Software Engineering
2026, Vol. 12, No. 1, pp. 12–18
https://doi.org/10.11648/j.se.20261201.12

Research Article

An Edge-Cloud Hybrid Architecture for Integrating Standalone and Web Applications in Mission-Critical Systems

Kabeer Vayalilakath[1, *], Abdul Rauf[1], Shada Fathima[2]

[1]Department of Computer Science, University of Calicut, Calicut, India

[2]Department of Computer Applications, Cochin University of Science and Technology (CUSAT), Cochin, India

Abstract

Standalone applications are widely deployed in mission-critical environments due to their high execution speed, reliability, and ability to operate without continuous network connectivity. In contrast, web-based applications offer scalability, centralized data management, and ubiquitous accessibility, but often suffer from latency, offline limitations, and dependency on network availability. Relying exclusively on either paradigm is insufficient for modern enterprise systems that demand both responsiveness and scalability. This paper presents an edge-cloud hybrid software architecture that integrates standalone and web applications to achieve offline resilience, centralized coordination, and scalable collaboration. In the proposed framework, an edge-based local application performs time-sensitive operations and maintains persistent local storage, while a cloud-backed server manages global synchronization, multi-user access, and security enforcement. A conflict-aware synchronization mechanism based on RESTful services and distributed consistency principles ensures reliable data convergence across heterogeneous environments. The framework is implemented and experimentally evaluated using a hospital management system case study. Experimental results demonstrate low synchronization latency, stable resource utilization, and high fault tolerance under increasing workloads. The findings confirm that the proposed hybrid architecture effectively bridges offline and online computing, making it suitable for mission-critical domains such as healthcare, finance, and transportation.

Keywords

Edge-cloud Computing, Hybrid applications, Standalone-Web Integration, Distributed Systems, Hospital Management Systems, Software Architecture

1. Introduction

Over the past two decades, software systems have evolved from standalone desktop applications to distributed and web-based platforms. Standalone applications are traditionally valued for their performance efficiency, robustness, and ability to function without continuous network connectivity, making them suitable for environments with strict reliability requirements [1]. Conversely, web-based applications provide cen-

*Corresponding author: kabeer@farookcollege.ac.in (Kabeer Vayalilakath)

Received: 21 December 2025; **Accepted:** 5 January 2026; **Published:** 27 January 2026

tralized control, simplified deployment, and global accessibility, enabling large-scale collaboration and data sharing [2].

Despite these advantages, both paradigms exhibit inherent limitations. Standalone systems often lack scalability and centralized data coordination, while web-based systems are highly dependent on network availability and may experience latency or service disruption under unstable connectivity conditions [3]. These limitations become particularly critical in mission-critical domains such as healthcare, banking, and transportation, where uninterrupted access, data availability, and real-time responsiveness are essential [17, 18].

Recent advances in distributed computing have motivated the adoption of edge-cloud hybrid architectures, particularly in healthcare systems where cloud scalability must be balanced with latency, availability, and data locality requirements [17, 19]. Such architectures aim to preserve the responsiveness and resilience of local systems while benefiting from the scalability and coordination capabilities of cloud platforms. However, existing solutions often emphasize either offline-first operation or centralized scalability, without offering a unified framework that effectively balances both.

This paper addresses this gap by proposing an edge- cloud hybrid architecture that integrates standalone and web applications into a cohesive system capable of offline operation, centralized synchronization, and scalable collaboration.

The main contributions of this paper are as follows:

A practical edge-cloud hybrid architecture that integrates standalone and web applications while preserving offline functionality.

A conflict-aware synchronization mechanism ensuring data consistency across local and centralized databases.

An experimental implementation and evaluation using a hospital management system case study.

A performance analysis demonstrating scalability, fault tolerance, and low synchronization latency.

2. Literature Review

Standalone applications have historically formed the foundation of enterprise computing due to their localized processing efficiency and minimal dependency on external infrastructure [4]. Early client-server models attempted to overcome isolation by introducing centralized databases; however, tight coupling between clients and servers often resulted in performance bottlenecks and reduced fault tolerance [5].

The emergence of distributed systems introduced principles such as transparency, scalability, and resilience [5]. Cloud computing further advanced this paradigm by enabling elastic resource allocation and global accessibility [6]. Nevertheless, studies have consistently reported challenges related to latency, bandwidth dependency, and security in fully cloud-based systems [7, 10].

To mitigate these issues, researchers have explored hybrid and offline-first designs. Progressive Web Applications and cross-platform frameworks provide limited offline support and code reusability, but often struggle with persistent consistency and deep system integration [8]. More recent work emphasizes edge-assisted hybrid architectures, where local processing complements centralized coordination [9, 13].

In mission-critical domains such as healthcare and transportation, hybrid architectures have demonstrated improved reliability and operational continuity by combining local data handling with centralized synchronization [9, 14]. From a data management perspective, eventual consistency models [15] and Conflict-free Replicated Data Types (CRDTs) [12] enable concurrent updates without global locking, making them well-suited for intermittently connected systems.

Security remains a key concern in hybrid deployments spanning edge and cloud layers. Multi-tier security models incorporating encryption, authentication, and access control are essential for protecting sensitive data across distributed environments [10, 16]. Overall, the literature converges on the conclusion that neither purely standalone nor fully cloud-based architectures sufficiently address the requirements of modern mission-critical applications, reinforcing the need for integrated edge-cloud solutions.

Cloud computing has been extensively explored in healthcare systems to support scalable electronic health records, clinical decision support, and data sharing across institutions. Studies report that cloud-based healthcare platforms improve accessibility and cost efficiency but also introduce concerns related to latency, security, and regulatory compliance [17, 18, 20]. These limitations have encouraged the adoption of hybrid and edge-assisted architectures that retain local operational control while leveraging cloud-based coordination [19, 21].

To clarify novelty, we compare our architecture with two widely used edge-cloud solutions and a healthcare-focused hybrid AI example. Wang et al. propose an edge-cloud integrated framework for hybrid stream analytics that prioritizes low-latency inference at the edge coupled with high-capacity cloud training and modular task placement for streaming Recurrent Neural Networks (RNN) workloads [22, 23].

KubeEdge is a production-grade, Kubernetes-native edge platform that extends cloud orchestration to edge nodes and focuses on container lifecycle management, device integration, and metadata synchronization for large-scale edge deployments [24].

Recent healthcare-specific hybrid AI work by Nguyen *et. al.* demonstrates end-to-end pipelines where edge devices perform local inference/pre-processing while global models and digital twin maintenance occur in the cloud, with federated or secure aggregation used to protect patient data and enable collaborative learning across sites [25]. These healthcare frameworks are optimized for privacy, clinical-grade inference, and real-time monitoring but typically rely on specialized AI workflows rather than on transactional offline reconciliation for legacy standalone systems [26].

By contrast, our work focuses on the integration of standalone applications with web-based systems in mission-

critical environments such as hospital management. The proposed approach employs an edge-resident standalone client that maintains persistent local storage to support uninterrupted operation during network outages. Local updates are recorded using a change-log mechanism and synchronized with the central system in batches once connectivity is restored. Conflict-aware reconciliation techniques, including timestamp-based rules and CRDTs, are used to resolve inconsistencies across distributed data copies. This design ensures transactional integrity and consistent system behavior under intermittent network conditions, which are not explicitly addressed by the existing edge–cloud frameworks.

3. Edge-Cloud Hybrid Architecture and Methodology

3.1. Architectural Overview

The proposed framework adopts an edge-cloud hybrid architecture, a model increasingly recommended for healthcare information systems due to its ability to support offline operation, low-latency access, and centralized data coordination [19, 20]. It is composed of three primary layers:

Edge Layer (Standalone Application): A local application deployed on client machines performs time- critical operations and maintains a local database. This layer ensures high responsiveness and uninterrupted service during network outages.

Synchronization Layer: This layer manages data exchange between the edge and cloud layers using secure RESTful APIs. It handles change detection, version control, and conflict resolution to maintain consistency across distributed environments.

Cloud Layer (Web Application): The centralized server provides global data aggregation, multi-user access, reporting, and administrative control.

Figure 1 illustrates the working model of the proposed hybrid application from the end-user perspective.

The central server layer serves as the backbone of the system, providing data storage, synchronization, and web-based access for distributed users. A dedicated synchronization mechanism ensures consistency between local and global databases, incorporating conflict resolution policies to handle concurrent modifications.

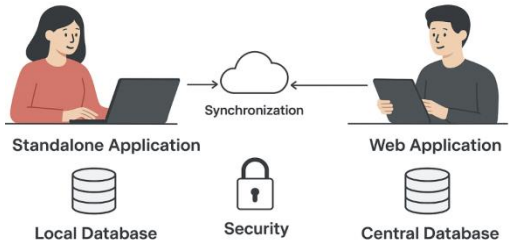


Figure 1. *Working model of the edge-cloud hybrid application.*

3.2. Layered Architecture

The internal structure of the system follows a layered design integrating security at multiple levels.

The architecture incorporates a three-tier security model, consistent with best practices recommended for cloud-based healthcare systems to address privacy and data protection requirements [20], ensuring secure authentication, encrypted communication, and controlled access across all layers.

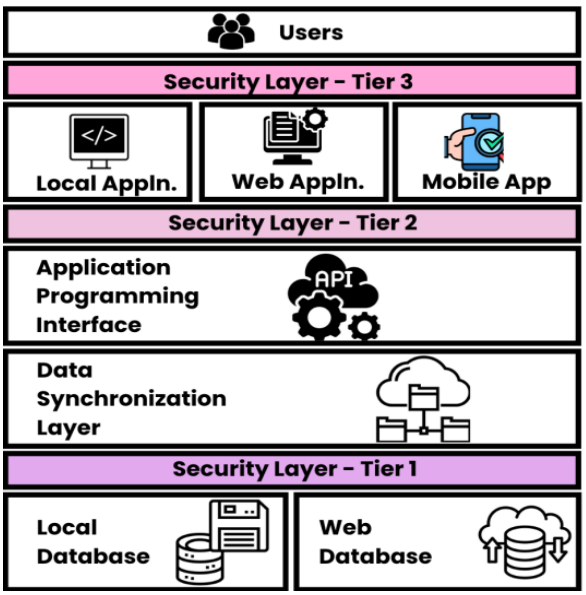


Figure 2. *Edge-Cloud Hybrid Architecture.*

From a technological perspective, the proposed model can be realized using lightweight local databases, such as SQLite, in combination with scalable backend systems including PostgreSQL or MongoDB [11]. Communication between the local and central layers is facilitated through APIs or microservices, while synchronization mechanisms such as CRDTs contribute to consistency and fault tolerance in distributed environments [12]. The layered structure of the proposed edge–cloud hybrid architecture is illustrated in Figure 2.

3.3. Comparative Analysis

The proposed hybrid architecture offers several distinct advantages. It combines the robustness and responsiveness of local applications with the scalability and manageability of centralized systems, thereby ensuring high performance and continuous service availability. In addition, it provides strong fault tolerance, enabling uninterrupted operations even during network disruptions. Furthermore, the architecture enhances overall user satisfaction by supporting fast offline interactions while simultaneously offering collaborative online functionalities.

Table 1. *Comparative analysis of application architectures.*

Feature / Aspect	Standalone Application	Web Application	Hybrid Model
Performance	High	Network- dependent	High (local + Cloud)
Scalability	Limited	High	Mid to High
Offline Capability	Full	Minimal	Full with Sync
Data Centralization	None	Centralized	Centralized + caching
Fault Tolerance	Hardware dependent	Network dependent	High
Suitability for Critical Systems	Limited	Moderate	High
Operational Continuity	Vulnerable if hardware fails	Vulnerable if network fails	Highly resilient

A comparative analytical study highlighting the strengths of the proposed hybrid model against traditional standalone and web-based systems is presented in Table 1.

4. Edge-Cloud Synchronization Procedure

The synchronization process (refer to Figure 3) begins with the client collecting all offline operations recorded in its change log since the last successful sync, packaging them into a cumulative update set. This set is then securely transmitted to the server, where each change is validated. The server checks whether the local record version matches the current server version: if it does, the update is safely applied; if not, a conflict is detected. Conflicts are resolved using strategies such as last-write-wins for simple data, Conflict free Replicated Data Type (CRDTs) for collaborative data types, or manual overrides for critical records.

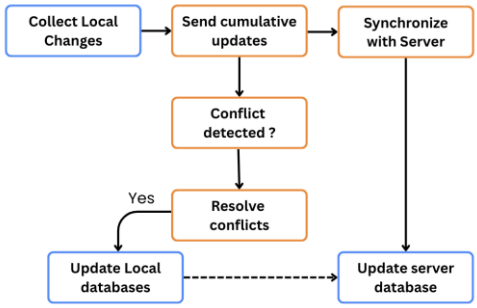


Figure 3. *Flowchart of the synchronization process.*

Once conflicts are resolved, the server commits the updates, logs them, and returns an acknowledgment along with any new updates made by other clients since the last sync. The client then integrates these server updates into its local database, clears committed operations from the change log, and resets

the synchronization timestamp. This step-wise approach ensures continuous offline operation, minimizes network traffic by batching updates, maintains data integrity and consistency across clients, and provides fault tolerance and scalability. By combining local responsiveness with centralized coordination, it is particularly suited for critical domains where both performance and reliability are essential. This approach minimizes network overhead, supports continuous offline operation, and ensures data integrity across distributed nodes.

5. Case Study and Experimental Setup

5.1. Hospital Management System Case Study

Hospitals require information systems that remain operational despite unreliable connectivity, a challenge widely reported in cloud-based healthcare deployments [17, 18]. The proposed framework was applied to a simulated hospital management system, enabling offline execution of critical tasks such as patient registration, billing, and appointment scheduling. Once connectivity is restored, local updates are synchronized with the central server, ensuring consistency across departments [13].

5.2. Implementation Details

The prototype was implemented using Python 3.11 and Django 5.0. Data exchange was performed via REST APIs over HTTPS. While SQLAnywhere was used in the prototype, the framework is database-agnostic and compatible with modern relational and cloud-native databases such as PostgreSQL or MySQL [11].

5.3. Experimental Environment

Experiments were conducted under the following conditions:

Client: Windows 10, Intel i5, 8 GB RAM

Server: Windows Server 2016, Intel i5, 8 GB RAM

Network: Local LAN (50 Mbps)

Synchronization Interval: 10 seconds

The system was evaluated for latency, synchronization accuracy, conflict rate, and CPU utilization during operation.

6. Results and Discussion

The experimental evaluation demonstrates that the proposed synchronization mechanism exhibits stable and predictable behavior under progressively increasing workloads, as illustrated in Figure 4. As the number of synchronized records grows, synchronization latency increases in a near-linear manner, reaching approximately seven seconds at higher record volumes, while no synchronization conflicts were detected throughout the evaluation. These findings empirically validate the conflict-aware synchronization strategy proposed in this work, confirming its ability to preserve data consistency across distributed replicas under intermittent connectivity conditions, thereby supporting the second contribution outlined in the Introduction.

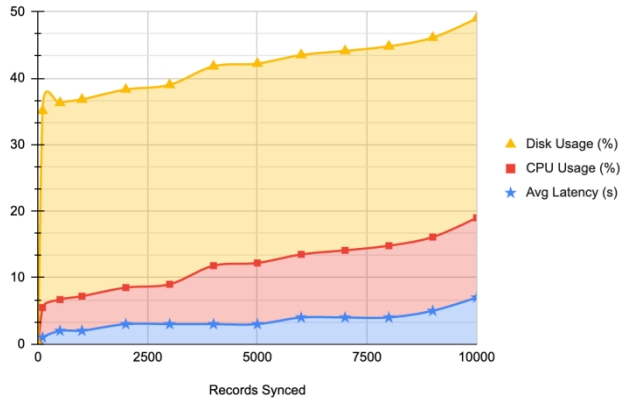
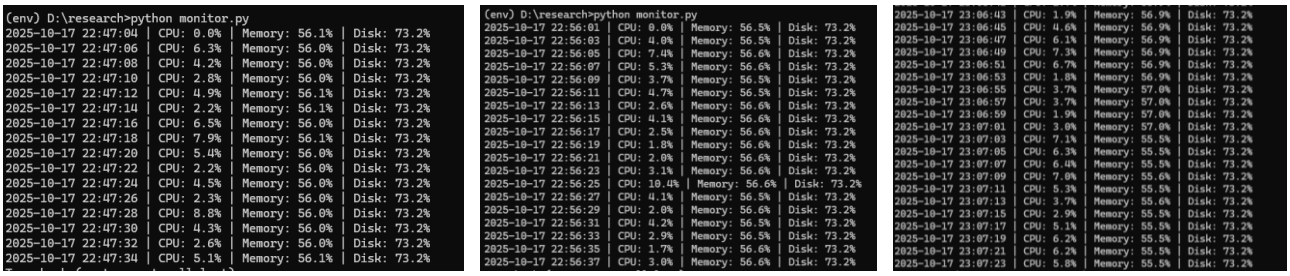


Figure 4. *Key performance metrics for the HMS use case.*

From an architectural perspective, the observed results substantiate the first contribution of this study - the edge–cloud hybrid architecture for integrating standalone and web applications. Specifically, the system's ability to sustain uninterrupted local transaction processing during network unavailability, combined with reliable post-reconnection reconciliation through batched change-log synchronization, demonstrates that the proposed design effectively bridges offline and online operational modes. This behavior directly addresses the research objective of enabling transactional continuity and consistency in mission-critical environments without imposing continuous network dependence.

The experimental implementation and evaluation using a hospital management system case study further support the third contribution of this work. The results indicate that the framework maintains acceptable performance characteristics as data volume increases, while ensuring completeness and integrity of synchronized clinical records. By eliminating the need for manual reconciliation and reducing the risk of data loss during connectivity disruptions, the proposed model enhances workflow reliability and supports scalability requirements inherent to real-time healthcare information systems.

A limitation identified during the evaluation is a modest increase in network utilization when synchronization cycles are triggered at high frequency. While this behavior does not adversely affect correctness or consistency, it suggests scope for future optimization through adaptive synchronization policies that dynamically adjust synchronization intervals based on workload intensity and system activity. Representative workload test observations supporting this analysis are illustrated in Figure 5.



a) 100 records syncing in Local PC b) 500 records syncing in Local PC c) 1000 records syncing in Local PC

d) 100 records syncing in Cloud Server e) 500 records syncing in Cloud Server f) 1000 records syncing in Cloud Server

Figure 5. *Results of Experimental Study.*

7. Conclusion and Future Work

This paper presented an edge-cloud hybrid architecture that integrates standalone and web applications to overcome the limitations of each paradigm. By combining local processing at the edge with centralized cloud coordination, the framework delivers high availability, fault tolerance, and scalable collaboration.

The hospital management system case study demonstrates the practical applicability and performance benefits of the proposed approach. Experimental results confirm low latency, reliable synchronization, and efficient resource utilization.

Future work will explore edge intelligence, adaptive synchronization strategies based on workload patterns, and zero-trust security models to further enhance reliability and scalability.

Abbreviations

CRDT	Conflict-free Replicated Data Types
RNN	Recurrent Neural Networks
API	Application Programming Interface
REST	Representational State Transfer
HMS	Hospital Management System
RUSA	Rashtriya Uchchatar Shiksha Abhiyan
DBT	Department of Biotechnology

Acknowledgments

The authors would like to express their sincere gratitude to the college management for providing the necessary infrastructure and a supportive academic environment for conducting this research. The authors also acknowledge the facilities and institutional support made available through the RUSA (Rashtriya Uchchatar Shiksha Abhiyan) scheme and the DBT Star College Scheme, which indirectly contributed to the successful completion of this work. In addition, the authors gratefully acknowledge RITS Software for sharing server space and providing access to data records that enabled system testing and experimental validation of the proposed framework.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2015.

[2] A. Rosen, “The Architecture of Web Applications,” *Communications of the ACM*, vol. 63, no. 4, pp. 78–87, 2020.

[3] I. Alsmadi and M. Zarour, “Web Application Performance and Scalability Issues,” *Journal of Systems and Software*, vol. 167, 2020.

[4] R. Pressman and B. Maxim, *Software Engineering: A Practitioner’s Approach*, 9th ed., McGraw-Hill, 2019.

[5] A. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, Pearson, 2017.

[6] M. Armbrust et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[7] L. Tawalbeh et al., "Cloud Computing Security Challenges," *Future Generation Computer Systems*, vol. 72, pp. 267–283, 2017.

[8] I.Malavoltaetal.,"HybridMobileApps,"in*Proceedings of the MOBILESoft Conference*, 2015.

[9] R. Randhawa and A. Chhabra, "Hybrid Hospital Management System," *International Journal of Computer Applications (IJCA)*, vol. 97, no. 7, 2014.

[10] A. Ometov et al., "Security in Cloud, Edge, and Fog Computing," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, 2020.

[11] M. Stonebraker, "The Case for PostgreSQL," *ACM SIGMOD Record*, vol. 34, no. 3, 2005.

[12] M. Kleppmann and A. Beresford, "Conflict-Free Replicated Data Types," in *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, 2017.

[13] H. Al-Bahadili, "E-Health Hybrid Systems," *International Journal of Medical Informatics*, vol. 137, 2020.

[14] R. Singh and P. Sharma, "Hybrid Reservation Systems," *International Journal of Computer Science and Applications (IJCSA)*, vol. 9, no. 1, 2019.

[15] W. Vogels, "Eventually Consistent," *Communications of the ACM*, vol. 52, no. 1, 2009.

[16] M. Ali et al., "Security Challenges in Hybrid Applications," *IEEE Access*, vol. 8, 2020.

[17] Griebel, L., Prokosch, HU., Köpcke, F. et al. A scoping review of cloud computing in healthcare. BMC Med Inform Decis Mak 15, 17 (2015). https://doi.org/10.1186/s12911-015-0145-7

[18] S. Sachdeva, "Unraveling the role of cloud computing in health care system," *NLM-PubMed Central*, 2024. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11004887/

[19] R. Sivan, "Security and privacy in cloud-based e-health systems," *Symmetry*, vol. 13, no. 5, 742, 2021. https://www.mdpi.com/2073-8994/13/5/742

[20] M. Abughazalah, "Centralized vs. decentralized cloud computing in healthcare systems," *Applied Sciences*, vol. 14, no. 17, 7765, 2024. https://www.mdpi.com/2076-3417/14/17/7765

[21] W. E. Kedi, C. Ejimuda, and M. D. Ajegbile, "Cloud computing in healthcare: A comprehensive review", *World Journal of Advanced Engineering Technology and Sciences*, vol. 12, no. 2, pp. 290-298, 2024.

[22] A. Atanda, "Cloud computing in the healthcare industry: a systematic literature review ", *Global Journal of Information Technology: Emerging Technologies*, vol. 13, no. 2, pp. 64-71, 2023.

[23] S. Wang, X. Zhang, Y. Zhang, and L. Wang, "An Edge– Cloud Integrated Framework for Flexible and Efficient Hybrid Stream Analytics," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6323–6336, 2022. https://doi.org/10.1109/JIOT.2021.3118294

[24] The KubeEdge Authors, "KubeEdge: Kubernetes Native Edge Computing Framework," Online documentation, 2023. Available: https://kubeedge.io

[25] D. C. Nguyen, M. Ding, P. N. Pathirana, and A. Seneviratne, "Federated Learning in Healthcare: A Survey," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1-37, 2022. https://doi.org/10.1145/3488248

[26] C. Butpheng, K.-H. Yeh, and X. Xiong, "Security and Privacy in IoT–Cloud-Based e-Health Systems: A Comprehensive Review," *Symmetry*, vol. 12, no. 7, Art. no. 1191, 2020. https://doi.org/10.3390/sym12071191