with Regulatory Frameworks and Standards

A wide range of global sector regulatory and regulation policies around the world is able to reign over SaaS monetization. It is through observability of monetization as well as audibility of monetization that the operations can be well aligned and checked against these laws.

Key regulations influencing monetization include:

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

● **GDPR:** Requires clear data lineage and explicit consent for billing-related data usage.

● **SOC 2:** Mandates audit trails and system monitoring to ensure financial control and security.

● **HIPAA:** Demands strict logging of data access and feature provisioning in healthcare SaaS platforms.

● **IFRS 15 / ASC 606:** Defines how and when revenue should be recognized, which impacts billing transparency and auditability [13].

To comply, monetization systems must implement:

● Data access controls and role-based permissions for billing data.

●Retention policies aligned with regional mandates (e.g., Europe vs. U.S.).

● Audit reporting formats compatible with external reviewers and internal compliance teams.

● Real-time policy enforcement that dynamically adapts to jurisdictional rules.

The governance as a code approach is even being adopted into the platforms, where the compliance rules are now made into templates, which auto-enforce all rules across all services [23, 24]. Since such regulatory checks can be directly integrated into the observability and auditability pipelines of the billing processes, not only can we ensure that we are in compliance with the law, but also make ourselves virtually ready for third-party audits, as well as build and gain customer confidence in the process.

7. Challenges, Best Practices, and Future Directions

With all the tooling and even architectural patterns aimed at supporting the delivery of robust observability and auditability of SaaS monetization systems, there exist certain recurrent issues related to the overall pursuit of implementing such features:

● **Data Volume and Granularity:** Multi-tenant environments generate vast amounts of fine-grained telemetry. Managing high cardinality metrics and event logs without overwhelming storage and compute resources remains difficult.

● **Cross-Tenant Isolation:** Ensuring that telemetry and audit data are correctly partitioned across tenants is critical to avoid data leakage and compliance violations.

● **Policy Drift:** In distributed systems, policies may evolve, leading to inconsistencies between declared and enforced rules if not version-controlled and monitored.

● **Interoperability:** Integrating observability pipelines with diverse billing, CRM, and compliance systems often requires custom connectors or middleware.

To address these issues, several best practices have emerged:

● Implement structured logging and standardized tracing identifiers across billing and policy services.

● Use immutable storage for audit trails with cryptographic verification (e.g., hash chains).

Received: August 06, 2025

International Journal of Applied Mathematics
Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

● Adopt observability-as-code practices for consistent instrumentation and telemetry tagging.

● Employ real-time dashboards and anomaly detection to surface monetization anomalies early.

Looking ahead, future directions include:

● AI-driven anomaly detection to flag revenue leakage, pricing misconfigurations, or policy violations.

● Explainable observability, where systems can generate human-readable explanations for monetization decisions or billing outcomes.

● Federated compliance observability, enabling global SaaS platforms to enforce region-specific regulations while maintaining centralized visibility.

● Open standards for policy auditing, trace interoperability, and billing telemetry formats.

To sum up, observability and auditability are not secondary features but form the pillar of the reliability, transparency, and compliance of SaaS monetization systems. The investments in these areas will make the platforms better placed to withstand regulatory complexity, win the trust of the users, and operate smoothly in the financial sector.

Table 1: Strategic Enhancements for Observability and Auditability in SaaS Monetization

Dimension	Enhancement Focus	Strategic Impact
Telemetry Governance	Tenant-aware tagging, high-cardinality segmentation	Enhances granularity while maintaining traceability and data partitioning
Audit Integrity	Cryptographically verifiable storage (e.g., append-only logs)	Strengthens legal defensibility and audit-readiness
Policy Monitoring	Real-time policy validation feedback loops	Reduces errors from policy drift; ensures alignment with contractual expectations
Toolchain Alignment	API-standard connectors for CRM, ERP, and compliance suites	Boosts system interoperability and reduces integration latency
Anomaly Detection	Behavioral baselines, outlier detection via AI	Detects billing fraud, unusual access patterns, or entitlement misalignments
User Transparency	Contextual telemetry reporting and justification logs	Builds customer trust and improves support response quality

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Dimension	Enhancement Focus	Strategic Impact
Standardization	Adoption of open telemetry and policy audit specifications	Facilitates cross-platform portability and regulator interoperability
Global Compliance View	Federated observability dashboards by jurisdiction	Supports multi-region reporting and local legal obligations
SaaS Lifecycle Sync	Observability-as-code integrated into CI/CD pipelines	Aligns feature releases with monetization policy enforcement and telemetry tagging

8. Conclusion

In conclusion, the integration of observability and auditability within SaaS monetization systems emerges as a cornerstone for ensuring transparency, regulatory compliance, and operational resilience in increasingly complex multi-tenant environments. By leveraging metrics, logs, traces, and policy-as-code frameworks, organizations can achieve real-time visibility and traceability across billing workflows, entitlement management, and revenue recognition processes. At the same time, immutable audit trails and compliance-aligned governance models safeguard against disputes, policy drift, and regulatory breaches. Although challenges such as data granularity, cross-tenant isolation, and evolving global standards persist, best practices like structured logging, cryptographically verifiable audit storage, and observability-as-code approaches provide a pathway toward robust, scalable solutions. Looking ahead, the convergence of AI-driven anomaly detection, explainable telemetry, and federated compliance monitoring is poised to redefine the landscape, transforming observability and auditability from supportive functions into strategic enablers of trust, accountability, and sustainable SaaS growth.

9. References

1.Sigelman, B. H., Barroso, L. A., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., ... & Shanbhag, C. (2010). Dapper, a large-scale distributed systems tracing infrastructure.

2.Suicimezov, N., & Georgescu, M. R. (2014). IT governance in the cloud. Procedia Economics and Finance, 15, 830-835.

3.Jia, R., Yang, Y., Grundy, J., Keung, J., & Hao, L. (2021). A systematic review of scheduling approaches on multi-tenancy cloud platforms. Information and Software Technology, 132, 106478.

4.Marie-Magdelaine, N. (2021). Observability and resource management in cloud-native environments (Doctoral dissertation, Université de Bordeaux).

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

5.Rajesh, S. C., & Jain, U. Real-Time Billing Systems for Multi-Tenant SaaS Ecosystems.

6.Mustafa, S., Bilal, K., Malik, S. U. R., & Madani, S. A. (2018). SLA-aware energy-efficient resource management for cloud environments. IEEE Access, 6, 15004-15020.

7.Pourmajidi, W., Zhang, L., Miranskyy, A., Steinbacher, J., Godwin, D., & Erwin, T. (2021). The challenging landscape of cloud monitoring. Knowledge Management in the Development of Data-Intensive Systems, 157-189.

8.Yousif, S. A., & Al-Dulaimy, A. (2017, July). Clustering cloud workload traces to improve the performance of cloud data centers. In Proceedings of the World Congress on Engineering (Vol. 1, pp. 7-10).

9.Gonzalez, H., & Rivera, C. (2025). Designing Resilient Multi-Tenant Architectures for Space Mission Clouds.

10.Jothimani, A. P. (2022). Enabling Secure Cloud Governance using Policy as Code.

11.Moghaddam, F. F. (2017). Multi-layered policy generation and management in clouds (Doctoral dissertation, Georg-August-Universität Göttingen).

12.Färjsjö, F., & Stenberg, E. (2017). Ensuring Continuous Security in the Cloud and Compliance with GDPR.

13.Bhat, S. A., Sofi, I. B., & Chi, C. Y. (2020). Edge computing and its convergence with blockchain in 5G and beyond: Security, challenges, and opportunities. IEEE Access, 8, 205340-205373.

14.Prabadevi, B., Deepa, N., Pham, Q. V., Nguyen, D. C., Reddy, T., Pathirana, P. N., & Dobre, O. (2021). Toward blockchain for edge-of-things: a new paradigm, opportunities, and future directions. IEEE Internet of Things Magazine, 4(2), 102-108.

15.Gadekallu, T. R., Pham, Q. V., Nguyen, D. C., Maddikunta, P. K. R., Deepa, N., Prabadevi, B., ... & Hwang, W. J. (2021). Blockchain for edge of things: Applications, opportunities, and challenges. IEEE Internet of Things Journal, 9(2), 964-988.

16.Katari, A., & Kalla, D. (2021). Cost Optimization in Cloud-Based Financial Data Lakes: Techniques and Case Studies. ESP Journal of Engineering & Technology Advancements (ESP-JETA), 1(1), 150-157.

17.Ravi, V. K., & Musunuri, A. (2020). Cloud cost optimization techniques in data engineering.

18.Kothandapani, H. P. (2023). Emerging trends and technological advancements in data lakes for the financial sector: An in-depth analysis of data processing, analytics, and infrastructure innovations. Quarterly Journal of Emerging Technologies and Innovations, 8(2), 62-75.

Received: August 06, 2025

Volume 38 No. 7s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

19. Sakinala, K. (2025). Monitoring and Observability for Cloud-Native Applications. Journal of Computer Science and Technology Studies, 7(8), 101-115.
20. Yadav, P. S. (2024). Real-Time Insights in Distributed Systems: Advanced Observability Techniques for Cloud-Native Enterprise Architectures. ResearchGate.
21. Ganesan, P. (2022). Observability in Cloud-Native Environments: Challenges and Solutions. International Journal of Core Engineering & Management.
22. Rübsamen, T. (2016). Evidence-based Accountability Audits for Cloud Computing.
23. Rübsamen, T., Reich, C., Knahl, M., & Clarke, N. (2014). An architecture for cloud accountability audits. BW-CAR| SINCOM, 55.
24. Deimling, F., & Fazzolari, M. (2023, July). Amoe: A tool to automatically extract and assess organizational evidence for continuous cloud audit. In IFIP Annual Conference on Data and Applications Security and Privacy (pp. 369-385). Cham: Springer Nature Switzerland.

Received: August 06, 2025