

Advanced synchronization strategies in metacloud systems for reliable distributed data storage

Ghassan Adnan Hammoodi Al-Bdairi, Saad Talib Hasson

Department of Information Networks, College of Information Technology, University of Babylon, Hillah, Iraq

Article Info

Article history:

Received Dec 23, 2025

Revised Jul 27, 2026

Accepted Aug 12, 2026

Keywords:

Blockchain validation

Data synchronization

Distributed storage

Metacloud

Multi-cloud architecture

ABSTRACT

The exponential growth of distributed data storage and the limitations of single-provider cloud infrastructures have motivated the emergence of metacloud systems, integrated environments that connect multiple independent cloud platforms to enhance data reliability and security. This paper presents an advanced synchronization framework for metacloud systems, addressing the challenge of maintaining consistent and secure data across heterogeneous cloud nodes. Three synchronization strategies are proposed and analyzed: the primary source synchronization algorithm (PSSA), which ensures rapid one-way replication from a central node; the merge synchronization algorithm (MSA), which intelligently reconciles concurrent updates from multiple sources; and the bi-directional synchronization algorithm (BDSA), which provides complete decentralization through peer-to-peer synchronization. A detailed comparative discussion highlights the computational efficiency, reliability, and scalability trade-offs among the three approaches. The findings demonstrate that adaptive integration of these models can significantly improve system fault tolerance and performance. Finally, future research directions are outlined, emphasizing the role of artificial intelligence (AI), blockchain validation, and quantum-safe security in advancing next-generation metacloud synchronization frameworks.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Ghassan Adnan Hammoodi Al-Bdairi

Department of Information Networks, College of Information Technology, University of Babylon

Hillah, Iraq

Email: ghassan782023@gmail.com

1. INTRODUCTION

In recent years, the proliferation of digital data and the increasing reliance on cloud storage platforms have transformed how individuals and organizations manage, share, and protect information [1], [2]. Cloud computing technologies have become an integral component of modern information infrastructure, offering scalability, cost-efficiency, and global accessibility [3], [4]. Major cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud platform (GCP) deliver reliable solutions for data storage and computing resources on demand [5]–[7]. However, as cloud systems have grown in complexity, data security, reliability, and synchronization have become pressing challenges that limit the long-term dependability of these infrastructures [8]. Despite continuous improvements in cloud architectures and encryption mechanisms, most cloud storage systems still operate under the control of a single provider [9], [10]. This centralized management model poses several critical risks [11]. For instance, a technical failure [12], data corruption [13], or even a policy-driven restriction by one provider can instantly make essential information inaccessible [14]. Furthermore, geopolitical factors, government surveillance requests, or censorship laws can lead to the blocking or deletion of user data, even

when the data owner has not violated any regulations [15]. This vulnerability emphasizes the need for a decentralized and redundant storage approach, where information is distributed across multiple independent clouds located in diverse jurisdictions.

To address these limitations, the concept of metacloud was introduced. A metacloud can be defined as a “cloud of clouds”, functioning as a unified platform that interconnects multiple cloud service providers into a single coherent environment [16]. This architecture enhances data reliability by enabling synchronization and replication among several independent cloud nodes. Essentially, the metacloud acts as a coordination layer that ensures data integrity, redundancy, and availability across geographically dispersed servers. In the event that one cloud becomes inaccessible or compromised, the metacloud can instantly restore information from another synchronized source. However, constructing such a metacloud system introduces a new set of challenges related to data synchronization. Synchronization refers to maintaining consistent data across all nodes in a distributed system, ensuring that every modification or update made in one location is accurately reflected in all others [17]. In the context of a metacloud, synchronization must occur between heterogeneous platforms, each with its own access policies, file systems, latency characteristics, and security protocols [18]. Additionally, the system must distinguish between legitimate user operations and unauthorized or erroneous changes that could propagate corruption across the network.

Traditional synchronization mechanisms, such as those used in database replication or distributed file systems, are often inadequate in this context because they assume homogeneous environments with predictable communication links. A metacloud, by contrast, operates in a heterogeneous and dynamic environment. It may include public clouds, private enterprise clouds, and regional service providers, each governed by different standards and service level agreements (SLAs). Therefore, designing synchronization algorithms that are efficient, secure, and adaptive becomes a complex but essential task for ensuring data integrity and system resilience. The primary objective of this paper is to propose and analyze advanced synchronization strategies that can operate effectively within the metacloud environment. These strategies aim to maintain data consistency, minimize synchronization conflicts, and reduce network overhead, even when dealing with multiple independent storage providers. Specifically, three synchronization algorithms are introduced and evaluated: the primary source synchronization algorithm (PSSA), the merge synchronization algorithm (MSA), and the bi-directional synchronization algorithm (BDSA). Each algorithm represents a different approach to managing synchronization flow depending on the system architecture and operational requirements. The proposed strategies are designed to address several important research challenges in distributed metacloud environments, including achieving efficient synchronization across heterogeneous cloud systems without relying on centralized dependency, maintaining an optimal balance between reliability, latency, and computational complexity during synchronization processes, and preventing unauthorized modifications or data loss across interconnected cloud infrastructures.

By exploring these questions, this study contributes to advancing metacloud technologies as a secure and intelligent data management paradigm. Beyond their theoretical significance, the algorithms presented in this paper have practical implications for several domains, including e-government systems, healthcare data management, financial institutions, and international research collaborations, where data availability and confidentiality are critical. Therefore, this paper extends the work on metacloud-based reliability by focusing on the synchronization mechanisms that ensure the accuracy and resilience of distributed storage systems. It establishes a foundation for future developments in adaptive synchronization and intelligent data management, paving the way for a new generation of fault-tolerant, decentralized cloud architectures. The rest of this paper is organized as follows: section 2 presents synchronization strategies in metacloud systems. Section 3 gives the comparative discussion. Section 4 provides results and discussion. Section 5 provides the future recommendations and trends. Lastly, section 6 presents the conclusion of this paper.

2. SYNCHRONIZATION STRATEGIES IN METACLOUD SYSTEMS

The core functionality of a metacloud lies in its ability to synchronize and replicate data across multiple independent cloud platforms to ensure data reliability, security, and accessibility. Synchronization serves as the foundation for maintaining data consistency between clouds that may differ in architecture, operating policies, and service conditions [19]. In a distributed environment composed of heterogeneous clouds, the synchronization process must account for several challenges [20]: i) variability in communication latency and bandwidth, ii) the possibility of partial data updates or failed transfers, iii) conflict resolution between simultaneous file modifications, and iv) security verification to prevent malicious or unauthorized changes. To address these challenges, this study proposes three synchronization strategies for metacloud systems: i) PSSA: it is a unidirectional model, ii) MSA: it is a multi-source model, and iii) BDSA: it is a fully peer-to-peer model. Each strategy is designed for specific system requirements, balancing computational efficiency, reliability, and synchronization accuracy. The following subsections describe these algorithms in detail.

2.1. Primary source synchronization algorithm

The PSSA follows a unidirectional synchronization model in which one node acts as the authoritative source, and all other nodes operate as replication targets [21]. The approach assumes that the primary source always contains the most reliable and updated version of the data. During synchronization, all connected nodes periodically verify their datasets against the primary source. Additionally, Figure 1 shows the diagram of the proposed PSSA method. A hash map structure is maintained to represent the integrity of each file, allowing constant-time $O(1)$ lookup and comparison. Let, n refers to the number of connected cloud nodes, m is the total number of files per node, $F_{i,j}$ denotes the state of file i in cloud j and $H(F_{i,j})$ indicates the hash value of that file.

The synchronization condition can be expressed as in (1).

$$H(F_{i,j}) = H(F_{i,source}) \forall j \in \{1, 2, \dots, n\} \quad (1)$$

If the hash values differ, the target file is updated from the primary source. The total synchronization time complexity can be approximated as in (2).

$$T_{sync} = O(n \times m) \quad (2)$$

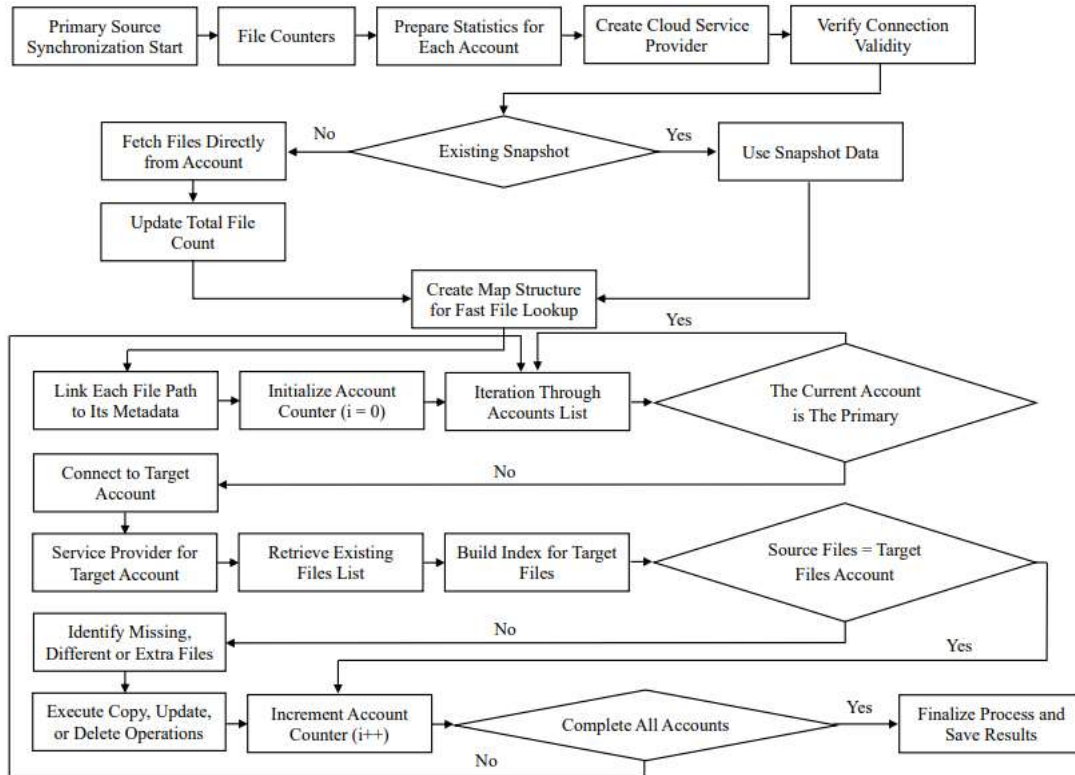


Figure 1. The diagram of the proposed PSSA method

This algorithm ensures rapid and simple synchronization, as only one node initiates data propagation [22]. However, it is vulnerable to single-point-of-failure (SPOF) at the primary source. If the source becomes inaccessible or compromised, the entire synchronization chain halts. In practice, PSSA is effective for hierarchical or centralized metacloud systems, such as those used by government institutions or large enterprises, where strict version control is required, and updates are typically pushed from a central authority to multiple distributed repositories.

2.2. Merge synchronization algorithm

The MSA introduces a multi-source synchronization model that allows updates to originate from several nodes simultaneously. It is built around the concept of intelligent data aggregation, where the system

merges updates from all nodes and resolves conflicts based on file metadata, timestamps, and integrity checks. Furthermore, Figure 2 presents the flowchart of the proposed MSA approach. The process occurs in three main phases, which describes as follows:

- Collection phase: all participating clouds send metadata for recently modified files to a central coordination module. Each file is characterized by its hash signature $H(F_i)$, timestamp $T(F_i)$, and size $S(F_i)$.
- Analysis phase: the system compares metadata from all nodes to detect differences or conflicts. In case of multiple versions of the same file, conflict resolution is achieved using a timestamp-based policy (the latest version prevails) or a majority validation rule.
- Distribution phase: after resolving inconsistencies, the verified version of each file is redistributed across all nodes to restore consistency. The general time complexity for the MSA can be represented as in (3). Where k represents the average number of modified or conflicting files per synchronization cycle.

$$T_{sync} = O(n \times m \times k) \quad (3)$$

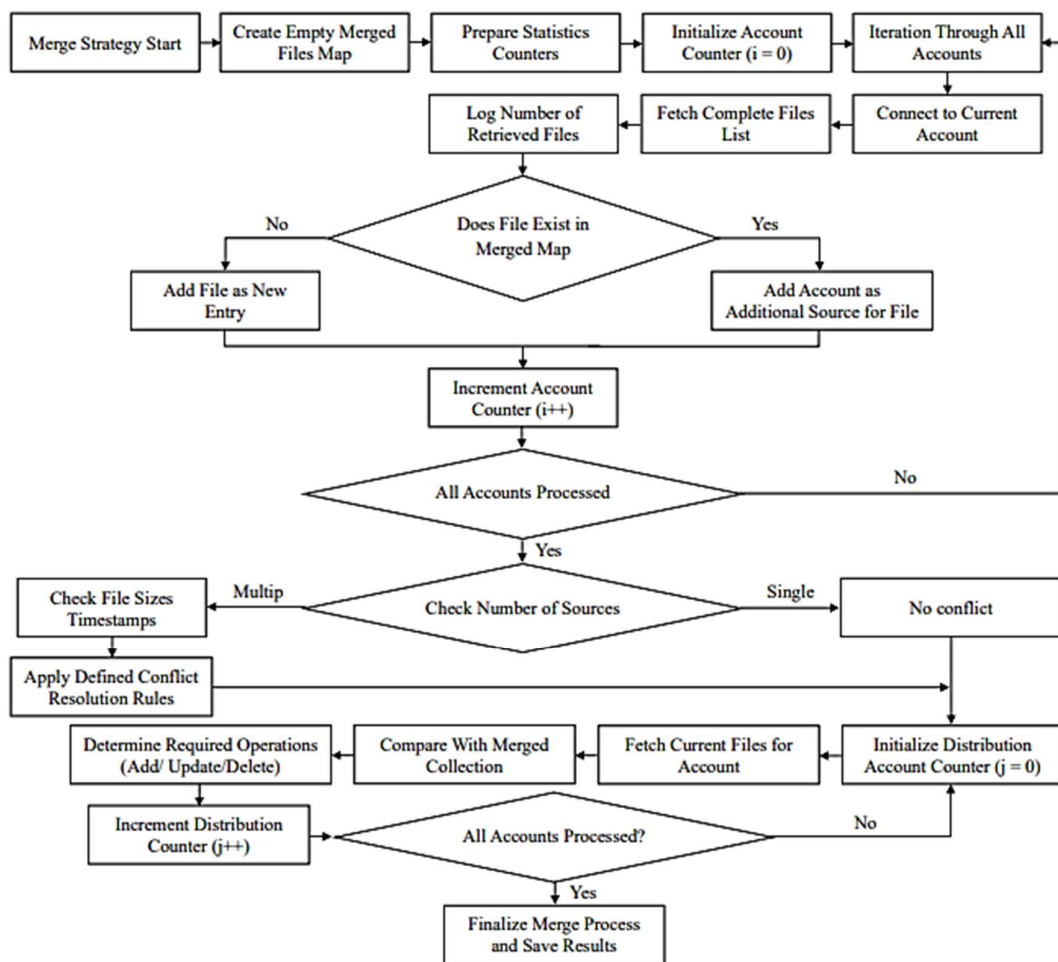


Figure 2. The flowchart of the proposed MSA approach

Although the MSA incurs higher computational cost than PSSA, it provides stronger reliability and resilience in environments where multiple users or institutions collaboratively update data. The intelligent merging capability of this strategy makes it ideal for research networks, collaborative data repositories, or enterprise systems where shared access to synchronized data is essential. Furthermore, by integrating hash comparison and timestamp validation, MSA minimizes the risk of overwriting valid data during conflict resolution.

2.3. Bi-directional synchronization algorithm

The BDSA represents the most advanced synchronization model among the three strategies [23]. It operates under a peer-to-peer architecture, where all nodes function both as data sources and as synchronization targets. Unlike PSSA and MSA, which rely on centralized coordination, BDSA employs direct inter-node communication to ensure comprehensive synchronization. Moreover, the diagram of the proposed BDSA algorithm is illustrated in Figure 3. The model is built upon graph theory principles, where each node in the metacloud is a vertex, and every synchronization link between two nodes represents an edge. For a metacloud consisting of n nodes, the total number of synchronization connections is given by the combination in (4).

$$E = \frac{n(n-1)}{2} \quad (4)$$

Each edge represents a data exchange channel performing bidirectional synchronization. Thus, the time complexity can be approximated as in (5).

$$T_{sync} = O(n^2 \times m) \quad (5)$$

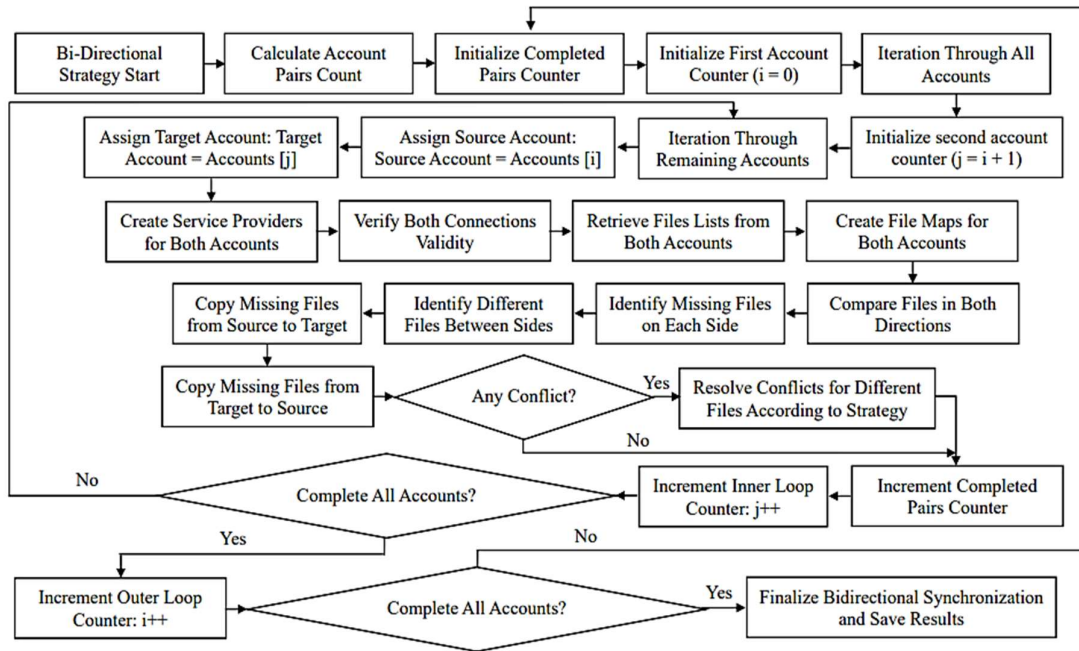


Figure 3. The diagram of the proposed BDSA algorithm

The system continuously monitors file changes in every node and propagates updates to its connected peers. When a modification occurs at any node, the update is transmitted to all adjacent nodes, which in turn synchronize the change with their peers, ensuring eventual consistency throughout the network. The advantages of BDSA include full decentralization, high fault tolerance, and resistance to censorship or regional restrictions, since no single node controls the synchronization process. However, this comes at the cost of increased bandwidth usage and computational load, making it suitable primarily for mission-critical distributed environments, such as intergovernmental networks or blockchain-based cloud infrastructures.

3. COMPARATIVE DISCUSSION

The comparative analysis of the three synchronization strategies (i.e., PSSA, MSA, and BDSA), reveals important distinctions in structural design, coordination complexity, scalability, and overall suitability for different operational contexts within a metacloud system. Each algorithm embodies a unique trade-off

between efficiency, reliability, and autonomy, forming a progressive spectrum that ranges from simple centralized control to fully decentralized peer-to-peer coordination. Table 1 summarizes the computational and operational characteristics of the three synchronization strategies.

Table 1. Summary of characteristics of the three synchronization strategies

Algorithm	Synchronization model	Complexity	Coordination type	Key advantages	Limitations
PSSA	Unidirectional (primary source)	$O(n \times m)$	Centralized	Fast, simple, low overhead	Single-point dependency
MSA	Multi-source Merge	$O(n \times m \times k)$	Semi-centralized	Robust conflict resolution, intelligent merging	Higher computational cost
BDSA	Peer-to-peer (bidirectional)	$O(n^2 \times m)$	Fully decentralized	High reliability, censorship resistance	High bandwidth consumption, complex synchronization

The PSSA represents the most direct and resource-efficient strategy. By relying on a single authoritative cloud as the origin of truth, synchronization becomes straightforward and computationally lightweight. Updates propagate one way from the main node to all secondary nodes, enabling fast deployment, and minimal management overhead. However, this simplicity introduces structural fragility. The entire metacloud's integrity depends on the availability and trustworthiness of the primary source. Any corruption, misconfiguration, or service interruption at that node can cascade through the network, resulting in the synchronized replication of faulty data. Consequently, while PSSA provides an efficient mechanism for static or low-update environments, it lacks the robustness required for dynamic, collaborative ecosystems where bidirectional information flow and validation are necessary.

The MSA extends the synchronization model to accommodate multiple data contributors. Its design reflects real-world environments in which information is continually generated and modified by various organizations or users operating within the same metacloud domain. MSA mitigates the limitations of PSSA by introducing a controlled merging process that collects and analyzes metadata from several nodes before distributing the validated data back across the network. This ensures that each update undergoes verification and conflict resolution, preventing the propagation of inconsistencies. The presence of a coordination layer gives MSA a balanced position between full centralization and total decentralization. It significantly enhances reliability and data accuracy, making it ideal for applications such as international research repositories, shared industrial systems, or federated cloud infrastructures. Nevertheless, the reliance on a coordination component introduces potential performance bottlenecks, especially in large-scale deployments where the number of simultaneous updates is high. Proper optimization of communication protocols and parallel merging mechanisms is therefore critical to maintaining acceptable synchronization latency.

The BDSA represents the most decentralized model, operating through direct peer-to-peer communication among all participating clouds. This strategy eliminates dependence on any central authority and transforms the metacloud into a cooperative network in which each node acts as both a source and a receiver of updates. Its distributed nature provides outstanding fault tolerance: even if several nodes become unreachable, the remaining ones continue to exchange and replicate data seamlessly. Once a disconnected node rejoins, it automatically reconciles differences with its peers, restoring full consistency. BDSA's inherent decentralization also enhances security and resilience. Since no single node controls the synchronization process, malicious interference or censorship targeting a particular provider or jurisdiction cannot compromise the overall network. Such attributes make BDSA highly suitable for mission-critical infrastructures, intergovernmental data systems, or politically sensitive datasets that must remain continuously accessible. However, these benefits come with considerable computational and networking costs. The number of synchronization links grows quadratically with the number of nodes, significantly increasing bandwidth consumption and management complexity. This makes BDSA best suited for specialized environments where reliability outweighs cost and where distributed computing resources can support the added overhead.

From a comparative reliability standpoint, the three algorithms reflect the classical balance between control and resilience. PSSA excels in maintaining centralized consistency but performs poorly when the primary node fails. MSA offers a middle-ground solution, introducing collaborative verification while maintaining coordination efficiency. BDSA, meanwhile, maximizes independence and survivability by removing single-point vulnerabilities altogether. In practical deployment, a system architect would select among these strategies according to operational constraints:

- PSSA for environments requiring strict version authority and minimal system complexity
- MSA for cooperative networks demanding multi-source consistency with manageable overhead
- BDSA for decentralized or adversarial contexts where continuous availability is paramount

Advanced synchronization strategies in metacloud systems for ... (Ghassan Adnan Hammoodi Al-Bdairi)

Beyond static selection, the comparative insights also suggest the potential for adaptive hybrid synchronization. In such a model, a metacloud could dynamically shift between these synchronization modes in response to network conditions, reliability metrics, or security threats. For example, under normal operation, the system might employ PSSA for efficiency; during high-traffic collaborative periods, it could transition to MSA; and in cases of detected network instability or censorship risk, it might switch to BDSA to preserve accessibility. This adaptability would allow the metacloud to combine the speed of centralized control with the resilience of distributed architectures, effectively creating a self-optimizing synchronization environment. Consequently, the comparative evaluation underscores that each algorithm contributes distinct advantages to metacloud reliability. None is universally optimal; rather, their effectiveness depends on the intended purpose, scale, and risk profile of the deployment. Together, they establish a strategic foundation for constructing flexible, intelligent, and secure synchronization frameworks capable of sustaining high data integrity across heterogeneous cloud infrastructures.

4. RESULTS AND DISCUSSION

4.1. Experimental setup and latency model

The performance evaluation of the proposed synchronization strategies was conducted using an end-to-end latency measurement framework. Latency is defined as the total propagation time from initiating a synchronization operation until confirmation that the updated file is fully available across all target cloud providers. This definition captures the complete synchronization lifecycle, including data transfer, processing, and verification stages. To ensure result reliability and reduce the influence of transient network fluctuations, each experimental scenario was repeated five times, and the final values were obtained using the arithmetic mean. The experimental environment was constructed using real application programming interface (API) response measurements from five heterogeneous cloud storage providers, namely Google Drive, Dropbox, Yandex Disk, OneDrive, and Box. The baseline latency characteristics of these providers are summarized in Table 2, which reports the measured list, download, and upload response times.

These results clearly demonstrate the heterogeneity of cloud environments, where significant variations exist in API responsiveness across providers. Such differences directly influence synchronization performance and justify the need for adaptive strategies capable of operating under non-uniform latency conditions. In addition, three file size categories were used to evaluate workload sensitivity, as summarized in Table 3. This classification allows evaluation of synchronization behavior under lightweight, moderate, and data-intensive workloads.

Table 2. Baseline latency measurements of cloud providers (ms)

Provider	List (ms)	Download (ms)	Upload (ms)
Google Drive	150	180	220
Dropbox	130	160	190
Yandex Disk	260	310	380
OneDrive	170	200	240
Box	210	250	290

Table 3. File size categories used in experiments

Category	Size range	Transfer factor
Small	< 1 MB	1.0×
Medium	1–5 MB	3.2×
Large	5–10 MB	7.5×

4.2. Synchronization performance analysis

The main experimental results are presented in Table 4, which reports the average end-to-end propagation latency for all combinations of provider count ($N = 3, 4$, and 5), file size categories, and synchronization strategies, including PSSA, MSA, and BDSA. The results show a consistent performance ordering across all scenarios, where primary source synchronization achieves the lowest latency, followed by MSA, while BDSA consistently exhibits the highest latency. For small file workloads, primary source synchronization records the lowest latency values, ranging from 704 ms to 744 ms depending on the number of providers. Merge synchronization introduces a moderate increase in latency due to additional cross-provider comparison operations, while bi-directional synchronization exhibits the highest latency due to its bidirectional consistency enforcement mechanism. For medium and large file sizes, a significant increase in latency is observed across all strategies. However, the relative performance ordering remains

unchanged, indicating that the underlying synchronization logic consistently governs performance behavior regardless of workload scale. Furthermore, Table 5 presents a consolidated performance comparison, summarizing minimum latency, maximum latency, and average overhead across all experimental configurations. It indicates that primary source synchronization achieves the lowest minimum latency (704 ms), while bi-directional synchronization records the highest maximum latency (5,598 ms). In terms of average overhead, merge synchronization introduces approximately 9% overhead compared to primary source, while bi-directional synchronization increases overhead by approximately 32%. These results highlight a clear trade-off between synchronization efficiency and system consistency, where higher reliability mechanisms introduce additional computational and communication costs.

Table 4. Average synchronization latency (ms) across strategies and system sizes

N	File size	PSSA (ms)	MSA (ms)	BDSA (ms)
3	Small (< 1 MB)	704	832	945
3	Medium (1 – 5 MB)	1,950	2,061	2,545
3	Large (5 – 10 MB)	4,431	4,305	5,598
4	Small (< 1 MB)	717	864	1,003
4	Medium (1 – 5 MB)	1,940	2,032	2,547
4	Large (5 – 10 MB)	4,296	4,600	5,383
5	Small (< 1 MB)	744	843	1,021
5	Medium (1 – 5 MB)	1,935	2,184	2,538
5	Large (5 – 10 MB)	4,440	4,729	5,505

Table 5. Performance summary of synchronization strategies

Metric	PSSA	MSA	BDSA
Min latency (ms)	704	832	945
Max latency (ms)	4,440	4,729	5,598
Avg overhead vs Primary	—	+9%	+32%
Scaling overhead (3 → 5)	+6%	+1%	+8%
Conflict resolution overhead	None	Low	High

4.3. Scaling overhead analysis

The impact of system scalability was evaluated by increasing the number of cloud providers from three to five. The corresponding percentage changes in latency are summarized in Table 6. The results indicate that primary source synchronization exhibits relatively stable scaling behavior, with minimal performance degradation as the number of providers increases. Merge synchronization also maintains controlled overhead, while bi-directional synchronization shows slightly higher sensitivity to scaling due to increased coordination complexity. Despite this, all strategies demonstrate acceptable scalability characteristics for moderate-sized metacloud deployments.

Table 6. Scaling overhead (3-5 providers)

File size	PSSA (%)	MSA (%)	BDSA (%)
Small (<1 MB)	+6	+1	+8
Medium (1-5 MB)	-1	+6	+0
Large (5-10 MB)	+0	+10	-2

The experimental results demonstrate a fundamental trade-off between synchronization speed, scalability, and consistency. The PSSA method achieves the best performance in terms of latency due to its simplified architecture, making it suitable for latency-sensitive applications. The MSA method provides a balanced compromise between efficiency and consistency, while the BDSA method prioritizes data integrity and fault tolerance at the cost of higher latency and computational overhead. Furthermore, the results confirm that file size has a dominant impact on synchronization performance across all strategies. Large file transfers significantly increase latency regardless of algorithm design, indicating that network transmission cost is the primary performance bottleneck in high-volume synchronization scenarios. Overall, the findings suggest that no single synchronization strategy is universally optimal. Instead, the choice of method should be guided by application-specific requirements, including latency tolerance, consistency needs, and system scale.

5. FUTURE RECOMMENDATIONS AND TRENDS

The development of metacloud synchronization strategies is still in its early stages, and the evolution of distributed computing continues to reveal new challenges and opportunities. As the complexity

of global data systems increases, future synchronization frameworks must not only ensure consistency, and reliability but also adapt intelligently to the changing conditions of network infrastructures, user demands, and security threats. This section outlines the major research directions and emerging trends that will shape the future of metacloud synchronization and distributed data management.

5.1. Adaptive synchronization models

One of the most promising directions for future work is the design of adaptive synchronization mechanisms capable of dynamically selecting or combining synchronization strategies based on real-time system states. Rather than relying exclusively on PSSA, MSA, or BDSA, an adaptive metacloud can monitor parameters such as bandwidth availability, node reliability, latency, and user access patterns to automatically determine the most efficient synchronization mode. For example, during periods of stable connectivity and minimal updates, a lightweight unidirectional model (similar to PSSA) could be employed to minimize resource consumption. In contrast, under high collaboration or frequent updates, the system might transition to a multi-source model (MSA) for enhanced conflict resolution. When encountering political restrictions, censorship, or cloud provider failures, the system could automatically activate a peer-to-peer mode (BDSA) to preserve accessibility and redundancy. Such a dynamic synchronization approach would significantly enhance operational flexibility and resilience, enabling the metacloud to self-optimize according to situational demands without human intervention.

5.2. Artificial intelligence and predictive synchronization

Integrating artificial intelligence (AI) and machine learning (ML) into synchronization management represents another critical advancement. Predictive synchronization models can analyze historical usage data, network behavior, and workload patterns to forecast upcoming synchronization events. Such algorithms have proven their efficiency in identifying subjects [24]–[26]. For instance, deep learning algorithms could identify repetitive update cycles or detect potential network bottlenecks before they occur [27], allowing the system to preemptively schedule data transfers or allocate additional resources. Similarly, anomaly detection models can identify suspicious synchronization behavior, such as unauthorized file modifications or irregular access attempts—thereby enhancing system security. AI-based optimization can also minimize energy consumption by prioritizing synchronization tasks according to data sensitivity, size, or time constraints. By applying reinforcement learning (RL) and deep reinforcement learning (DRL), the synchronization framework can dynamically select between PSSA, MSA, and BDSA according to real-time network conditions, bandwidth fluctuations, node availability, and synchronization workloads. Such adaptive learning mechanisms enable continuous optimization of synchronization efficiency, latency reduction, and fault tolerance enhancement over time. In addition, future synchronization frameworks may incorporate explainable artificial intelligence (XAI) techniques such as Shapley additive explanations (SHAP) and local interpretable model-agnostic explanations (LIME) to improve transparency in synchronization decision-making. These methods can provide interpretable insights into why a specific synchronization strategy was selected under particular network conditions, thereby increasing user trust, improving system auditability, and facilitating intelligent resource management in adaptive metacloud environments. Future studies may also compare heuristic synchronization strategies with advanced ML scheduling models, including transformer-based time-series prediction algorithms, to evaluate synchronization forecasting accuracy, adaptability, and computational efficiency under dynamic cloud workloads.

5.3. Blockchain-based validation and trust mechanisms

The next evolution of metacloud synchronization may incorporate blockchain technology to establish a verifiable, tamper-resistant audit trail of synchronization events. Each synchronization transaction can be recorded as a block containing metadata such as timestamp, hash values, and node identifiers. This immutable ledger ensures that every file modification or replication event is transparently logged and cryptographically verified. The decentralized nature of blockchain aligns naturally with the distributed architecture of the metacloud. It eliminates the need for a central authority to validate synchronization integrity, thereby increasing trust among participating providers and users. Moreover, smart contracts can be utilized to automate synchronization policies, enforce access permissions, or trigger rollback operations in response to detected anomalies. In highly sensitive environments such as financial systems or government data networks, blockchain-based synchronization could provide an unprecedented level of accountability, auditability, and resilience against insider threats.

5.4. Edge and fog computing integration

The future of data synchronization will likely extend beyond centralized cloud environments toward edge and fog computing architectures. These paradigms bring computation and storage closer to the data

source, whether internet of things (IoT) devices, sensors, or local servers, reducing latency and improving response time. Integrating metacloud synchronization with edge systems can enable local processing of synchronization tasks before propagating updates to the global network. This hierarchical approach reduces bandwidth consumption and accelerates data consistency at the periphery of the network. In applications such as autonomous vehicles, smart cities, or industrial IoT systems, where real-time data synchronization is critical, combining metacloud and edge computing concepts could provide a unified and efficient infrastructure that supports both global redundancy and local immediacy.

5.5. Quantum-safe synchronization and post-cryptography security

As quantum computing technologies advance, existing cryptographic methods used in synchronization protocols may become vulnerable to decryption. Therefore, future metacloud systems must adopt quantum-safe encryption schemes to protect synchronization channels and stored data. Post-quantum cryptographic algorithms, such as lattice-based and hash-based encryption, are currently being developed to resist quantum attacks. Incorporating these algorithms into synchronization frameworks will ensure long-term data confidentiality and trustworthiness. Furthermore, combining quantum-safe methods with distributed ledger systems and AI-driven authentication could yield a multilayered defense structure, capable of safeguarding synchronization integrity against future computational threats.

5.6. Standardization and interoperability

A recurring challenge in multi-provider cloud synchronization is the lack of uniform standards. Each cloud service employs unique APIs, data formats, and authentication protocols, creating interoperability barriers. Future research should therefore focus on establishing standardized synchronization protocols and interoperability frameworks. International organizations and research consortia could collaborate to define open synchronization APIs and compatibility layers, allowing diverse cloud providers to integrate seamlessly into a unified metacloud ecosystem. This standardization would not only enhance reliability but also reduce vendor lock-in and promote fair competition among providers. Future experimental validation should also include heterogeneous cloud datasets and infrastructures, such as AWS, Microsoft Azure, GCP, and private cloud environments, to evaluate the generalizability and interoperability of the proposed synchronization strategies under diverse operational conditions and provider-specific architectures.

5.7. Sustainable and energy-efficient synchronization

As data volumes grow exponentially, the environmental impact of continuous synchronization becomes a significant concern. Future synchronization frameworks must adopt green computing principles, emphasizing energy-efficient algorithms, intelligent resource scheduling, and low-carbon data routing. ML models can predict low-energy network windows for non-urgent synchronization, while adaptive compression techniques can minimize redundant data transfer. By integrating sustainability metrics into synchronization decision-making, the metacloud can contribute to broader goals of digital sustainability and carbon footprint reduction. Future performance evaluation frameworks should additionally incorporate computational efficiency metrics such as synchronization runtime, bandwidth utilization, memory overhead, and energy consumption to assess the feasibility of large-scale metacloud deployments. These performance indicators can provide deeper insights into the scalability and operational sustainability of synchronization algorithms under high-demand distributed environments.

6. CONCLUSION

This study explored metacloud synchronization as a reliable framework for enhancing data security, integrity, and continuity across heterogeneous cloud environments. A comparative evaluation of PSSA, MSA, and BDSA revealed distinct advantages and trade-offs: PSSA offered efficiency and simplicity for centralized systems, MSA improved consistency through coordinated merging, and BDSA provided superior resilience via decentralized architecture. The findings confirm that no single strategy universally addresses all conditions—optimal reliability depends on context-aware deployment matched to data scale, sensitivity, and volatility. The proposed adaptive metacloud synchronization framework marks a step toward intelligent, self-regulating data infrastructures. Integrating AI, blockchain validation, edge computing, and quantum-safe encryption can enable future systems to achieve greater trust, efficiency, and sustainability—anticipating failures and maintaining stability under unpredictable conditions. The study also underscores the need for interoperability standards and energy-efficient mechanisms to ensure global compatibility and environmental responsibility. Ultimately, the metacloud paradigm bridges centralization and decentralization, offering a dynamic, self-optimizing model that strengthens both the theoretical foundation and practical pathways for secure, scalable, and sustainable digital infrastructure.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ghassan Adnan	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Hammoodi Al-Bdairi														
Saad Talib Hasson	✓	✓		✓		✓		✓	✓	✓	✓	✓	✓	

C : Conceptualization	I : Investigation	Vi : Visualization
M : Methodology	R : Resources	Su : Supervision
So : Software	D : Data Curation	P : Project administration
Va : Validation	O : Writing - Original Draft	Fu : Funding acquisition
Fo : Formal analysis	E : Writing - Review & Editing	

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] A. Sunyaev, *Internet computing: principles of distributed systems and emerging internet-based technologies*, Cham, Switzerland: Springer Nature, 2024, doi: 10.1007/978-3-031-61014-1.

[2] P. Yang, N. Xiong, and J. Ren, "Data security and privacy protection for cloud storage: a survey," *IEEE Access*, vol. 8, pp. 131723–131740, 2020, doi: 10.1109/ACCESS.2020.3009876.

[3] O. R. Arogundade and D. K. Palla, "Virtualization revolution: transforming cloud computing with scalability and agility," *International Advanced Research Journal in Science, Engineering and Technology*, vol. 10, no. 6, 2023, doi: 10.17148/iarjset.2023.106104.

[4] R. Soni, K. Bhatia, and N. Rajput, "A thorough analysis of cloud computing technology," in *Recent Advances in Sciences, Engineering, Information Technology & Management*, London, United Kingdom: CRC Press, 2024, pp. 137–145, doi: 10.1201/9781003598152-19.

[5] R. Gohil and H. Patel, "Comparative analysis of cloud platform: Amazon Web Services, Microsoft Azure, and Google Cloud provider: a review," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1–5, doi: 10.1109/ICCCNT61001.2024.10725914.

[6] Rachita, M. Sharma, S. Chopra, and B. Galhotra, "Comparative analysis of Amazon Web Services (AWS), Azure, and Google Cloud: core services, market strategy, and pricing models," *New Horizons of Science, Technology and Culture*, vol. 4, pp. 33–45, 2025, doi: 10.9734/bpi/nhstc/v4/6019.

[7] P. Borra, "Comparison and analysis of leading cloud service providers (AWS, Azure and GCP)," *International Journal of Advanced Research in Engineering and Technology*, vol. 15, no. 3, pp. 266–278, 2024, doi: 10.17605/OSF.IO/T2DHW.

[8] M. Trigka and E. Dritsas, "Edge and cloud computing in smart cities," *Future Internet*, vol. 17, no. 3, 2025, doi: 10.3390/fi17030118.

[9] A. Ghani, A. Badshah, S. U. Jan, A. A. Alshdadi, and A. Daud, "Issues and challenges in cloud storage architecture: a survey," 2020, *arXiv:2004.06809*.

[10] A. Tahir *et al.*, "A systematic review on cloud storage mechanisms concerning e-healthcare systems," *Sensors*, vol. 20, no. 18, 2020, doi: 10.3390/s20185392.

[11] R. Qureshi and I. Koo, "A comprehensive survey of cybersecurity threats and data privacy issues in healthcare systems," *Applied Sciences*, vol. 16, no. 3, 2026, doi: 10.3390/app16031511.

[12] A. Patel, P. Pandey, H. Ragothaman, R. Molleti, and A. Tanikonda, "Securing cloud AI workloads: protecting generative AI models from adversarial attacks," in *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, 2025, pp. 1–7, doi: 10.1109/ICAIC63015.2025.10848877.

[13] C. Vallin, R. Bhimanadhuni, and R. S. Chappell, "Silent data corruption challenges in modern AI public cloud deployments," *IEEE Micro*, vol. 46, no. 1, pp. 52–54, 2026, doi: 10.1109/MM.2025.3641148.

[14] S. Potluri, "Multi-layered security policy enforcement for confidential data in serverless cloud functions," *International Journal of Emerging Trends in Computer Science and Information Technology*, vol. 6, no. 1, pp. 124–134, 2025, doi: 10.63282/3050-9246.ijetcsit-v6i1p114.

[15] A. R. Gohdes, *Repression in the Digital Age: Surveillance, Censorship, and the Dynamics of State Violence*. Oxford, United Kingdom: Oxford University Press, 2024, doi: 10.1093/oso/9780197743577.001.0001.

- [16] A. Vineela, N. Kasiviswanath, and C. S. Bindu, "Advanced redistribution meta storage algorithm for securing big data in cloud computing," *International Journal of Advanced Intelligence Paradigms*, vol. 29, no. 2–3, pp. 263–277, 2024, doi: 10.1504/IJAIP.2024.142672.
- [17] V. K. Tambi, "Multi-cloud data synchronization using Kafka stream processing," *SSRN Electronic Journal*, 2025, doi: 10.2139/ssrn.5366148.
- [18] W. M. Chen, T. W. Kuo, and P. C. Hsiu, "Heterogeneity-aware multicore synchronization for intermittent systems," *ACM Transactions on Embedded Computing Systems*, vol. 20, no. 5s, pp. 1–22, 2021, doi: 10.1145/3476992.
- [19] E. Kamau, T. Myllynen, S. D. Mustapha, G. O. Babatunde, and A. A. Alabi, "A conceptual model for real-time data synchronization in multi-cloud environments," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 5, no. 1, pp. 1139–1150, 2024, doi: 10.54660/ijmrge.2024.5.1.1139-1150.
- [20] O. Olorunboba, "Architecting resilient multi-cloud database systems: distributed ledger technology, fault tolerance, and cross-platform synchronization," *International Journal of Research Publication and Reviews*, vol. 6, no. 2, pp. 2358–2376, 2025, doi: 10.55248/gengpi.6.0225.0918.
- [21] Y. Weng and Y. Zhang, "A survey of secure time synchronization," *Applied Sciences*, vol. 13, no. 6, 2023, doi: 10.3390/app13063923.
- [22] D. Wang, Z. Mei, H. Zhang, and H. Li, "A novel PSS timing synchronization algorithm for cell search in 5G NR system," *IEEE Access*, vol. 9, pp. 5870–5880, 2021, doi: 10.1109/ACCESS.2020.3048692.
- [23] D. Marfil, F. Boronat, A. Sapena, and A. Vidal, "Synchronization mechanisms for multi-user and multi-device hybrid broadcast and broadband distributed scenarios," *IEEE Access*, vol. 7, pp. 605–624, 2019, doi: 10.1109/ACCESS.2018.2885580.
- [24] N. A. N. Za'im, F. T. AL-Dhief, M. Azman, M. R. M. Alsemawi, N. M. A. Latiff, and M. M. Baki, "The accuracy of an online sequential extreme learning machine in detecting voice pathology using the Malaysian voice pathology database," *Journal of Otolaryngology - Head and Neck Surgery*, vol. 52, no. 1, 2023, doi: 10.1186/s40463-023-00661-6.
- [25] T. A. Tuib, B. H. Saoudi, Y. M. Hussein, T. H. Mandeel, and F. T. Al-Dhief, "Convolutional neural network with binary moth flame optimization for emotion detection in electroencephalogram," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 1172–1178, 2024, doi: 10.11591/ijai.v13.i1.pp1172-1178.
- [26] M. A. A. Albadr *et al.*, "Fast learning network algorithm for voice pathology detection and classification," *Multimedia Tools and Applications*, vol. 84, no. 17, pp. 18567–18598, 2025, doi: 10.1007/s11042-024-19788-3.
- [27] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: a survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019, doi: 10.1109/COMST.2019.2904897.

BIOGRAPHIES OF AUTHORS



Ghassan Adnan Hammoodi Al-Bdairi    received the B.S. in Computer Science from Shatt Al-Arab University College, Iraq in 2002 and also received M.S. in Computer Science from Volodymyr Dahl East Ukrainian National University, Ukraine in 2013. He also received a Ph.D. in the Faculty of Computer Science and Computing Technology, Department of Computer Complex and Networks, Russian Technological University (MIREA), Russia in 2023. Lecturer in Department of Information Networks, College of Information Technology, University of Babylon, Iraq. A researcher specializing in cloud computing applications and a member of the Iraqi Programmers Association. He has participated in several conferences in Russia and Ukraine. His research interests include network engineering, associated security risks, client data preservation in cloud environments, routing protocols, and social network analysis. He can be contacted at email: ghassan782023@gmail.com or ghassan782010@hotmail.com.



Saad Talib Hasson    is a professor of Electrical and Electronic Engineering. He holds a Ph.D. in Operations Research (Network Modeling and Simulation) (2002), an M.Sc. in Applied Sciences (1989), and a B.Sc. in Electrical and Electronic Engineering (1982). He is an expert in network modeling and simulation, wireless sensor networks, and optimization. He supervises researchers and graduate students in the Department of Information Networks, College of Information Technology, University of Babylon, Iraq. He specializes in information technology and has experience in the field of wireless communications, vehicular communications, sensor networks, optimization, and modeling and simulation. skilled in designing, modeling, and analyzing wireless network systems (WNS) with expertise in WSN, reconfigurable intelligent surfaces (RIS), internet of vehicles (IoV), UAV-assisted communication, and wireless power transfer (WPT). He can be contacted at email: Saad.aljebori@uobabylon.edu.iq.