VIVA-Tech International Journal for Research and Innovation *Volume 1, Issue 9 (2026)*
ISSN(Online): 2581-7280 *PP 01-04*

VIVA-TECH IJRI
ANNUAL RESEARCH JOURNAL
VIVA-TECH INTERNATIONAL JOURNAL FOR RESEARCH AND INNOVATION
ANNUAL RESEARCH JOURNAL
ISSN(ONLINE): 2581-7280

Serverless Architecture for Real-Time Data Processing in Smart Applications

Dr. Pradnya Mhatre[1], Ajay Bhivsane[2], Sumit Kadam[3]

[1](*Department of Master of Computer Application, Viva Institute of Technology, University of Mumbai, India)*
[2] *(Department of Master of Computer Application, Viva Institute of Technology, University of Mumbai, India)*
[3] *(Department of Master of Computer Application, Viva Institute of Technology, University of Mumbai, India)*

Abstract : *The rapid growth of smart applications such as Internet of Things (IoT), smart cities, healthcare monitoring, and real-time analytics systems has significantly increased the demand for scalable and low-latency data processing architectures. Traditional server-based systems often struggle to meet these requirements due to infrastructure management overhead and limited scalability. Serverless architecture has emerged as a promising solution by enabling event-driven, on-demand execution of computing resources without server management. This paper explores the role of serverless architecture in real-time data processing for smart applications. It discusses core components, system architecture, advantages, challenges, and practical use cases. The study demonstrates how serverless platforms enhance scalability, reduce operational costs, and improve responsiveness in real-time smart systems.*

Keyword: *Event-driven computing, Internet of Things, Real-time data processing, Scalability, Serverless architecture*

1. Introduction

Smart applications rely heavily on real-time data generated from sensors, devices, and user interactions. Applications such as smart healthcare systems, intelligent transportation, smart grids, and industrial automation continuously produce high- velocity data streams. Processing such data in real time requires architectures that are scalable, reliable, and cost-efficient.

Traditional monolithic and server-based architectures demand constant resource provisioning and maintenance, which limits their efficiency in dynamic environments.

Serverless computing addresses these challenges by abstracting infrastructure management from developers. In a serverless model, developers focus only on writing application logic, while cloud providers manage scaling, availability, and execution. This makes serverless architecture particularly suitable for real- time data processing where workloads are unpredictable and event-driven.

2. Problem Statement

Real-time data processing in smart applications presents several challenges, including handling massive data volumes, maintaining low latency, and ensuring system scalability. Traditional server-based systems often suffer from resource underutilization, delayed scaling, and high operational costs. Additionally, managing infrastructure complexity becomes difficult as applications grow. There is a need for an architecture that can efficiently process real- time data while minimizing infrastructure management and costs.

1) Objectives:

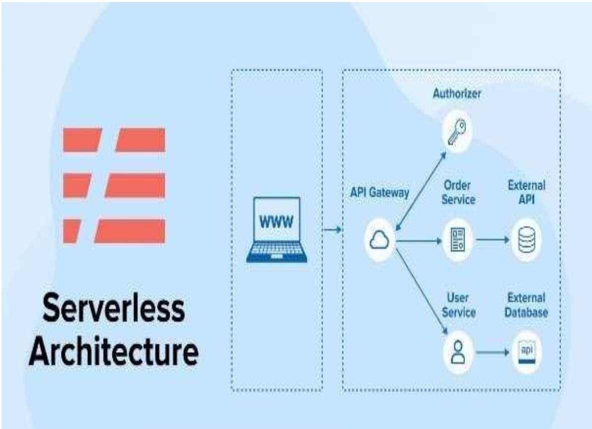
The main objectives of this research are:

- To study the fundamentals of serverless architecture
- To analyze real-time data processing requirements in smart applications
- To design a serverless-based system architecture for real-time processing
- To evaluate the benefits and challenges of serverless platforms
- To identify key application areas where serverless architecture is effective

2) Literature Review:

Recent research highlights the growing adoption of serverless computing for real- time and data-intensive applications. Studies indicate that serverless platforms provide automatic scalability and cost optimization by charging only for execution time.
Researchers have explored serverless models in IoT systems, where event-driven functions process sensor data efficiently.
Several studies emphasize the integration of serverless architecture with stream processing frameworks such as Apache Kafka and cloud-native messaging services. While serverless computing improves scalability, researchers also point out limitations such as cold start latency and execution time constraints. Overall, existing literature supports serverless architecture as a viable solution for real-time smart applications, provided its limitations are addressed.

3) Serverless System Architecture:

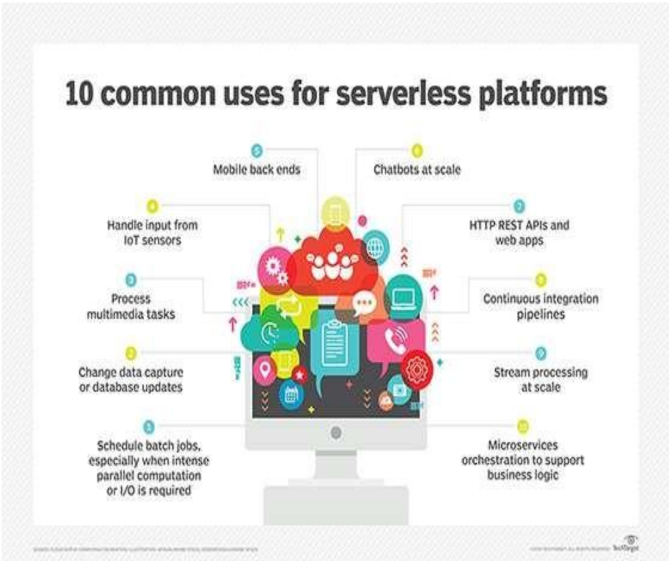


A typical server-less architecture for real- time data processing consists of multiple layers. The data source layer includes IoT sensors, mobile devices, and smart systems that generate continuous data streams. The ingestion layer uses event brokers or API gateways to capture incoming data. The processing layer consists of serverless functions that execute business logic in response to events. Finally, the storage and visualization layer stores processed data and presents insights through dashboards and applications.

3. Research Methodology

This study follows a qualitative and architectural analysis approach. Existing server-less platforms and real-time processing models are analyzed through literature review and system design comparison. The architecture is evaluated based on scalability, latency, cost efficiency, and ease of deployment. Use cases from smart applications are examined to validate the effectiveness of the proposed approach.

4. Applications of Server-less Architecture



Server-less architecture is widely used in smart applications such as smart healthcare, where patient data from wearable devices is processed in real time. In smart cities, serverless platforms handle traffic monitoring and environmental sensing. Other applications include real-time fraud detection, chatbots, media streaming, and IoT-based automation systems.

Challenges and Limitations: Despite its advantages, serverless architecture faces challenges such as cold start latency, limited execution time, and dependency on cloud providers. Debugging and monitoring distributed server-less applications can also be complex. Security and data privacy remain critical concerns, especially in applications handling sensitive real-time data.

5. Conclusion

Serverless architecture offers a powerful and flexible solution for real-time data processing in smart applications. By eliminating server management and enabling event-driven execution, it improves scalability and reduces operational costs. Although challenges exist, ongoing advancements in cloud technologies are addressing these limitations. Serverless computing is expected to play a significant role in the future of real-time smart systems.

Future Scope :
Future research can focus on optimizing cold start performance and integrating artificial intelligence with serverless platforms. The use of edge computing combined with serverless models can further reduce latency. Additionally, real-world deployment studies can provide deeper insights into performance and reliability.

Acknowledgements

We would like to express our sincere gratitude to Dr. Pradnya Mhatre for her valuable guidance, continuous support, and encouragement throughout the completion of this research work. Her expert advice and constructive suggestions greatly contributed to improving the quality and direction of our study.

We are deeply thankful to Viva Institute of Technology for providing us with the necessary academic environment, facilities, and resources to successfully carry out this research. The support and cooperation of the faculty members of the Master of Computer Application department have been highly appreciated.

We also extend our heartfelt thanks to University of Mumbai for giving us the opportunity to undertake this research as a part of our academic curriculum. Finally, we express our gratitude to our family and friends for their constant motivation, encouragement, and support throughout this work.

References

[1] Amazon Web Services, *Serverless Computing – Overview*, AWS Documentation, 2024.
[2] Google Cloud, *Cloud Functions: Event- Driven Serverless Compute Platform*, Google Cloud Docs, 2024.
[3] J. Spillner, T. Mateos, and D. Bermbach, "Faaster, better, cheaper: The rise of serverless computing," *IEEE Internet Computing*, vol. 21, no. 4, pp. 20–28, 2017.
[4] E. Jonas et al., "Occupy the cloud: Distributed computing for the 99%," *Proceedings of the ACM Symposium on Cloud Computing*, pp. 445–451, 2017.
[5] W. Shi et al., "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
[6] M. Chen, Y. Zhang, Y. Li, S. Mao, and V. C. M. Leung, "Big data analytics for IoT and smart cities," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 128–136, 2018.
[7] I. Baldini et al., "Serverless computing: Current trends and open problems," *Research Advances in Cloud Computing*, Springer, pp. 1–20, 2017.
[8] L. Wang et al., "Peeking behind the curtains of serverless platforms," *USENIX Annual Technical Conference*, pp. 133–145, 2018.
[9] A. Singh and M. M. Singh, "Real-time data processing using cloud-based serverless architecture," *International Journal of Computer Applications*, vol. 182, no. 44, pp. 1–6, 2019.
[10] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, "A preliminary review of enterprise serverless cloud computing," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1–14, 2017.
[11] Z. Zhang et al., "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.
[12] M. Adzic and V. Chatley, "Serverless computing: Economic and architectural impact," *Proceedings of the 11th Joint Meeting on Foundations of Software Engineering*, pp. 884–889, 2017.
[13] Apache Software Foundation, *Apache Kafka Documentation*, 2024.
[14] S. Hendrickson et al., "Serverless computation with open Lambda," *Proceedings of the 8th USENIX Workshop on Hot Topics in Cloud Computing*, 2016.