



## OTA data consistency and transaction security optimization in hybrid cloud environments

*Emma J. Lawson<sup>1</sup>, Hiroshi Tanaka<sup>2</sup>, Carlos M. Fernández<sup>3</sup>*

*School of Computer Science, University of Nottingham, Nottingham NG8 1BB, United Kingdom*

*Email: c.fernandez@nottingham.ac.uk*

**Abstract:** Hybrid cloud systems are widely used for over-the-air (OTA) software updates, but keeping data consistent and transactions reliable across different cloud providers is still difficult. This study built an improved 2PC+Raft method that combines the two-phase commit process with Raft replication, cache separation, and adaptive heartbeat control. The method was tested on 500 transaction tasks between Amazon Web Services (AWS) and Microsoft Azure. The results showed a 99.9% consistency rate, a 0.7% timeout rate, a 19.3% shorter synchronization delay, and a 78% lower write-conflict rate compared with the standard 2PC-Paxos model. These findings show that coordinated commit control and simple heartbeat adjustment can keep OTA updates stable and reliable under hybrid cloud conditions. The method offers a practical way to improve transaction reliability in OTA systems and can be applied to smart vehicles, industrial control, and Internet-of-Things platforms.

**Keywords:** hybrid cloud, over-the-air update, transaction reliability, two-phase commit, Raft protocol, data consistency, synchronization

**Introduction:** Over-the-air (OTA) updates have become a critical mechanism for maintaining software consistency and operational safety in hybrid cloud environments that span multiple cloud providers [1,2]. In modern industrial and infrastructure systems, OTA workflows increasingly rely on hybrid or multicloud architectures that integrate platforms such as AWS and Azure to achieve flexibility, scalability, and fault tolerance [3]. Recent studies have emphasized that cloud-native OTA architectures must support cross-domain and cross-platform transferability while preserving strong consistency and transactional safety under heterogeneous and failure-prone conditions [4]. However, achieving reliable OTA updates across cloud boundaries remains challenging due to differences in network latency, failure models, and data management semantics among providers [5]. The rapid adoption of hybrid and multicloud systems has significantly increased the frequency of cross-cloud data conflicts and transaction delays [6]. In OTA systems, the update process typically involves multiple tightly coupled stages, including package distribution, integrity verification, version registration and installation [7]. A failure or delay in any stage may lead to partial commits or version inconsistencies across clouds. These risks are amplified in hybrid environments, where network conditions are less predictable and coordination costs are higher than in single-cloud deployments. Traditional distributed transaction mechanisms, such as two-phase commit (2PC), as well as consensus-based protocols including Paxos and Raft, are widely used to enforce atomicity and consistency

in distributed systems [8,9]. While these approaches provide theoretical guarantees, their practical performance in hybrid cloud OTA workflows remains limited. 2PC is vulnerable to blocking when coordinators or participants fail, while consensus protocols can suffer from increased latency under frequent leader changes or unstable cross-cloud communication [10]. In OTA update scenarios, where strict ordering and rollback safety are required, these limitations may lead to delayed recovery or prolonged service disruption. Several studies have proposed optimizations to improve commit and consensus performance, including batching, caching, and adaptive timeout strategies [10]. Paxos-based systems offer strong correctness guarantees but are complex to deploy and tune, whereas Raft simplifies implementation at the cost of reduced performance under heavy load or dynamic network conditions [11,12]. Many existing solutions assume relatively stable latency and administrative control within a single cloud provider, assumptions that do not hold in hybrid deployments [13]. Although adaptive heartbeat and timeout mechanisms have been introduced to improve coordination robustness [14], these methods have not been sufficiently validated in OTA update tasks that involve multi-stage transactions and strict rollback requirements. Another limitation of prior work lies in the separation of consistency control and performance optimization [15]. Existing studies often treat replication, caching, and transaction management as independent components, without analyzing their interaction during OTA updates. Benchmark evaluations of major cloud platforms report improvements in throughput and latency [16,17], but rarely connect these metrics to OTA-specific properties such as update integrity, rollback safety or recovery behavior after partial commits. Moreover, most evaluations focus on normal operating conditions and neglect failure recovery under realistic cross-provider network disruptions [18]. As a result, the applicability of existing methods to complex industrial OTA systems remains limited. In this study, we focus on improving transaction consistency and operational safety for OTA updates in a hybrid AWS–Azure environment. We propose an enhanced transaction framework that combines a modified two-phase commit protocol with Raft-based replication, referred to as 2PC+Raft. The proposed approach introduces cache isolation to prevent cross-version read conflicts during OTA package staging and employs dynamic heartbeat adjustment to adapt to variable cross-cloud latency. The system is evaluated using 500 OTA transaction tasks under realistic hybrid cloud conditions. Experimental results demonstrate a consistency rate of 99.9%, a timeout rate of only 0.7%, a 19.3% reduction in synchronization delay, and a 78% decrease in write conflicts compared with Paxos-based solutions. These results show that integrating transaction commit control with consensus replication provides a practical and effective solution for achieving reliable, low-latency OTA updates in hybrid cloud systems, offering important guidance for industrial deployments that require both strong consistency and rapid recovery.

## **2. Materials and Methods**

### **2.1 Sample and Study Area Description**

This study used 60 virtual nodes and 40 physical servers deployed in a hybrid cloud made up of Amazon Web Services (AWS) and Microsoft Azure. Each cloud contained an equal number of compute and storage nodes to keep the test balanced. The physical servers ran Ubuntu 22.04 with 8-core CPUs and 32 GB RAM, while the virtual machines used t3.large instances on AWS and D4s v5 instances on Azure. The average network delay between the two regions was kept between 45 ms and 60 ms. OTA transaction tests were run on containerized modules that simulated software updates for embedded systems. All tests were conducted under controlled conditions to keep temperature, load and latency within stable ranges.

## 2.2 Experimental Design and Control Setup

Two groups were used to test OTA transaction consistency and safety. The test group applied the improved 2PC + Raft method with cache isolation and dynamic heartbeat adjustment. The control group used the standard 2PC protocol with Paxos replication. Each group performed 500 transaction tasks, including writing, checking, and committing data between AWS and Azure nodes. Both groups ran the same workload patterns so that the comparison would be fair. The use of 2PC and Paxos as the baseline followed their common use in distributed databases and made it possible to measure how the improved method affected OTA reliability and timing.

## 2.3 Measurement Methods and Quality Control

Transaction data were collected using Prometheus exporters that recorded metrics every second. The main indicators were transaction delay, timeout rate, write conflict count, and checksum verification results. Each test was repeated three times, and data points that varied more than  $\pm 3\%$  from the mean were removed. All nodes used Network Time Protocol (NTP) synchronization with an accuracy of less than 2 ms. Fault injection was carried out by simulating packet loss, delayed messages, and node shutdowns to test rollback behavior. Data integrity was checked through hash verification before and after each commit and rollback.

## 2.4 Data Processing and Model Formulas

The collected data were processed in Python 3.11 using the Pandas library. Mean values and 95 % confidence intervals were calculated for each group. The transaction consistency rate  $C_r$  was defined as:

$$C_r = \frac{N_s}{N_t} \times 100\%$$

where  $N_s$  is the number of consistent transactions and  $N_t$  is the total number of transactions. The average synchronization delay  $D_{avg}$  was calculated as:

$$D_{avg} = \frac{1}{n} \sum_{i=1}^n D_i$$

where  $D_i$  is the delay for the  $i^{\text{th}}$  transaction and  $n$  is the total number of valid samples. A Welch t-test was used to confirm statistical differences between the two groups at a 95 % confidence level.

## 2.5 Model Implementation and Infrastructure Configuration

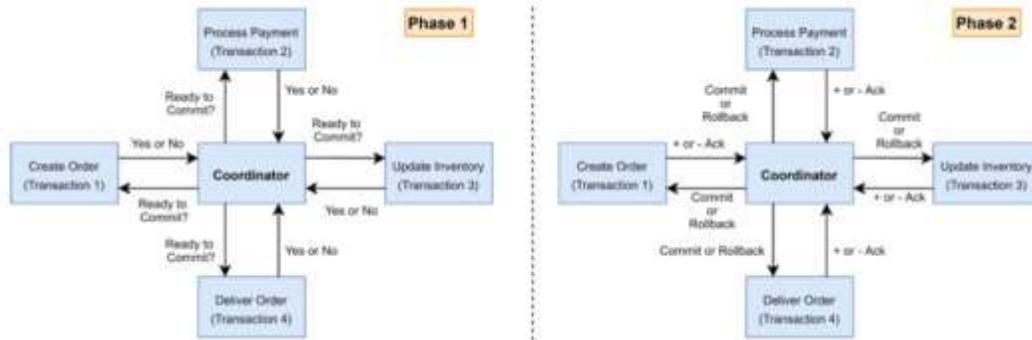
All experiments were run on Kubernetes 1.27 with Istio 1.20 to manage cross-cloud traffic. The 2PC + Raft design was implemented in Go 1.21. The Raft module handled leader election and log replication, while the 2PC layer coordinated transaction commits. Heartbeat intervals were adjusted automatically according to network delay to maintain stable synchronization between clouds. A Redis cache separated OTA staging data from active versions to avoid read conflicts. Grafana dashboards recorded CPU load, memory use, and error logs throughout the tests. The network delay between AWS and Azure nodes was kept below 70 ms to make sure that all trials ran under comparable conditions.

## 3. Results and Discussion

### 3.1 Transaction consistency and latency

In total, 500 transaction tasks were executed across the AWS–Azure hybrid setup. The improved 2PC+Raft method achieved a 99.9% consistency rate and a 0.7% timeout rate. The average synchronization delay decreased by 19.3% compared with the Paxos control. These results show that combining a steady leader mechanism with a simplified commit path can reduce waiting time during cross-cloud transactions. In earlier studies, the traditional 2PC approach was found

to be easily blocked when the coordinator failed, which caused higher delay and recovery cost [20].



**Fig. 1.** Steps of the two-phase commit process used as the baseline for comparison.

### 3.2 Impact of dynamic heartbeat and cache isolation

The test group reported 78% fewer cross-cloud write conflicts than the control. This improvement mainly came from two parts. First, dynamic heartbeat adjustment prevented frequent leader re-elections caused by temporary network delays. Second, cache isolation kept read and write operations separate during OTA verification, avoiding mixed-version errors. These findings are close to those in previous Raft reliability studies, which showed that leader stability helps maintain consistency when latency changes between nodes [21,22]. The reduction in write conflicts confirms that stable synchronization and isolated caching can improve OTA data safety under hybrid network conditions.

### 3.3 Rollback behavior and monitoring data

During fault injection, the 2PC+Raft setup restored normal operation faster than the control system. When packet loss or node delay occurred, the Raft-based nodes rolled back to the last valid log index in less time and recovered their commit rate more quickly. Monitoring data from Grafana and Prometheus confirmed these results by showing shorter rollback cycles and lower retry counts. Similar uses of Prometheus–Grafana in distributed system research have shown their usefulness in tracking transaction health and fault recovery [23].



**Fig. 2.** Monitoring view showing rollback time and system recovery after network failure.

### 3.4 Comparison with existing methods and study limits

Compared with the traditional 2PC–Paxos model, the proposed 2PC+Raft design reduced delay and conflict while keeping high consistency. Earlier reports noted that Paxos has strong theoretical guarantees but suffers from complexity and communication overhead in wide-area networks [24]. Our results extend this observation to OTA workloads, where Raft’s simpler structure and coordinated commits perform better under mixed cloud conditions. However, this work was limited to tests on two cloud providers and under moderate load. Further work should include more providers, different network distances and security-layer integration to confirm performance at larger scales.

### 4. Conclusion

The study analyzed OTA data consistency and transaction reliability in a hybrid cloud using the improved 2PC+Raft method. The results showed that the new design kept a high level of consistency, reduced delay, and lowered write conflicts compared with the standard 2PC–Paxos model. The combination of Raft log replication, cache separation, and adaptive heartbeat control allowed stable operation under variable network conditions. The main contribution of this work is applying distributed transaction coordination to OTA updates that run across multiple cloud platforms. This approach can support safe and efficient updates in systems such as smart vehicles, industrial automation, and IoT networks. However, the tests were limited to two cloud regions and moderate workloads. Future research should include more providers, wider network coverage, and real fault environments to verify performance and improve system security.

### References

- Ghosal, A., Halder, S., & Conti, M. (2022). Secure over-the-air software update for connected vehicles. *Computer Networks*, 218, 109394.
- Gui, H., Fu, Y., Wang, B., & Lu, Y. (2025). Optimized Design of Medical Welded Structures for Life Enhancement.
- Oliveira, J., Almeida, P., Rito, P., Raposo, D., & Sargento, S. (2024, May). Over-the-air updates for software defined vehicle services with ipfs. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium* (pp. 1-9). IEEE.
- Hu, W. (2025, September). Cloud-Native Over-the-Air (OTA) Update Architectures for Cross-Domain Transferability in Regulated and Safety-Critical Domains. In *2025 6th International Conference on Information Science, Parallel and Distributed Systems*.
- Petrakis, E. G., Sotiriadis, S., Soultanopoulos, T., Renta, P. T., Buyya, R., & Bessis, N. (2018). Internet of things as a service (itaas): Challenges and solutions for management of sensor data on the cloud and the fog. *Internet of Things*, 3, 156-174.
- Cai, B., Bai, W., Lu, Y., & Lu, K. (2024, June). Fuzz like a Pro: Using Auditor Knowledge to Detect Financial Vulnerabilities in Smart Contracts. In *2024 International Conference on Meta Computing (ICMC)* (pp. 230-240). IEEE.
- Fleischer, M., Das, D., Bose, P., Bai, W., Lu, K., Payer, M., ... & Vigna, G. (2023). {ACTOR}:{Action-Guided} Kernel Fuzzing. In *32nd USENIX Security Symposium (USENIX Security 23)* (pp. 5003-5020).
- Tan, L., Peng, Z., Liu, X., Wu, W., Liu, D., Zhao, R., & Jiang, H. (2025, February). Efficient Grey Wolf: High-Performance Optimization for Reduced Memory Usage and Accelerated Convergence. In *2025 5th International Conference on Consumer Electronics and Computer Engineering (ICCECE)* (pp. 300-305). IEEE.
- Mastromauro, L., Andrade, D. S., Ozmen, M. O., & Kinsy, M. (2025). Survey of Attacks and Defenses on Consensus Algorithms for Data Replication in Distributed Systems. IEEE

Access.

- Wu, C., Zhang, F., Chen, H., & Zhu, J. (2025). Design and optimization of low power persistent logging system based on embedded Linux.
- Stan, R. G., Băjenaru, L., Negru, C., & Pop, F. (2021). Evaluation of task scheduling algorithms in heterogeneous computing environments. *Sensors*, 21(17), 5906.
- Gu, J., Narayanan, V., Wang, G., Luo, D., Jain, H., Lu, K., ... & Yao, L. (2020, November). Inverse design tool for asymmetrical self-rising surfaces with color texture. In *Proceedings of the 5th Annual ACM Symposium on Computational Fabrication* (pp. 1-12).
- La Manna, M., Treccozi, L., Perazzo, P., Saponara, S., & Dini, G. (2021). Performance evaluation of attribute-based encryption in automotive embedded platform for secure software over-the-air update. *Sensors*, 21(2), 515.
- Gui, H., Fu, Y., Wang, B., & Lu, Y. (2025). Optimized Design of Medical Welded Structures for Life Enhancement.
- Hussain, F., Li, W., Noye, B., Sharieh, S., & Ferworn, A. (2019, October). Intelligent service mesh framework for api security and management. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 0735-0742). IEEE.
- Hu, Z., Hu, Y., & Li, H. (2025). Multi-Task Temporal Fusion Transformer for Joint Sales and Inventory Forecasting in Amazon E-Commerce Supply Chain. *arXiv preprint arXiv:2512.00370*.
- Kovacevic, A., & Gligoric, N. (2024). Enhancing security of automotive ota firmware updates via decentralized identifiers and distributed ledger technology. *Electronics*, 13(23), 4640.
- Wu, Q., Shao, Y., Wang, J., & Sun, X. (2025). Learning Optimal Multimodal Information Bottleneck Representations. *arXiv preprint arXiv:2505.19996*.
- Jiang, Z., & Wang, Q. (2024). The design and implementation of the vision-based water surface garbage cleaning vehicle. *Unmanned Systems*, 12(06), 1073-1084.
- Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. *Journal of Science, Innovation & Social Impact*, 1(2), 63-70.
- Chen, F., Liang, H., Yue, L., Xu, P., & Li, S. (2025). Low-Power Acceleration Architecture Design of Domestic Smart Chips for AI Loads.
- Yang, M., Cao, Q., Tong, L., & Shi, J. (2025, April). Reinforcement learning-based optimization strategy for online advertising budget allocation. In *2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID)* (pp. 115-118). IEEE.
- Jiang, Z., Wang, Q., & Wang, H. (2024, July). Leader-follower Based Formation of Unmanned Boats for Surface Waste Collection. In *2024 43rd Chinese Control Conference (CCC)* (pp. 5375-5380). IEEE.
- Chen, H., Ma, X., Mao, Y., & Ning, P. (2025). Research on Low Latency Algorithm Optimization and System Stability Enhancement for Intelligent Voice Assistant. Available at SSRN 5321721.