International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Ensuring Data Integrity and Regulatory Compliance in Large-Scale Cloud Database Systems

Ronak Jani*

Lead DBA, Take-Two Interactive, Wesley Chapel, FL, USA
Email: ronak.jani@hotmail.com

Abstract

The purpose of the project is to identify architectural mechanisms that ensure data integrity and regulatory compliance in large-scale cloud database systems operating under distributed and multi-cloud conditions. The research problem lies in reconciling high-throughput transactional performance with cryptographically verifiable state evolution and auditability requirements imposed by modern regulatory frameworks. The study addresses this problem through comparative architectural analysis, structural modeling of ledger-based data structures, and synthesis of contemporary research on confidential execution, audit synchronization, semantic analytics, and CI/CD validation mechanisms. The analysis demonstrates that state-oriented verification models, combined with batching strategies, cryptographic aggregation, and synchronized audit layers, reduce verification complexity and embed compliance into the structural configuration of database engines. The conclusions indicate that sustainable regulatory resilience emerges from coordinated interaction between cryptographic data structures, concurrency control, and lifecycle governance rather than from append-only logging alone. The significance of the project lies in providing a unified analytical framework for designing cloud database systems where scalability and compliance are structurally aligned.

Keywords: audit architectures; cloud databases; cryptographic verification; data integrity; distributed transactions; multi-cloud systems; regulatory compliance; verifiable ledger.

Received: 3/1/2026
Accepted: 5/1/2026
Published: 5/11/2026

---

* *Corresponding author.*

1. Introduction

The rapid expansion of cloud-native infrastructures has intensified the problem of ensuring verifiable data integrity under conditions of distributed control and heterogeneous deployment environments. Regulatory frameworks increasingly require demonstrable protection against tampering, rollback, and unauthorized state modification. Traditional database mechanisms designed for performance and availability alone are insufficient when compliance demands cryptographic evidence of state evolution and linear history.

The objective of this research is to demonstrate that data integrity and regulatory compliance in large-scale cloud database systems can be achieved as intrinsic architectural properties through state-oriented cryptographic verification models, rather than as auxiliary mechanisms layered on top of transaction logging.

The research is guided by the following scientific hypothesis: embedding cryptographic verification into state-indexed data structures, combined with controlled batching and synchronized audit mechanisms, enables scalable transactional performance while satisfying regulatory requirements for verifiability and auditability in distributed and multi-cloud environments.

To achieve this objective, the study addresses the following research tasks:

(1) to examine the architectural transition from log-centric verification models to state-oriented cryptographic ledger structures;

(2) to evaluate the impact of batching strategies and persistence timing on integrity latency and regulatory exposure;

(3) to analyze the role of audit synchronization, confidential execution environments, semantic analytics, and CI/CD validation in extending compliance across the data lifecycle.

The scientific novelty of the study lies in the integrated interpretation of cryptographic ledger databases, audit architectures, and lifecycle governance mechanisms as components of a unified compliance-oriented architectural configuration, rather than as isolated technical solutions.

The remainder of the article is structured as follows. Section 2 describes the methodological framework and materials. Section 3 presents the results of the architectural analysis. Section 4 discusses the implications of the findings for scalability and regulatory resilience. Section 5 concludes the study.

2. Methods and materials

The study is based on an analytical examination of contemporary research devoted to verifiable databases, confidential frameworks, audit architectures, and intelligent analytics.

The study of Cong Yue and his colleagues [1] examined the design space of verifiable ledger databases and proposed a state-oriented POS-tree architecture that integrates transactional semantics with efficient proof

generation. The work of Heidi Howard and his colleagues [2] developed a confidential consortium framework combining integrity, confidentiality, and high availability in multiparty deployments. The study of Varun Gandhi and his colleagues [3] investigated audit architectures capable of synchronous log availability and high event coverage in distributed environments. The work of Marcos K. Aguilera and his colleagues [4] proposed signature batching mechanisms that reduce cryptographic overhead in data center environments. The study of Juan Alonso and his colleagues [5] analyzed architectural models of multi-cloud native applications and identified synchronization asymmetries and governance challenges. The research of Hong Zhong and his colleagues [6] explored knowledge graph–based intelligent auditing and semantic anomaly detection. The study of Jun Song and his colleagues [7] introduced a holistic data management approach for large-scale machine learning workloads with integrated lineage control. The work of Hongtao Yang and his colleagues [8] examined CI/CD integration for distributed data pipelines to prevent transformation drift.

To write the article, a comparative analytical method, structural modeling, systematization of architectural approaches, and source analysis were used. The methodological framework enabled the synthesis of heterogeneous research directions into a unified compliance-oriented perspective.

The applied methodology ensures coherence between cryptographic verification models, architectural batching mechanisms, and regulatory enforcement strategies across distributed cloud systems.

3. Results

The architecture of integrity enforcement in large-scale cloud database systems reveals a structural shift from transaction-level logging toward state-oriented verification regimes. Within distributed environments where the storage provider cannot be assumed trustworthy, the integrity guarantee increasingly migrates from conventional access control to cryptographically anchored data evolution models. Ledger-based database designs demonstrate that tamper evidence is no longer confined to append-only logs; it is embedded into the structural representation of states and their transitions. The systematization of architectural approaches is presented below (Table 1).

Table 1: Structural Approaches to Ensuring Data Integrity in Large-Scale Cloud Database Systems (compiled by the author based on [1-6])

Architectural Dimension	**Log-Centric Ledger Models**	**State-Oriented Ledger Models**	**Confidential / Consortium Frameworks**	**Intelligent Audit Architectures**
Integrity Anchor	Transaction log Merkle tree	State-indexed POS-tree structure	Enclave-protected execution layer	Semantic graph representation
Index Protection	Partial or externalized	Cryptographically protected	Enforced inside the trusted runtime	Contextual cross-entity validation
Verification Model	Inclusion + append-only proofs	Batched inclusion + current-state proofs	Preventive execution constraints	Pattern-based anomaly detection
Scalability Mechanism	Log batching	Structural batching over blocks	Distributed attestation	Graph-based correlation
Trust Distribution	Detection via audit	Detection via audit	Prevention via an enclave	Hybrid structural-semantic model
Regulatory Exposure	Historical reconstruction dependent	Snapshot-verifiable state	Hardware trust assumptions	Probabilistic anomaly detection

Systems constructed on transparency abstractions confirm that append-only verification and fork detection remain foundational security primitives, yet their practical implementation determines regulatory suitability under high-throughput workloads [1].

A decisive transformation emerges when transaction semantics are integrated into verifiable ledger databases. Designs that maintain Merkle trees directly over transaction logs incur verification costs proportional to the total number of transactions, yielding append-only proofs of complexity $O(\log N)$ and current-value proofs of complexity $O(N)$, where N denotes the number of transactions [1]. Such scaling characteristics expose regulatory fragility in environments processing millions of updates daily. Empirical benchmarking demonstrates that ledger systems without optimized batching exhibit significantly lower throughput, with commercial designs categorized as “Low” under comparative analysis, whereas optimized state-based architectures achieve “High” throughput while maintaining logarithmic proof complexity relative to the number of blocks $O(\log B)$ [1].

The introduction of a two-level POS-tree reorganizes the integrity surface of the database. By separating upper-level block commitments from lower-level state indexing, proof generation becomes dependent on block

granularity rather than raw transaction count. Experimental evaluation shows that average tree heights in optimized architectures are 5 for the upper level and 7 for the lower level, compared to 17 in traditional Merkle-based designs [1]. Although each POS-tree node occupies four times the size of a Merkle node, resulting in a per-key proof size of 2.1KB versus 0.69KB in minimalistic Merkle structures, verification latency remains comparable due to effective batching and reduced traversal depth [1]. The structural implication is that regulatory auditability benefits more from controlled structural depth than from a minimal hash footprint.

Throughput measurements under YCSB workloads further clarify the compliance-performance trade-off. Balanced workloads under distributed configurations demonstrate peak throughput improvements of up to 3.7× over log-centric implementations, 1.7× over partially batched variants, and 1.8× over hybrid relational-ledger systems [1]. Under TPC-C workloads containing complex multi-field transactions, optimized state-oriented systems sustain 2.3× higher throughput than disk-bound ledger models, though absolute throughput decreases by 2.1× relative to synthetic key-value benchmarks due to increased coordination overhead [1]. These figures illustrate that regulatory guarantees—when embedded into index structures rather than externalized to logs—do not inevitably constrain operational scale.

Persistence strategy materially influences integrity latency. When persistence intervals extend to 1280ms, abort rates in write-heavy workloads increase to 21.6%, while remaining at 1.5% and 3.5% for read-heavy and balanced workloads, respectively [1]. The numerical divergence indicates that deferred durability must be calibrated to workload composition to preserve both serializability and regulatory consistency. Verification delay similarly reshapes performance envelopes: throughput rises with batching windows up to 800ms but declines beyond that threshold due to network transmission overhead of oversized proofs [1]. Integrity assurance, therefore, operates within a bounded optimization window defined by proof aggregation size and concurrency intensity.

Audit-layer evolution introduces an additional dimension of compliance reinforcement. Architectures emphasizing synchronous log availability achieve high event coverage without sacrificing real-time detectability, countering limitations of asynchronous audit pipelines [3]. The structural insight is that log completeness and temporal proximity jointly determine forensic reliability. Where log coverage is fragmented, compliance validation becomes probabilistic rather than deterministic.

Signature overhead historically constrained large-scale integrity enforcement within data centers. Recent cryptographic batching mechanisms reduce signature amplification costs and eliminate per-operation cryptographic bottlenecks, effectively breaking prior scalability barriers [4]. By decoupling authentication cost from individual transaction frequency, signature-intensive verification ceases to be a limiting factor in compliance-sensitive environments.

Confidential execution frameworks reshape integrity governance in multiparty settings. Secure enclave–based architectures provide integrity and availability guarantees across consortium deployments, mitigating Byzantine risks without relying solely on gossip-based detection [2]. The shift from detection-only models toward prevention-capable coordination reduces exposure windows under adversarial conditions. Nonetheless, reliance on trusted execution hardware introduces a distinct vulnerability surface linked to enclave integrity assumptions.

Data-intensive machine learning workloads impose additional constraints on integrity mechanisms. Holistic data management platforms demonstrate that metadata lineage, state consistency, and workload orchestration must be unified to avoid fragmentation between analytical pipelines and transactional cores [7]. Integrity in such systems extends beyond correctness of stored tuples to reproducibility of derived artifacts, transforming compliance into a lifecycle property rather than a storage attribute.

Continuous integration paradigms within distributed data warehouses extend integrity enforcement into deployment cycles. Automated CI/CD validation of data pipelines reduces configuration drift and prevents schema-level inconsistencies before they propagate into ledger states [8]. Empirical evidence indicates that embedding validation within pipeline execution mitigates regulatory nonconformities arising from silent transformation errors.

Multi-cloud native architectures complicate the verification perimeter. Architectural heterogeneity across providers generates synchronization asymmetries and partial failure modes that traditional single-ledger abstractions do not anticipate [5]. The structural solution increasingly relies on abstraction-layer harmonization and cross-cloud digest reconciliation, rather than centralized authority.

Knowledge graph–driven intelligent auditing further expands verification capabilities. Semantic correlation of transactional events enables anomaly detection across relational and temporal dimensions, surpassing hash-based verification in detecting contextual irregularities [6]. Integrity validation thereby transitions from binary tamper detection to pattern-sensitive compliance reasoning.

Across these trajectories, a coherent pattern materializes: large-scale cloud database integrity is no longer a property of append-only storage alone. It is an emergent configuration formed by cryptographic data structures, batching regimes, concurrency control, audit synchronization, cryptographic acceleration, enclave coordination, lifecycle orchestration, and semantic analytics. Performance measurements confirm that logarithmic proof complexity combined with structural batching sustains tens of thousands of transactions per second under verification workloads exceeding 200×10^3 operations per second in distributed settings [1]. At this operational magnitude, regulatory compliance ceases to be an external overlay and becomes structurally embedded within the database engine itself.

4. Discussion

Integrity in large-scale cloud database systems no longer depends solely on the immutability of transaction logs; it is increasingly shaped by the structural placement of verification mechanisms within the storage engine itself. The results indicate that systems relying on transaction-level Merkle constructions inherit a verification burden proportional to the cumulative number of committed operations. When current-value proofs require $O(N)$ complexity and append-only proofs depend on $O(\log N)$, scalability tensions become visible under sustained high-throughput workloads. The architectural decision to anchor verification at the state level rather than at the transaction log alters that dependency. The structural logic of integrity enforcement is illustrated (Figure 1).

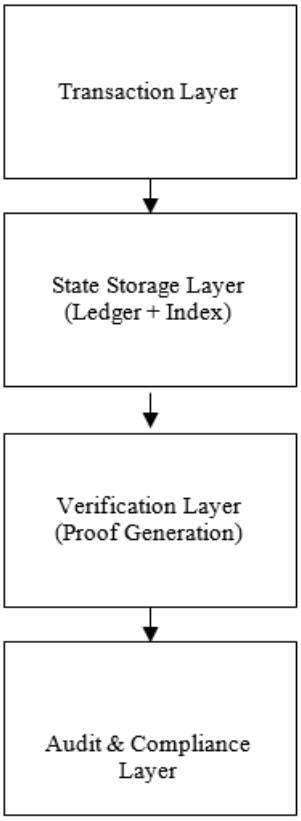

Figure 1: Structural Scheme of Integrity Enforcement in Large-Scale Cloud Database Systems (compiled by the author based on [1-3])

Once verification complexity shifts toward $O(\log B)$ relative to block count, the integrity cost becomes a function of batching strategy rather than operational history volume. The structural implication is significant: regulatory resilience scales with structural depth, not chronological length.

The separation between index integrity and ledger integrity emerges as a decisive boundary. Log-centric systems demonstrate that the absence of index authentication permits stale reads or pointer manipulation without immediate detectability. Regulatory compliance frameworks—particularly those requiring demonstrable non-repudiation and historical consistency—cannot tolerate verification regimes that depend on post hoc scanning of entire histories. The incorporation of hash-protected index structures transforms verification from retrospective reconstruction into structural inclusion. Once the index becomes cryptographically anchored, the server's ability to reorder, omit, or substitute state transitions narrows to detectable boundaries. The ledger ceases to be a passive audit trail and becomes an active constraint on query semantics.

Batching introduces an additional layer of complexity. The numerical evidence demonstrates that throughput peaks within bounded delay windows, after which network overhead and abort rates erode gains. A persistence interval of 1280ms increases abort rates in write-heavy workloads to 21.6%, while balanced workloads remain below 3.5%. This divergence reveals that integrity assurance interacts with workload composition rather than remaining neutral to it. Deferred verification, while reducing contention and enabling proof aggregation, creates

a temporal exposure window. The comparative systematization of timing strategies in ledger-based cloud databases is presented (Table 2).

Table 2: Comparative Characteristics of Timing Strategies in Ledger-Based Cloud Databases (compiled by the author based on [1-4])

Verification Strategy	**Proof Generation Moment**	**Integrity Exposure Window**	**Performance Effect**	**Audit Interaction Model**	**Regulatory Risk Profile**
Immediate (Synchronous) Verification	During transaction commit	Absent	Higher contention, reduced batching efficiency	Continuous real-time validation	Minimal temporal exposure
Short-Interval Deferred Verification	After a short persistence window	Limited, bounded	Optimized batching, balanced throughput	Periodic append-only reconciliation	Controlled temporary vulnerability
Extended Deferred Verification	After an extended batching interval	Increased	High batching efficiency, potential abort amplification	Batch audit confirmation	Elevated short-term exposure
Auditor-Triggered Verification	On the digest reconciliation request	Variable, audit-dependent	Offloaded client overhead	Cross-user digest synchronization	Dependent on audit frequency
Enclave-Enforced Preventive Execution	During execution within a trusted runtime	Structurally constrained	Hardware-dependent overhead	Attested execution verification	Trust is concentrated in the enclave boundary

The system must therefore balance immediate regulatory certainty against operational efficiency. Zero-delay verification eliminates exposure but reduces batching efficiency; extended delay enhances throughput but introduces short-lived integrity vulnerability. Neither configuration is universally optimal.

Audit architecture reshapes the distribution of trust. Detection-only models rely on user gossip or auditor synchronization to reveal forks in history. Prevention-oriented frameworks, particularly those leveraging confidential execution environments, attempt to constrain misbehavior at execution time. The distinction is not

merely theoretical. Detection presumes eventual reconciliation and tolerates transient divergence; prevention seeks to eliminate divergence altogether. In environments where regulatory penalties depend on provable non-occurrence rather than eventual detection, this difference becomes operationally consequential. Yet confidential hardware assumptions introduce their own risk surface, shifting trust from distributed consensus to hardware attestation.

Signature overhead historically constrained verifiable data centers. The emergence of cryptographic aggregation techniques demonstrates that authentication cost need not scale linearly with transaction frequency. Once signature verification ceases to dominate execution latency, integrity enforcement can coexist with tens of thousands of operations per second. The removal of this bottleneck alters architectural trade-offs. Integrity no longer competes directly with throughput; instead, network transmission size and index traversal depth become the dominant constraints.

The relationship between data pipelines and integrity mechanisms extends the discussion beyond storage semantics. Continuous integration of data transformations into ledger-backed warehouses reveals that compliance risk often originates at the transformation layer rather than at the persistence layer. Automated validation within CI/CD workflows constrains schema drift and prevents silent inconsistencies from propagating into immutable states. Integrity, in this configuration, becomes a lifecycle property. The database enforces correctness not only of stored values but of the processes that generate them.

Multi-cloud architectures intensify synchronization complexity. When ledger fragments span heterogeneous providers, digest reconciliation must traverse network partitions, latency asymmetries, and divergent failure semantics. Linear scalability observed under uniform distributed setups does not automatically generalize to federated cloud environments. Cross-provider append-only verification requires consistent block sequencing and harmonized timeout policies. Structural invariance of state trees mitigates some divergence, yet coordination overhead persists. Integrity guarantees in multi-cloud systems depend as much on inter-provider governance as on cryptographic soundness.

Knowledge-graph–driven audit frameworks extend verification from binary tamper detection toward semantic anomaly identification. Hash-based proofs confirm structural correctness; semantic graphs expose relational irregularities that remain invisible to purely structural verification. The integration of contextual analytics shifts compliance monitoring from deterministic validation toward pattern-based inference. This introduces interpretive complexity. Structural proofs provide mathematical certainty; semantic audits provide probabilistic signals. The coexistence of both layers suggests a hybrid compliance model where cryptographic invariants guarantee baseline correctness and semantic reasoning identifies higher-order deviations.

Failure recovery mechanisms illuminate another boundary. Two-phase commit ensures atomicity yet introduces blocking behavior during coordinator failure. Non-blocking alternatives reduce liveness risk at the expense of additional message complexity. From a regulatory standpoint, the priority often lies in preserving atomic integrity rather than minimizing coordination overhead. However, in geographically distributed clusters, prolonged blocking can translate into availability violations that carry contractual implications. The architecture must

reconcile atomic durability with operational continuity.

Storage efficiency contributes indirectly to compliance sustainability. Systems that update ledger structures per operation incur rapid storage growth, increasing retention cost and potentially complicating long-term audit retention policies. State-batched approaches demonstrate lower storage amplification. Reduced structural height and aggregated blocks limit Merkle expansion. The regulatory significance lies in evidentiary retention: smaller, structured proofs facilitate archival persistence and long-term verifiability without exponential growth.

The results collectively expose a conceptual shift. Data integrity in cloud databases no longer resides exclusively in append-only logs, nor exclusively in consensus protocols. It emerges from the coordinated interaction of cryptographic trees, concurrency control, batching windows, audit synchronization, signature aggregation, pipeline validation, and cross-cloud governance. Each layer imposes its own boundary condition. When these layers align, integrity enforcement scales without collapsing throughput. When misaligned, performance optimization reintroduces compliance fragility.

Residual tensions remain visible. Deferred verification cannot eliminate the existence of a temporary integrity window. Trusted execution environments reduce attack surfaces yet centralize trust assumptions. Multi-cloud digest harmonization scales operational complexity. Semantic auditing enhances anomaly detection but complicates interpretability. The architecture of compliance, therefore, resists simplification into a single dominant mechanism.

Large-scale cloud database systems achieve durable regulatory compliance not by isolating integrity into a singular cryptographic primitive but by embedding it into the structural grammar of storage and coordination. The durability of this grammar depends on disciplined synchronization between performance engineering and verification design. The discussion underscores that scalability and compliance are not mutually exclusive properties; they are interdependent outcomes of structural configuration. The findings support the proposed hypothesis that state-oriented cryptographic architectures, when combined with controlled batching and synchronized audit mechanisms, enable scalable integrity enforcement under regulatory constraints.

A more precise interpretation of the results requires a stricter comparison with prior research. Earlier studies largely interpret state-indexed cryptographic structures as a means of reducing verification overhead, implicitly preserving the assumption that integrity remains an externally validated property [1]. The present analysis challenges this assumption by demonstrating that verification, once embedded into state representation and index structures, alters the semantics of data evolution itself. In contrast to audit-centric models, where correctness is reconstructed after execution, the examined architectures constrain admissible state transitions at the structural level, aligning partially with findings on confidential execution and synchronous audit designs, yet extending them toward storage-level enforcement [2,3]. A similar reinterpretation applies to batching strategies. Existing works predominantly frame batching as a throughput optimization mechanism, emphasizing cryptographic amortization. The results indicate that batching simultaneously introduces bounded temporal intervals during which verification is deferred. These intervals are not negligible: under extended persistence windows, abort rates increase markedly in write-intensive workloads, revealing that performance gains are coupled with transient

integrity exposure. Prior models do not explicitly capture this dependency, treating delay as an implementation detail rather than as a parameter affecting regulatory reliability [4]. The limitations of the study are directly linked to its analytical design. The reported performance characteristics derive from heterogeneous experimental environments (including YCSB and TPC-C workloads), which prevents strict comparability of throughput, latency, and proof size across architectures. No unified benchmarking framework was applied, and no independent experimental validation was conducted, which constrains the empirical generalization of the findings. In addition, conclusions involving confidential execution rely on assumptions regarding enclave trustworthiness, while the treatment of multi-cloud systems remains architectural and does not incorporate network-level fault asymmetries or inter-provider latency variability [5].However, the study is subject to several limitations. The analysis is based on synthesized empirical results reported in prior work rather than on independent experimental deployment. Additionally, assumptions regarding trusted execution environments and cross-cloud governance may not generalize uniformly across all regulatory jurisdictions. Future research may extend this framework through empirical validation in heterogeneous production environments and by incorporating jurisdiction-specific compliance constraints into architectural models.

5. Conclusion

The conducted analysis confirms that sustainable data integrity in large-scale cloud database systems requires a transition from transaction-level log verification toward state-indexed cryptographic architectures. The first task has demonstrated that POS-tree–based state anchoring reduces verification complexity from transaction-dependent scaling toward block-oriented scaling, enhancing regulatory resilience. The second task has established that batching and deferred verification must be calibrated within bounded temporal windows to prevent unacceptable exposure while maintaining throughput. The third task has shown that audit synchronization, confidential execution frameworks, cryptographic batching, semantic analytics, and CI/CD validation collectively extend compliance from storage-level guarantees to lifecycle governance.

The study substantiates that regulatory compliance becomes structurally embedded within database engines when cryptographic data structures, concurrency control, audit mechanisms, and semantic monitoring operate in coordinated alignment. Scalability and compliance are therefore not antagonistic objectives but interdependent outcomes of architectural configuration. From a theoretical perspective, the study contributes to database systems research by reframing regulatory compliance as an architectural property emerging from structural coordination rather than as an external enforcement requirement.

Acknowledgements

References

[1] Cong Yue, T. T. A. Dinh, Z. Xie, M. Zhang, G. Chen, B. C. Ooi, and X. Xiao, “GlassDB: An efficient verifiable ledger database system through transparency,” Proc. VLDB Endow., vol. 16, no. 6, pp. 1359–1371, 2023.

[2] H. Howard, F. Alder, E. Ashton, A. Chamayou, S. Clebsch, M. Costa, A. Delignat-Lavaud, C. Fournet,

A. Jeffery, M. Kerner, F. Kounelis, M. A. Kuppe, J. Maffre, M. Russinovich, and C. M. Wintersteiger, "Confidential Consortium Framework: Secure multiparty applications with confidentiality, integrity, and high availability," Proc. VLDB Endow., vol. 17, no. 2, pp. 225–240, 2023.

[3] V. Gandhi, S. Banerjee, A. Agrawal, A. Ahmad, S. Lee, and M. Peinado, "Rethinking system audit architectures for high event coverage and synchronous log availability," in Proc. USENIX Security Symp., Anaheim, CA, USA, Aug. 2023, pp. –, ISBN: 978-1-939133-37-3.

[4] M. K. Aguilera, C. Burgelin, R. Guerraoui, A. Murat, A. Xygkis, and I. Zablotchi, "DSig: Breaking the barrier of signatures in data centers," in Proc. USENIX Annu. Tech. Conf., Santa Clara, CA, USA, Jul. 2024, pp. –, ISBN: 978-1-939133-40-3.

[5] J. Alonso, L. Orue-Echevarria, V. Casola, et al., "Understanding the challenges and novel architectural models of multi-cloud native applications – a systematic literature review," J. Cloud Comput., vol. 12, no. 6, 2023. [Online]. Available: https://doi.org/10.1186/s13677-022-00367-6

[6] H. Zhong, D. Yang, S. Shi, et al., "From data to insights: The application and challenges of knowledge graphs in intelligent audit," J. Cloud Comput., vol. 13, art. no. 114, 2024. [Online]. Available: https://doi.org/10.1186/s13677-024-00674-0

[7] J. Song, J. Ding, I. Kandy, Y. Lin, Z. Wei, Z. Zhou, Z. Peng, J. Shan, H. Mao, X. Huang, X. Song, C. Chen, Y. Li, T. Yang, W. Jia, X. Dong, K. Lei, R. Shi, P. Zhao, and W. Chen, "Magnus: A holistic approach to data management for large-scale machine learning workloads," Proc. VLDB Endow., vol. 18, no. 12, pp. 4964–4977, 2025.

[8] H. Yang, Z. Xu, S. Yudin, and A. Davidson, "Unlocking the power of CI/CD for data pipelines in distributed data warehouses," Proc. VLDB Endow., vol. 18, no. 12, pp. 4887–4895, 2025.