Global Multidisciplinary Journal
eISSN: 2791-173X pISSN: 2791-2760

VOLUME04 ISSUE 08
Published 31-08-2025 *Page No. 20-28*

Zero-Trust Migration and Adaptive Defense for Multi-Tenant Cloud Ecosystems: A Unified Framework Against Lateral Movement, DDoS, and Identity-Driven Threats

Dr. Rafael Moreno

Department of Computer Science, Universidad de Lisboa

ABSTRACT

Background: Multi-tenant cloud computing enables economies of scale, flexible resource sharing, and rapid deployment; however, it raises complex security, privacy, and dependability challenges that span cryptography, architecture, and operational policy (Stallings, 2022; Cloud Security Alliance, 2022). Persistent risks include cross-tenant information leakage, unauthorized access through weak identity constructs, improper isolation at infrastructure and database levels, and auditability shortfalls in outsourced storage (Wang et al., 2010; Moreira, 2019; Yang & Jia, 2012).

Objective: This article develops an integrative, publication-ready theoretical framework for securing multi-tenant cloud systems that is strictly grounded in the provided literature. The framework synthesizes cryptographic primitives for privacy-preserving services, layered isolation strategies for multi-tenancy, adaptive zero-trust controls for dynamic threat surfaces, and measurable auditing mechanisms for storage integrity. The objective is to present a comprehensive design and evaluation methodology that informs both architecture and operations while remaining consistent with established guidance and research findings (Li et al., 2013; Sahai & Waters, 2005; Hariharan, 2025).

Methods: We perform a methodical synthesis of the cited literature to construct a conceptual architecture, accompanied by descriptive protocols and policy constructs. The approach draws on cryptographic approaches for searchable and functional encryption, storage auditing techniques, multi-tenancy isolation models from platform blueprints, database-level resource-sharing strategies, and zero-trust policy principles. For each component we derive threat models, security objectives, design constraints, trade-offs, and verification criteria, referencing empirical and theoretical precedents (Boneh et al., 2005; Gai et al., 2016; Huang & Xing, 2013).

Results: The paper produces: (1) a layered security blueprint for multi-tenant clouds integrating cryptographic controls at the data layer, isolation and scheduling techniques at the compute and hypervisor layers, and zero-trust policies at the identity and control plane; (2) a taxonomy of trade-offs (performance, expressiveness of search, audit overhead, administrative complexity) and mitigation strategies; (3) descriptive protocols for privacy-preserving search, fuzzy and attribute-based access, and storage auditing tailored to multi-tenant semantics; and (4) evaluation and benchmarking recommendations drawing from multi-tenancy database benchmarks and HPC sharing research to operationalize fairness and cost accounting (Gobel, 2014; Breslow et al., 2013).

Conclusions: Secure multi-tenant cloud design requires a coordinated application of cryptographic primitives, isolation engineering, and zero-trust operational controls. No single mechanism suffices: cryptography protects confidentiality and selective search, isolation prevents lateral leakage and contention, auditing ensures accountability, and adaptive policies supply continuous verification. The

framework identifies concrete gaps—particularly the need for standardized, low-overhead searchable encryption interfaces for multi-tenant databases and practical integration pathways for zero-trust within tenant mobility scenarios—and outlines a research agenda for empirical validation and standardization. All claims and design prescriptions are anchored in the referenced literature.

KEYWORDS

multi-tenancy, cloud security, searchable encryption, zero trust, storage auditing, isolation, privacy-preserving services

INTRODUCTION

Cloud computing has transformed the deployment, scalability, and economics of computing services by enabling shared infrastructure and abstracted service models. The central promise—elasticity and cost-efficiency derived from multi-tenancy—imposes unique security and privacy demands that are both architectural and operational (Stallings, 2022; Cloud Security Alliance, 2022). Multi-tenancy, broadly defined, allows multiple independent tenants to operate on shared physical resources while retaining logical separation and assurances of confidentiality, integrity, and availability. The tension between sharing (which produces economic benefits) and isolation (which produces security guarantees) frames the primary problem addressed herein: how to design multi-tenant cloud architectures that preserve tenant confidentiality and integrity while enabling rich functionality such as flexible search and efficient resource sharing.

The literature documents a series of interrelated problems. At the data layer, outsourcing storage and computation to untrusted or semi-trusted cloud providers requires cryptographic approaches that allow functionality—search, computation, aggregation—without exposing plaintext to the provider (Wang et al., 2010; Li et al., 2013; Gai et al., 2016). Existing solutions deliver varying expressiveness: some schemes allow exact-match search, others support boolean queries or limited arithmetic on encrypted data, and some offer attribute/fuzzy identity constructs that accommodate nuanced policy formulations (Boneh et al., 2005; Sahai & Waters, 2005). Yet practical adoption remains limited by performance overheads and engineering complexity, especially when integrated with high-throughput databases and big-data pipelines (Gai et al., 2016; Huang & Xing, 2013).

At the infrastructure layer, the challenge is to provide strong isolation in the presence of shared kernels, hypervisors, container runtimes, and orchestration frameworks. Architectural blueprints and operational guides describe strategies—cells, hosts scheduling, distinct volume types, and resource aggregates—that reduce co-tenancy risk or allow tenants to purchase isolation guarantees (Moreira, 2019; Red Hat, 2019; OpenStack, 2019). Simultaneously, database-level multi-tenancy introduces semantic sharing challenges—schema separation, row-level isolation, and resource fairness—where design choices affect both performance and security (Gobel, 2014; Pallavi & Jayarekha, 2017). Marketplace and accounting perspectives further complicate operational choices: fair pricing models, node sharing in HPC environments, and time-share allocation influence how providers expose and charge for isolation (Breslow et al., 2013; Gmach et al., 2012).

At the control and policy layer, identity and access management are critical. Zero-trust principles, which advocate continuous verification and least privilege, have gained traction as a response to adaptive threat models in cloud contexts—threats that exploit weak identity constructs, lateral movement opportunities, and insufficiently granular controls (Hariharan, 2025; Cloud Security Alliance, 2022). Integrating zero-trust within multi-tenant clouds yields particular friction: how to maintain tenant autonomy over identity, how to preserve privacy while performing continuous policy enforcement, and how to avoid untenable administrative overhead.

Finally, auditability, dependability, and verifiability of outsourced cloud storage remain open problems. Data possession proofs, remote auditing protocols, and integrity verification mechanisms are necessary to align provider incentives and tenant assurances (Yang & Jia, 2012; Wang et al., 2010). The literature explores schemes that offer probabilistic or deterministic proofs of possession and retrievability, yet challenges persist in multi-tenant contexts where auditing must be partitioned, privacy-preserving, and efficient across many tenants.

This article synthesizes the referenced literature to produce an integrative theoretical framework addressing these interlinked challenges. The framework aims to be prescriptive—specifying designs and protocols—while also being analytical, exploring trade-offs, limitations, and open research directions. It is explicitly constrained to the references provided and uses them as the sole evidentiary basis for claims, ensuring fidelity to prior art and guidance.

METHODOLOGY

The methodology followed in this work is a structured, theory-oriented synthesis of the provided references, performed with the explicit goal of generating a coherent, actionable architecture and evaluation approach for secure multi-tenant clouds. The method comprises five stages: (1) literature synthesis and mapping; (2) threat surface decomposition; (3) design articulation across layers; (4) trade-off and verification criteria development; and (5) benchmarking and operationalization guidance.

Literature synthesis and mapping involved categorizing each provided reference by its primary contribution: cryptographic primitives and privacy-preserving services (Boneh et al., 2005; Sahai & Waters, 2005; Li et al., 2013; Gai et al., 2016), storage dependability and auditing (Wang et al., 2010; Yang & Jia, 2012), multi-tenancy isolation mechanisms and platform configurations (Moreira, 2019; Red Hat, 2019; OpenStack, 2019), database multi-tenancy and resource sharing (Gobel, 2014; Pallavi & Jayarekha, 2017), operational economics and node-sharing (Breslow et al., 2013; Gmach et al., 2012), and guidance or surveys of security practices (Stallings, 2022; Cloud Security Alliance, 2022; Dey & Sarkar, 2021). Each work was analyzed to extract design patterns, security objectives, metrics, and constraints.

Threat surface decomposition is conceptual but explicit: we identify classes of threats relevant to multi-tenant clouds—data confidentiality breaches through misconfiguration or compromised infrastructure, unauthorized access through weak identity or credential theft, lateral movement between tenants enabled by insufficient isolation, denial-of-service or resource-exhaustion attacks, and auditing/forensic obfuscation by malicious providers or sophisticated attackers. These threat classes map to objectives that cryptography, isolation, and policy must fulfill (Wang et al., 2010; Cloud Security Alliance, 2022).

Design articulation produces a layered blueprint with specific roles for cryptography (data layer), isolation engineering (infrastructure layer), and continuous policy enforcement (control plane). For cryptographic design, we synthesize approaches from searchable encryption and functional encryption literature—highlighting trade-offs in expressiveness versus performance and integration approaches for databases and big-data systems (Li et al., 2013; Boneh et al., 2005; Sahai & Waters, 2005; Gai et al., 2016). For isolation engineering, we collate best practices and blueprints (cells, volume types, host scheduling, aggregates) and translate them into architectural patterns for tenant assignment and hardened scheduling (Moreira, 2019; Red Hat, 2019; OpenStack, 2019). For policy, we distill zero-trust principles and propose their operationalization in multi-tenant contexts, leaning on contemporary zero-trust analyses (Hariharan, 2025; Cloud Security Alliance, 2022).

Trade-off and verification criteria were derived by interrogating each design decision: what guarantees are provided, at what cost (latency, throughput, administrative burden), and how can these be measured? We propose measurement strategies anchored in existing benchmarking literature—both specialized multi-tenancy benchmarks and HPC sharing measurements—for validating performance, fairness, and accounting correctness (Gobel, 2014; Breslow et al., 2013).

Finally, benchmarking and operationalization guidance maps the theoretical components to deployable practices: how to configure volume types and hosts (OpenStack, 2019; Red Hat, 2019), how to implement multi-tenant database resource sharing patterns (Pallavi & Jayarekha, 2017), and how to design auditing and accountability pipelines (Yang & Jia, 2012; Wang et al., 2010). The methodology is descriptive rather than empirical—no experimental data are collected—consistent with the remit to provide a theory-rich, publication-quality article strictly based on the supplied references.

RESULTS

The synthesis produced four principal outputs: a layered security blueprint for multi-tenant clouds, a taxonomy of cryptographic integration patterns for searchable and privacy-preserving services, an isolation-engineering catalogue for tenant assignment and resource scheduling, and a benchmarking and auditing protocol template.

Layered Security Blueprint

The blueprint organizes the cloud stack into three interacting layers—Data Layer, Infrastructure Layer, and Control Plane—each with distinct responsibilities, mechanisms, and verification criteria.

Data Layer: The primary focus is confidentiality-preserving storage and selective functionality. Cryptographic primitives provide encrypted storage with controlled search and computation capabilities. Searchable encryption and flexible search constructs enable tenants to retain search functionality without revealing plaintext to the provider (Li et al., 2013). Fuzzy identity and attribute-based cryptography support expressive access control policies that map to user attributes or partial identities (Sahai & Waters, 2005). Functional evaluation on ciphertexts, as considered in the evaluation of 2-DNF formulas on ciphertexts, provides a theoretical foundation for limited computation on encrypted data, enabling complex queries without decryption at the provider side (Boneh et al., 2005). Trade-offs include computational overhead and the limited class of expressible queries; the practical approach is to selectively encrypt sensitive columns or fields and to combine cryptography with trusted execution for heavier computations where needed (Wang et al., 2010; Gai et al., 2016).

Infrastructure Layer: This layer implements isolation and resource management through a combination of scheduling, host and cell partitioning, and storage volume types. Multi-tenancy isolation with aggregates—grouping tenants into sets with explicit isolation guarantees—supports coarse-grained isolation strategies that are operationally feasible (Moreira, 2019). OpenStack and Red Hat operational guidance for scheduling hosts and cells and for creating distinct volume types maps directly to the pattern of allowing tenants to select isolation levels for storage and compute (Red Hat, 2019; OpenStack, 2019). Database-level multi-tenancy architectures—schema-per-tenant, table-per-tenant with tenant ID columns, or shared tables with row-level enforcement—each present different security and performance trade-offs; benchmarks and frameworks such as MuTeBench provide evaluation pathways (Gobel, 2014; Pallavi & Jayarekha, 2017).

Control Plane and Policy Layer: Zero-trust principles require continuous authentication, authorization, and policy evaluation for every access and cross-service call. Integrating zero-trust into multi-tenant clouds involves combining strong identity constructs, least-privilege role enforcement, and telemetry-driven policy updates

(Hariharan, 2025; Cloud Security Alliance, 2022). Operationally, this suggests implementing fine-grained tokenization, short-lived credentials, policy proxies mediating all control-plane interactions, and anomaly-detection hooks that trigger re-evaluation when atypical behaviors occur.

Taxonomy of Cryptographic Integration Patterns

The synthesis identifies four principal patterns for integrating cryptography into multi-tenant clouds:

Selective Field Encryption: Used when only certain sensitive fields require confidentiality. This pattern preserves most database functionality and reduces overhead, as non-sensitive fields remain plaintext for indexing and queries (Wang et al., 2010).

Searchable Encryption Gateways: An application-level gateway or middleware handles search tokenization and query translation for encrypted search schemes, reducing the need to retrofit database engines and allowing the provider to host encrypted blobs with searchable indexes managed by the tenant or a semi-trusted proxy (Li et al., 2013; Gai et al., 2016).

Attribute/Fuzzy Identity Encryption: For flexible policy, attribute-based encryption or fuzzy identity constructs allow encryption tied to attributes or approximate identities enabling policy-driven decryption while preserving certain privacy aspects (Sahai & Waters, 2005).

Functional/Partial Homomorphic Evaluation: For limited classes of computation on encrypted data (e.g., boolean expressions, aggregation), functional encryption approaches or homomorphic evaluation can be applied when performance budgets permit. The literature on evaluating 2-DNF formulas on ciphertexts demonstrates expressiveness bounds and motivates hybrid approaches combining encrypted computation with trusted execution for complex workloads (Boneh et al., 2005; Wang et al., 2010).

Isolation-Engineering Catalogue

Operational blueprints drawn from platform guidance and multi-tenancy research yield concrete isolation patterns:

Cell-Based Physical Isolation: Partitioning infrastructure into cells or host pools where tenants are scheduled together or separately according to their isolation requirements. Cells reduce blast radius and simplify policy enforcement (Red Hat, 2019).

Volume Types and Storage Tiers: Exposing volume types to tenants allows different performance and isolation properties (OpenStack, 2019). Tenants requiring cryptographic guarantees may be placed on specialized storage backends or configured to use provider-side encryption with tenant-managed keys.

Aggregate-Based Grouping: Aggregates group tenants with similar isolation requirements and can be combined with scheduling constraints to prevent undesired co-residency (Moreira, 2019).

Tenant-Aware Scheduling and Node Sharing: Node sharing for HPC-style environments requires careful accounting and fairness mechanisms; node sharing can be enabled with price and scheduling controls to balance utilization and isolation needs (Breslow et al., 2013; Gmach et al., 2012). Benchmarks and careful monitoring are necessary to avoid contention-based degradation.

Benchmarking and Auditing Protocol Template

To evaluate security, performance, and fairness, the framework recommends a measurement strategy informed by MuTeBench and HPC sharing studies:

Benchmark Dimensions: Throughput, latency, search expressiveness overhead, audit overhead (bandwidth and compute for proofs of possession), cross-tenant interference measures, and fairness metrics (completion time variance, resource utilization distribution) (Gobel, 2014; Breslow et al., 2013).

Auditing Protocols: Employ remote data possession proofs and retrievability checks adapted to tenant partitioning; the auditor must be able to verify per-tenant integrity without revealing tenant data to the auditor or provider unnecessarily (Yang & Jia, 2012; Wang et al., 2010).

Operational Recommendations: Instrumentation, tenant-specified isolation levels, and differential pricing for isolation-enforced offerings. Implement privacy-preserving analytics on telemetry aggregated with cryptographic protections where necessary (Li et al., 2013; Gai et al., 2016).

DISCUSSION

The framework clarifies complex trade-offs and practical constraints inherent in designing secure multi-tenant clouds. Here we elaborate these trade-offs, the theoretical implications of the chosen mechanisms, counter-arguments, and limitations exposed by the literature.

Trade-offs Between Functionality and Confidentiality

The central cryptographic dilemma is expressiveness versus performance. Schemes that permit rich queries over encrypted data or enable broader functional evaluation on ciphertexts typically incur greater computational overhead and more complex key management (Boneh et al., 2005; Li et al., 2013). Selective field encryption and searchable encryption gateways present a balanced approach: preserve rich functionality for non-sensitive data while protecting sensitive attributes. The counter-argument posits that partial encryption leaves attack surfaces—indexing information and metadata could leak patterns. This concern is addressed in the designs by carefully minimizing metadata leakage and adopting access patterns that obfuscate user behavior when practical (Gai et al., 2016). Moreover, the literature points to hybrid solutions: combine cryptography for high-sensitivity fields with trusted execution environments for heavier computations when provider trust can be constrained by contractual and technical controls (Wang et al., 2010).

Isolation Versus Utilization

Strong isolation patterns—cells, dedicated hosts, and stricter volume types—reduce co-residency risks but increase costs and potentially lower utilization, affecting providers' economic models (Moreira, 2019; Red Hat, 2019). HPC and node-sharing literature proposes nuanced allocation models and pricing strategies to reconcile fairness and utilization. Breslow et al. (2013) illustrate node sharing strategies in HPC that enable fair pricing while maintaining acceptable performance. The theoretical implication is that multi-tenant cloud providers must offer differentiated isolation tiers: tenants with higher security requirements pay for stronger isolation and lower oversubscription, while tenants willing to accept co-residency enjoy lower costs. This aligns with the marketplace models described by Gmach et al. (2012), though the challenge remains in transparent measurement and billing that accurately reflect isolation guarantees (Breslow et al., 2013).

Policy Complexity and Zero-Trust Integration

Zero-trust mandates continuous verification but introduces operational complexity—short-lived tokens, telemetric policy enforcement, and constant re-evaluation of authorizations (Hariharan, 2025; Cloud Security Alliance, 2022). Implementing zero-trust in multi-tenant clouds must reconcile tenant autonomy with provider-level enforcement: tenants control their own authorization semantics, but providers must enforce baseline policy for resource allocation and cross-tenant interaction. One proposed pattern is the use of a policy mediation layer: a tenant-controlled policy service interacts with provider-enforced proxies that mediate calls. This architecture preserves tenant sovereignty while enabling provider-level monitoring for anomaly detection. The counter-argument is that mediation layers necessitate complex trust relationships and may become attack vectors; thus, design must include cryptographic attestation and audit trails to limit compromise effect and to provide forensic clarity (Cloud Security Alliance, 2022).

Auditing, Accountability, and Verifiability

Remote storage auditing has matured conceptually, but practical deployment in multi-tenant contexts raises partitioning and privacy questions. Auditing protocols must verify per-tenant data possession without revealing tenant data to auditors or other tenants (Yang & Jia, 2012; Wang et al., 2010). Probabilistic proofs of retrievability provide scalable checks but may miss targeted deletions unless carefully tuned. The literature suggests periodic, randomized checks combined with tenant-initiated audits and escrowed proofs. Operationalizing these proofs requires careful key management and clear contractual language governing audit frequency, response to failures, and remediation protocols.

Benchmarking and Measurement Complexity

The proposed benchmarking template bridges multi-tenancy database evaluation and HPC node-sharing fairness metrics. MuTeBench demonstrates how to adapt OLTP-bench workloads to multi-tenancy scenarios for database-level evaluation (Gobel, 2014). These approaches, combined with HPC sharing fairness measures, allow providers to quantify cross-tenant interference and resource fairness (Breslow et al., 2013). Yet practical benchmarking faces difficulties: representative workloads, cost of running large-scale tests, and evolving tenant behavior patterns. Consequently, the literature suggests establishing continuously running micro-benchmarks and telemetry analysis to complement periodic full-scale benchmarks (Gobel, 2014).

Limitations and Gaps Identified in the Literature

Several gaps emerge from the synthesis:

Searchable Encryption at Scale: Li et al. (2013) and Gai et al. (2016) outline search and privacy-preserving strategies, but there is limited practical guidance on integrating these schemes into existing high-throughput database systems while preserving acceptable latency. The literature lacks deployment patterns that reconcile index maintenance, updates, and concurrency with encrypted search.

Attribute-Based and Fuzzy Identity Practicality: While Sahai & Waters (2005) provide theoretical foundations for fuzzy identity-based encryption, engineering these schemes into large-scale identity and access management systems that support tenant autonomy and policy flexibility remains under-explored.

Standardized Isolation SLAs: Moreira (2019), Red Hat (2019), and OpenStack (2019) propose isolation strategies but do not converge on standardized service-level agreements that quantify isolation beyond coarse terms. Tenants lack standardized metrics to compare isolation offerings across providers.

Auditing in Multi-Tenant Aggregates: Wang et al. (2010) and Yang & Jia (2012) offer auditing protocols, but multi-tenant aggregates and shared storage backends complicate auditing because of cross-tenant noise and potential leakage. The literature does not fully resolve how to perform privacy-preserving, multi-tenant-aware auditing at large scale.

Research Agenda

To address these gaps, the following research directions are prioritized:

Engineering Searchable Encryption for High-Concurrency Databases: Develop integration patterns, index structures, and update protocols that minimize latency while preserving encrypted search guarantees (Li et al., 2013; Gai et al., 2016).

Operational Attribute-Based Identity Systems: Prototype IAM systems that implement fuzzy or attribute-based cryptography in live tenant-facing services, evaluating performance and policy flexibility under realistic workloads (Sahai & Waters, 2005).

Standardized Isolation Metrics and SLAs: Define measurable metrics—co-residency probability, maximum resource co-sharing ratio, cross-tenant leakage risk—and evaluate them across different isolation offerings (Moreira, 2019; Red Hat, 2019).

Multi-Tenant Auditing Mechanisms: Create auditing protocols that partition proofs of possession by tenant and can be multiplexed efficiently across millions of objects without cross-tenant leakage (Yang & Jia, 2012; Wang et al., 2010).

CONCLUSION

This article presents a comprehensive, literature-grounded framework for securing multi-tenant cloud environments by integrating cryptographic data-layer protections, infrastructural isolation techniques, and adaptive zero-trust policy enforcement. The layered blueprint articulates specific roles for searchable and functional encryption, cell- and aggregate-based isolation, tenant-aware scheduling, and auditing protocols that collectively address confidentiality, integrity, and accountability. The taxonomy of cryptographic integration patterns highlights practical deployment choices and trade-offs: selective field encryption and searchable gateways balance performance and confidentiality, attribute-based schemes provide flexible policy primitives, and functional encryption enables limited computation under strict performance budgets.

Isolation engineering informs how providers can offer differentiated isolation tiers—implemented via host scheduling, cell partitioning, and volume types—while operational economics and fairness concerns demand transparent benchmarking and pricing strategies derived from HPC and multi-tenancy research. Zero-trust principles add a necessary layer of continuous verification and least-privilege enforcement but require careful integration to preserve tenant autonomy and manageable administrative overheads.

The framework identifies clear gaps—particularly in scalable searchable encryption integration, operational attribute-based identity systems, standardized isolation SLAs, and multi-tenant-aware auditing—that create fertile directions for future research and practical experimentation. Addressing these gaps will require interdisciplinary collaboration across cryptography, distributed systems engineering, cloud operations, and policy design.

In sum, securing multi-tenant clouds is not a single-technology problem; it is a socio-technical design exercise that must reconcile cryptographic guarantees, infrastructural constraints, economic incentives, and policy controls. The framework provided here, rooted in the existing literature, offers a starting point for architects and researchers to design, evaluate, and operationalize secure, privacy-aware multi-tenant cloud services.

REFERENCES

1. Stallings, W. (2022). Cryptography and Network Security: Principles and Practice. Pearson.
2. Cloud Security Alliance (CSA). (2022). Security Guidance for Critical Areas of Focus in Cloud Computing. CSA Publications.
3. Dey, S., & Sarkar, S. (2021). Cloud Computing Security: Concepts and Implementation. CRC Press.

4. Wang, C., Wang, Q., Ren, K., Lou, W., & Li, J. (2010). Toward secure and dependable storage services in cloud computing. IEEE Transactions on Services Computing, 5(2), 220-232.
5. Gai, K., Qiu, M., & Zhao, H. (2016). Privacy-preserving data encryption strategy for big data in mobile cloud computing. IEEE Transactions on Big Data, 3(2), 107-119.
6. Hariharan, R. (2025). Zero trust security in multi-tenant cloud environments. Journal of Information Systems Engineering and Management, 10.
7. Li, M., Yu, S., Ren, K., Lou, W., & Hou, Y. T. (2013). Toward privacy-assured cloud data services with flexible search functionalities. IEEE Transactions on Parallel and Distributed Systems, 24(6), 1312-1322.
8. Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. Advances in Cryptology – EUROCRYPT 2005, 457-473.
9. Huang, D., & Xing, T. (2013). A hybrid approach for scalable and secure storage in cloud computing. IEEE Transactions on Computers, 62(6), 1073-1085.
10. Yang, K., & Jia, X. (2012). Data storage auditing service in cloud computing: Challenges, methods, and opportunities. World Wide Web, 15(4), 409-428.
11. Boneh, D., Goh, E.-J., & Nissim, K. (2005). Evaluating 2-DNF formulas on cipher texts. Proceedings of the Theory of Cryptography Conference (TCC), 325-341.
12. Belmiro Moreira. (2019). Multi-tenancy isolation with aggregates. Launchpad Blueprints.
13. Red Hat. (2019). Schedule Hosts and Cells. Red Hat Enterprise Linux OpenStack Platform Administration Guide.
14. OpenStack. (2019). Create and associate a volume type. OpenStack Configuration Reference.
15. Gobel, A. (2014). MuTeBench: Turning OLTP-Bench into a Multi-Tenancy Database Benchmark Framework. The fifth International Conference on Cloud Computing, GRIDs and Virtualization.
16. Pallavi, G. B., & Jayarekha, P. (2017). An efficient resource sharing technique for multi-tenant databases. 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT).
17. Breslow, A. D., Tiwari, A., Schulz, M., Carrington, L., Tang, L., & Mars, J. (2013). Enabling fair pricing on hpc systems with node sharing. SC.
18. Gmach, D., Rolia, J., & Cherkasova, L. (2012). Selling t-shirts and time shares in the cloud.