JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Review on
Synergies in Modern Architecture: Microservices with Serverless / Cloud and Edge Computing

Hari Prasad Dola
MTech Research scholar
Department of CSE, DPGITM, India

Dr. Mukesh Yadav
Professor
Department of CSE, DPG ITM, India

Ms. Archana Rohilla
Assistant professor
Department of CSE, DPGITM, India

ABSTRACT
Integrating microservices, Serverless computing, cloud computing, and edge computing represents a profound transformation in the landscape of software architecture, enabling systems that are not only highly scalable and responsive but also adaptable to the varying demands of modern applications. This review paper explores the synergies between these paradigms, thoroughly analyzing each technology's core principles, advantages, and the challenges involved in their integration. The aim is to study how combining these technologies can create a distributed computing framework that meets the requirements of real-time, resource-efficient, and scalable systems in diverse domains, from IoT to mobile applications. The review further introduces a Geo-Tracking System prototype that exemplifies how each component contributes to a cohesive, resilient system. This work underscores the potential of these technologies to drive innovation in software development and deployment, supporting an evolving digital landscape. Finally, the review outlines future research areas and advancements that could simplify management and enhance security in such hybrid architectures.

Keywords
Synergies in Modern Architectures, Microservices, Cloud Architecture, Cloud Computing, Serverless, Edge Computing, Distributed Systems, Scalability, Real-Time Processing, Modularity, Hybrid Architectures.

1. INRODUCTION

Over the past two decades, software architecture has evolved from monolithic structures to modular and distributed paradigms. Cloud computing emerged as a game-changer, enabling companies to access vast resources on demand. The introduction of microservices architecture allowed developers to build applications as collections of smaller, independently deployable services. Serverless computing, meanwhile, abstracted infrastructure management, letting developers focus on coding and innovation rather than server maintenance. Edge computing extended the architecture by processing data closer to the user, addressing latency issues critical in IoT and other real-time applications. This review paper delves into these paradigms individually, discussing their unique benefits, and then explores the synergies that arise from their integration. By merging the strengths of these models, developers can design systems that are not only powerful and resilient but also cost-efficient and agile.

2. BASIC CONCEPTS

In this section, we cover each paradigm's current state and the latest advancements that enable their combined use:

2.2 Cloud Computing

Initially introduced through virtualization, cloud computing offers scalable, on-demand resources that can be tailored to diverse application needs. Major players like AWS, Google Cloud, and Microsoft Azure provide varied service models—Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each model offers unique levels of control and flexibility, making cloud computing a foundation for scalable applications.

2.2.1 Serverless Computing

As a cloud computing model, Serverless abstracts server management entirely, enabling developers to deploy functions that run only when triggered by specific events (e.g., HTTP requests, database updates). Popular platforms include AWS Lambda, Google Cloud Functions, and Azure Functions. Serverless computing is highly cost-effective due to its pay-as-you-go billing and is ideal for event-driven applications and microservices.

2.2.2 Edge Computing

Unlike cloud computing, which centralizes data processing, edge computing processes data near its source. This reduces latency and bandwidth use, making it essential for real-time applications, particularly in IoT. Edge computing is seeing rapid adoption across industries, from autonomous vehicles to healthcare, where real-time data processing is critical.

2.2.3 Microservices Architecture

Microservices structures applications as a suite of loosely coupled, independently deployable services. Each service typically manages its own database and performs specific functions, promoting fault isolation, scalability, and agility. Microservices are ideal for complex applications requiring modularity and can be deployed using various programming languages, frameworks, or platforms.

3. FUNCTIONALITY DESCRIPTION

The proposed architecture integrates microservices, Serverless computing, cloud infrastructure, and edge computing. Each technology brings unique advantages to the architecture, supporting a flexible, scalable, and resilient distributed system. Below, each component is discussed with greater detail on its role, integration, and benefits within the system.

3.1 Microservices Architecture[10][12]

Microservices architecture structures the system into smaller, self-contained services, each responsible for a specific functionality. This modular approach has numerous benefits for development, deployment, and maintenance:

3.1.1 Service Modularity and Independence

Each microservice addresses a distinct functionality—such as user authentication, data storage, or analytics—which enables easy management and modular upgrades. By separating functionalities, the architecture also supports fault tolerance, as issues in one service do not affect others.

3.1.2 Scalability and Load Management

Microservices can be independently scaled based on demand. For instance, a service handling real-time data processing can be scaled up independently during peak times, while less critical services can remain unchanged. This flexibility enhances the system’s resource efficiency.

3.1.3 Inter-Service Communication

Communication between microservices is achieved through lightweight REST APIs, which are efficient and ensure broad compatibility across platforms. Using APIs enables a loosely coupled system, which simplifies testing, updating, and redeployment of individual services without impacting the entire application.

3.2 Serverless Computing

Serverless computing brings an event-driven execution model to the architecture, where functions are triggered by specific events rather than being always active. Serverless functions, particularly AWS Lambda, play a key role in enabling responsive and cost-efficient operations[11][13]:

3.2.1 Event-Driven Execution

Serverless functions are activated in response to predefined events, such as an API request or a new data upload. This model reduces idle resource consumption, as compute resources are only used when required, making it both efficient and cost-effective.

3.2.2 Auto-Scaling Based on Demand

The Serverless model automatically scales functions based on demand. For instance, during a high influx of user interactions or data uploads, AWS Lambda automatically deploys additional function instances to handle the load. This flexibility is essential for applications with fluctuating or unpredictable workloads.

3.2.3 Use Cases in the System

Within this architecture, Serverless functions are designed to handle data processing tasks, real-time analytics, and background operations. For example, data from edge devices (e.g., GPS-enabled mobile devices) can be processed in real-time, with only relevant information forwarded to the cloud for further analysis or storage.

3.3 Cloud Infrastructure

The cloud infrastructure acts as the backbone of this architecture, providing centralized management, processing power, and storage capabilities. AWS is used in this system to coordinate and support microservices and Serverless functions:

3.3.1 Centralized Resource Management

The cloud serves as the control center for managing microservices, data storage, and Serverless functions. By leveraging cloud resources, the system can handle complex computational tasks, long-term data storage, and global accessibility, ensuring a cohesive operation.

3.3.2 High Availability and Reliability

Cloud providers like AWS offer robust redundancy and failover mechanisms, ensuring the system remains operational even in cases of hardware or network failures. This reliability is essential for applications with real-time requirements or that are deployed across regions.

3.3.3 Data Storage and Accessibility

The cloud provides scalable storage for structured and unstructured data, enabling secure and flexible data handling. Centralized data storage also allows for easy data sharing and synchronization across microservices and Serverless functions.

3.4 Edge Computing

Edge computing extends computational capabilities to the network edge, enabling data processing close to the data source. This architecture leverages edge devices, like mobile GPS sensors, to collect and process data with minimal latency:

3.4.1 Local Data Processing for Reduced Latency

Edge computing allows devices to process data locally, minimizing latency by avoiding data transfer to a central data centre for initial processing. For example, in a Geo-Tracking System, GPS-enabled mobile devices can process location data in real-time before sending summarized data to the cloud.

3.4.2 Bandwidth Optimization

By processing data at the edge, the system reduces the need to transfer large volumes of raw data to the cloud. Only essential or aggregated data is transmitted, saving bandwidth and reducing network congestion—an essential feature for applications with numerous edge devices.

3.4.3 Real-Time Responsiveness

Edge devices support applications requiring immediate responses [19], such as location tracking, IoT device monitoring, or time-sensitive notifications. In this architecture, edge devices handle time-sensitive tasks while the cloud manages data that requires deeper analysis or storage.

DATA FLOW AND SYSTEM COMMUNICATION

The architecture's data flow is carefully designed to balance the strengths of edge, cloud, and Serverless components, ensuring efficient, reliable, and low-latency data handling:

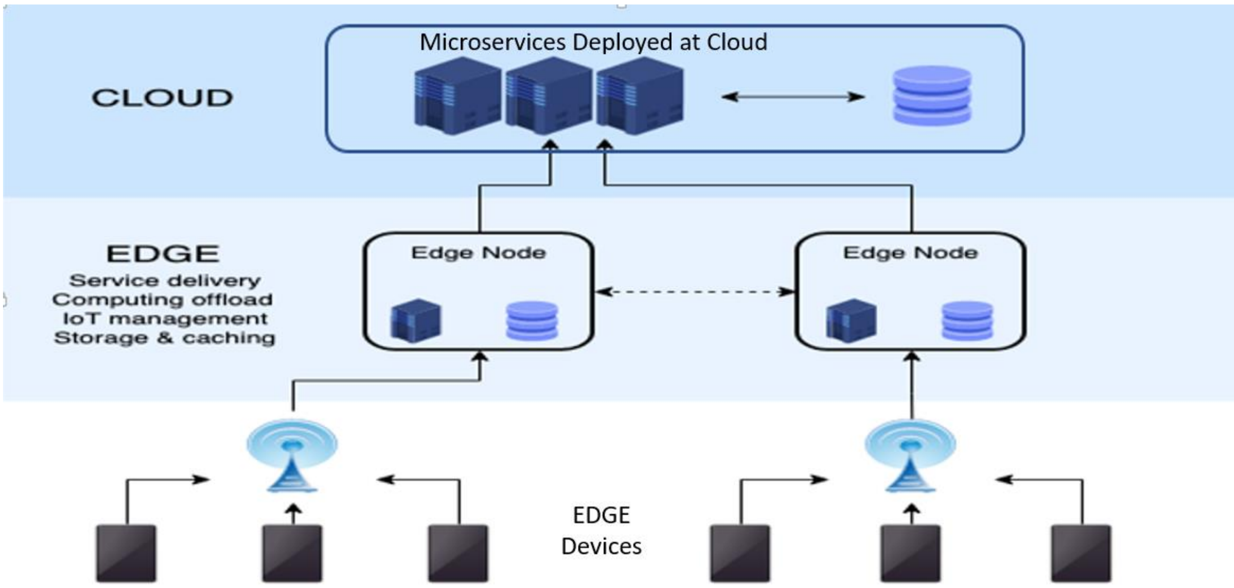

Fig 4.1: An Example of Data flow from an Edge Device to Cloud Environment via Microservices

4.1 Data Collection at the Edge

Edge devices (e.g., mobile devices) serve as the primary data collectors, capturing real-time information such as GPS coordinates. These devices handle preliminary data processing to ensure only relevant data is transmitted to the cloud.

4.2 Event-Driven Data Processing in Serverless

Serverless functions handle real-time data processing and analysis based on specific triggers, such as a new data packet from an edge device. These functions analyse data rapidly, responding to events without requiring always-on server resources.

4.3 Centralized Storage and Orchestration in the Cloud[24]

Processed data is sent to the cloud for storage, deeper analysis, or cross-regional synchronization. The cloud acts as the orchestrator, ensuring smooth inter-service communication and efficient data distribution across microservices.

5. SECURITY AND COMPLIANCE CONSIDERATIONS

In a distributed and hybrid architecture like this, ensuring security [17] and compliance is essential. Security protocols are built into every layer of the architecture

5.1 Data Encryption

Data is encrypted both at rest (in cloud storage) and in transit (between edge devices, Serverless functions, and the cloud) to safeguard against unauthorized access [20].

5.2 Access Control

Role-based access control (RBAC) [21] is implemented, ensuring that only authorized services or users can access specific data or functions. This is particularly important when sensitive data is handled or transmitted.

5.3 Compliance with Industry Standards

The architecture adheres to industry standards, especially in sectors such as healthcare or finance, where data privacy and regulatory compliance (e.g., GDPR) [18] are critical. Regular compliance audits and security testing are integrated into the development lifecycle.

6. SYSTEM PROTOTYPE: GEO-TRACKING SYSTEM USE CASE

The architecture's design and integration are validated through a real-world prototype - a Geo-Tracking System. This system captures, processes, and stores geolocation data collected from mobile devices, demonstrating how each component interacts and supports the overall system

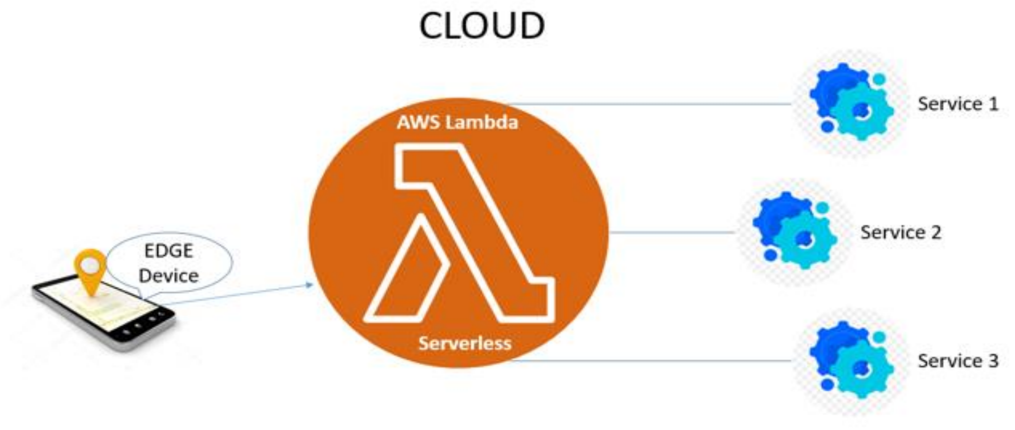

Fig 6.1: Geo-Tracking System Architecture

6.1 Edge Device Functionality

Mobile devices with GPS sensors collect location data and handle initial data processing tasks, such as filtering duplicate or irrelevant data points.

6.2 Serverless Processing

AWS Lambda functions process the geolocation data in real-time, running analytical tasks that may include calculating movement patterns or sending location-based notifications.

6.3 Cloud Coordination and Data Management

The cloud manages data storage and user interaction, allowing users to view their location history, configure tracking preferences, or receive alerts.

A detailed comparative analysis with Edge and Cloud Computing, highlighting each architecture's strengths and addressing various challenges in distributed system design.

Aspect	**Microservices with Edge**	**Microservices with Cloud**	**Benefits of Merging Cloud and Edge**
Latency	Low latency due to localized processing, essential for real-time IoT applications [19] (*S. T., et al., 2023. "Analyzing The Effect of Edge Computing on Real-Time Data Processing and Latency Reduction," IEEE UPCON*)	Centralized data processing can lead to higher latency, though cloud providers have optimized regions to reduce this [12] (*Ariel Ortiz, 2019. " Architecting Serverless Microservices on the Cloud with AWS," ACM SIGCSE*)	Reduced latency by processing time-sensitive data at the edge while leveraging the cloud for non-critical tasks, optimizing overall response times and improving the user experience, especially for real-time applications.
Scalability	Edge devices have limited scaling potential, but are suitable for task-specific processing in a modular setup (*Z. Stojanov [15], et al.,*	Cloud-based microservices scale effectively across vast infrastructures, leveraging on-demand resources [14] (*R. Xie, et*	Scalable processing: Edge handles real-time tasks, and cloud provides virtually unlimited scalability for data storage and processing,

	2023. "A Tertiary Study on Microservices: Research Trends and Recommendations," Elsevier)	*al., 2021. "When Serverless Computing Meets Edge Computing," IEEE Wireless Communications*)	enabling high availability and performance during demand peaks.
Data Processing	Localized, which reduces data transfer and conserves bandwidth [11] (*Shweta Khare et al., 2019. " Minimizing the makespan for edge-based stream processing of directed acyclic graphs," ACM/IEEE SEC*)	Centralized in the cloud, ideal for resource-intensive processing and complex analysis [16] (*R. Lichtenthäler, G. Wirtz, 2024. "Formulating a quality model for cloud-native software architectures," Springer*)	Bandwidth optimization: Edge devices preprocess and filter data, transmitting only essential information to the cloud, which performs deep analysis and long-term storage, reducing bandwidth use and cloud storage costs.
Fault Tolerance	Fault tolerance depends on the robustness of individual edge devices, making high resilience challenging [14] (*R. Xie, et al., 2021. "When Serverless Computing Meets Edge Computing," IEEE Wireless Communications*)	Cloud providers offer high fault tolerance with built-in redundancy and failover mechanisms, suitable for critical applications [24] (*H. Y. Shwe, et al., 2016. "Scalable Distributed Cloud Data Storage for IoT," IEEE*)	Enhanced resilience: Edge devices provide local continuity during cloud outages, and cloud ensures data backup, redundancy, and recovery, creating a fail-safe distributed system with minimal service interruptions.
Maintenance & Upgrades	Modular edge deployment simplifies updates for local processing units, though distributed updates can be complex *[17]* (*P. Antunes, et al., 2024. "Guiding the implementation of data privacy with microservices," Springer*)	Centralized cloud-based maintenance makes version control and updates easier to manage across services [23] (*J. Zhao, N. Xiao, 2008. "Framework of Scalable High Performance Multi-layer Systems," IEEE*	Centralized control with localized adaptability: Cloud orchestrates system-wide updates, while edge devices can implement localized updates with minimal downtime, ensuring consistent performance and functionality across regions.
Security & Compliance	Edge systems require device-level encryption and API security for data privacy, essential for compliance in IoT applications [18] (*Y.-S. Martin, A. Kung, 2018. "Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering," IEEE EuroS&PW*)	Cloud providers offer mature security protocols and regular compliance checks, ideal for applications handling sensitive data [17] (*Antunes, et al., 2024. "Data privacy with microservices," Springer*)Data Privacy	Improved security and compliance: Sensitive data is processed locally at the edge to minimize exposure, while the cloud provides robust data encryption, access control, and compliance with industry standards (e.g., GDPR, HIPAA).

7. CONCLUSION AND SCOPE OF IMPROVEMENT

The integration of microservices, Serverless, cloud, and edge computing technologies represents a robust approach to modern distributed systems. The combined architecture is well-suited for applications requiring scalability, flexibility, and real-time processing. However, managing such a hybrid architecture can be complex, as it demands sophisticated orchestration, monitoring, and security measures.

For future work, the scope of improvements include:

7.1 Orchestration Improvements

Developing and integrating advanced tools for seamless orchestration of cloud, edge, and Serverless resources.

7.2 Enhanced Security Protocols

Addressing security concerns inherent in distributed architectures by implementing encryption, secure access management, and monitoring tools.

7.3 Performance Optimization

Identifying ways to further reduce latency and optimize resource allocation across microservices, Serverless functions, and edge devices.

7.4 Mitigating Cold Start Issues With Serverless

Cold start issues [22] in Serverless computing are a common challenge, particularly in environments like AWS Lambda, Google Cloud Functions, and Azure Functions. A "cold start" occurs when a Serverless function is invoked without any pre-warmed instances available, leading to increased latency as the function's infrastructure is initialized.

7.5 Real-World Case Studies

Conducting more case studies across different sectors (e.g., healthcare, finance) to validate and refine this architecture's efficacy.

The future of distributed computing will likely see further integration of these paradigms, with enhanced support from emerging technologies such as AI-driven resource management and more advanced orchestration frameworks. This seminar highlights the transformative potential of these architectures and encourages further exploration into their combined application.

6 REFERENCES

6.1 Books referred:

[1] "Designing Data-Intensive Applications" by Martin Kleppmann

[2] "Building Microservices" by Sam Newman

[3] "Serverless Architectures on AWS" by Peter Sbarski

[4] "Cloud Native Patterns" by Cornelia Davis

6.2 Websites referred:

[5] https://scholar.google.com/

[6] https://www.researchgate.net/

[7] https://arxiv.org/

[8] Medium.org

[9] Dev.to

6.3 Research Papers and Journals referred:

[10] Hasan M.H., Osman M.H., Admodisastro N.I., Muhammad M.S., "Legacy systems to cloud migration: A review from the architectural perspective" in Journal of Systems and Software by Elsevier 202, 2023

[11] Shweta Khare, Hongyang Sun, Julien Gascon-Samson, Kaiwen Zhang, Aniruddha Gokhale, Yogesh Barve, Anirban Bhattacharjee, and Xenofon Koutsoukos. 2019. Linearize, predict and place: minimizing the makespan for edge-based stream processing of directed acyclic graphs. In Proceedings of the 4th ACM/IEEE Symposium on Edge Computing

[12] Ariel Ortiz. 2019. Architecting Serverless Microservices on the Cloud with AWS. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 1240.

[13] P. Raith, S. Nastic and S. Dustdar, "Serverless Edge Computing—Where We Are and What Lies Ahead," in IEEE Internet Computing, *vol. 27, no. 3, pp. 50-64, May-June 2023*

[14] R. Xie, Q. Tang, S. Qiao, H. Zhu, F. R. Yu and T. Huang, "When Serverless Computing Meets Edge Computing: Architecture, Challenges, and Open Issues," in IEEE Wireless Communications, *vol. 28, no. 5, pp. 126-133, October 2021*

[15] Stojanov Z., Hristoski I., Stojanov J., Stojkov A. "A Tertiary Study on Microservices: Research Trends and Recommendations" in Programming and Computer Software by Elsevier 49(8), pp.796-821 2023

[16] Lichtenthäler, R., Wirtz, G. Formulating a quality model for cloud-native software architectures: conceptual and methodological considerations. Cluster Comput ***27****, 4077–4093 (2024) in Springer.*

[17] Antunes, P., Guimarães, N. Guiding the implementation of data privacy with microservices. Int. J. Inf. Secur. ***23****, 3591–3608 (2024)in Springer*

[18] Y. -S. Martin and A. Kung, "Methods and Tools for GDPR Compliance Through Privacy and Data Protection Engineering," 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), London, UK, 2018

[19] S. T, A. Dutt, V. Veeraiah, B. Aishwarya, K. Lal and D. Kapila, "Analyzing The Effect of Edge Computing on Real-Time Data Processing and Latency Reduction," 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gautam Buddha Nagar, India, 2023.

[20] X. Zhang, H. Hu and Y. Xu, "The Design and Implementation of an Automatic Encryption and Decryption System for Sensitive Words in Databases Based on Spring Boot," 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 2024

[21] Y. Yuan, X. Yi and J. ShangGuan, "Designing Data Permissions in the Enterprise Application Environment," 2023 4th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 2023

[22] Hanfei Yu, Rohan Basu Roy, Christian Fontenot, Devesh Tiwari, Jian Li, Hong Zhang, Hao Wang, and Seung-Jong Park. 2024. RainbowCake: Mitigating Cold-starts in Serverless with Layer-wise Container Caching and Sharing. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1 (ASPLOS '24), Vol. 1. Association for Computing Machinery

[23] J. Zhao and N. Xiao, "Study on Framework of Scalable High Performance Multi-layer and Multi-pattern Information Systems," 2008 International Symposiums on Information Processing, Moscow, Russia, 2008

[24] H. Y. Shwe and P. H. J. Chong, "Scalable Distributed Cloud Data Storage Service for Internet of Things," 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 2016