International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Systematic review of microservice architecture: Advantages over monolithic systems in cloud environments

Erik Ghazaryan *

Senior Software Developer at Nagarro, Vienna, Austria.

International Journal of Science and Research Archive, 2025, 15(03), 906-911

Publication history: Received on 05 May 2025; revised on 12 June 2025; accepted on 14 June 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.15.3.1826

Abstract

This paper examines the characteristics of microservice architecture, highlighting its benefits compared with monolithic systems in cloud environments. The methodology is based on selecting relevant peer-reviewed articles from leading scientific repositories and industry reports. The results demonstrate that microservices deliver high scalability through horizontal scaling of individual components, reduce downtime by enabling isolated service recovery, and accelerate CI/CD workflows via parallel development and independent module releases. Moreover, the adoption of containerization and orchestration technologies (Docker, Kubernetes, etc.) enhances technological adaptability. At the same time, the analysis identifies several challenges: increased complexity of inter-service communication, the need for advanced observability systems, and more complicated configuration management. To address these risks, strategies are proposed ranging from unified API gateways and service meshes to automated request tracing and centralized monitoring. In conclusion, although the initial costs of designing and deploying a microservice architecture are significant, its use in the cloud lays the foundation for sustainable growth, greater innovation, and rapid response to evolving business requirements. The findings will interest software architects, CTOs, and researchers in cloud computing and software engineering.

Keywords: Microservice Architecture; Monolithic Systems; Scalability; Fault Tolerance; Containerization; Orchestration; CI/CD; Service Meshes

1. Introduction

The relevance of studying architectural strategies for software systems in cloud environments stems from the rapid growth of data volumes and the stringent demands placed on service performance and reliability [1]. Historically, the monolithic model where all system components are tightly coupled within a single codebase and delivered as one artifact has exhibited serious limitations in terms of horizontal scalability, fault tolerance, and deployment flexibility in the cloud [2]. In contrast, the microservice paradigm divides an application into a collection of autonomous services, each with its own development and deployment lifecycle [3].

The aim of this paper is to analyze the impact of microservices on scalability, fault tolerance, development velocity, and release time to market. The scientific novelty lies in combining empirical data and theoretical models to characterize the effects of cloud-based microservice adoption and to identify the key determinants of migration success.

The central hypothesis holds that, given effective management of inter-service communication and standardized CI/CD processes, transitioning to a microservice architecture yields statistically significant improvements in these metrics compared with traditional monolithic systems.

* Corresponding author: Erik Ghazaryan.

2. Materials and Methods

A comprehensive analysis of literature and industry reports from the past four years was conducted, focusing on studies that demonstrate the advantages of microservice architecture over monolithic solutions in cloud infrastructures. The available sources can be provisionally divided into several thematic groups.

The first group of works addresses the market context and foundational principles of microservice architecture. According to Statista [1], Gartner [14], Flexera [15], Grand View Research [16], and PR Newswire [17], public-cloud spending grows annually, driving the shift to microservices as an efficient way to leverage cloud resources. In his seminal guide, Newman [2] highlights the lack of clear recommendations for decomposing monoliths and formulates six key principles for fine-grained service design. Systematic reviews by Velepucha and Flores [3] and by Söylemez, Tekinerdogan, and Kolukısa Tarhan [12] aim to map the fragmented landscape of migration approaches, classify principles, patterns, and challenges, and offer a thematic "problem–solution" synthesis.

The second group includes works on domain-driven design and deployment patterns. Zhong et al. [7] conduct an empirical study of applying DDD to microservices, introducing observability metrics for modularity assessment. Aksakalli et al. [8] systematize communication patterns (pub/sub and point-to-point), substantiating their impact on service manageability. Tang et al. [10] develop a cost-aware deployment strategy for IoT microservices on multi-access edge computing (MEC), demonstrating reduced latency and operational costs. Laso et al. [13] propose the Human-as-a-Service concept, formalizing humans as service actors within the architecture.

The third group covers quality assessment, performance, load forecasting, and security. Tapia and Gaona [4] create a taxonomy of quality attributes and corresponding metrics. Blinowski, Ojdowska, and Przybyłek [9] perform load-testing experiments on AWS/ECS to compare monolithic and microservice deployments. Ramamoorthi [5] and Luo et al. [6] investigate machine-learning models for anomaly detection. Rahaman et al. [11] systematize static analysis methods for cloud-system security.

Thus, despite the breadth of research, contradictions and gaps remain. For example, Blinowski et al. [9] demonstrate the scalability advantages of microservices, whereas Newman [2] emphasizes the communication-overhead costs. There is a shortage of studies that integrate ML-based load forecasting with security and cost-management considerations in real cloud environments, and the application of DDD metrics does not align well with operational KPIs and CI/CD processes.

3. Results and Discussion

In classical monolithic solutions, scaling always impacts the entire application, even when the peak load affects only a narrow set of functions, which inevitably leads to inefficient use of computational resources and inflated cloud expenditures [2]. In contrast, microservices allow capacity to be scaled autonomously and independently for precisely those services experiencing elevated load. This "granular" scaling model minimizes downtime of other subsystems and reduces inter-service latency, while optimizing financial outlay on cloud infrastructure.

Implementing the Circuit Breaker (circuit interruption under overload) and Bulkhead (resource partitioning and isolation) patterns enables fault containment and prevents error propagation across the platform [4]. Such incident isolation ensures that critical functions remain available even during partial system degradation.

Figure 1 presents a schematic of flow segmentation and dependency constraints in a microservice architecture, showing how architectural barriers prevent a failure in one service from destabilizing the entire ecosystem.

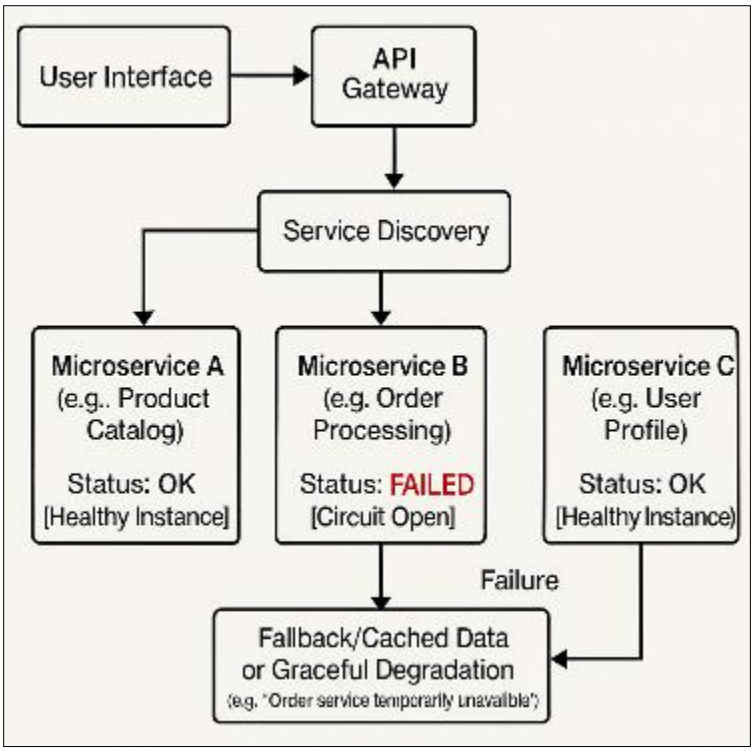

Figure 1 Fault isolation in the micro-service architecture [4]

By forming small, autonomous teams each dedicated to a specific service and free to choose the optimal technology stack (polyglot and heterogeneous by design) organizations can significantly accelerate the time-to-market for new features and simplify continuous integration and delivery (CI/CD) processes [7]. Deployment and updates for each microservice occur independently, drastically reducing the risk of failures associated with large monolithic releases. Adopting a microservice architecture enables organizations to cut the average lead time for changes [8]. A detailed comparison of key parameters is presented in Table 1.

Table 1 Comparison of monolithic and microservice architectures in cloud environments [2, 3, 10, 12]

Characteristic	**Monolithic Architecture**	**Microservice Architecture**
Scalability	Predominantly vertical; scaling the entire unit	Horizontal; independent scaling of individual services
Fault tolerance	Low; component failure leads to total system outage	High; failure isolation, graceful degradation of functionality
Release velocity	Low; lengthy release cycles	High; independent, frequent service deployments
Technology stack	Uniform across the application	Freedom to use diverse technologies for different services
Development complexity	Initially lower, but grows with the codebase size	Initially higher, but manageable at the service level
Operational complexity	Lower (single unit to manage)	Higher (multiple services and inter-service communication)
Infrastructure cost	Potentially higher due to inefficient scaling	Potentially lower through granular resource utilization
Maintenance and evolution	Challenging with large codebases and tight coupling	Easier for individual services due to lose coupling

Despite the undeniable appeal of microservice architectures, their decomposition into autonomous components complicates operational processes: It introduces the need for reliable request routing, management of service contracts and inter-service communication, and maintaining distributed data consistency under strict latency and high-load requirements [9, 14].

For a successful transition to microservices, a mature DevOps culture is critical, encompassing the implementation of automated continuous integration and delivery (CI/CD) pipelines, extensive contract testing and load testing, container deployment via Docker, and cluster management with Kubernetes; of equal importance are centralized logging, metrics collection and distributed transaction tracing, as well as the use of service meshes for flexible routing, load balancing and enforcement of security policies [6, 16].

For organizations with traditional business models, such as Kizlyar Supreme, a phased migration approach is optimal: microservices are first introduced into the most critical subsystems e-commerce platforms, CRM or analytics modules preserving the stability of core operational processes and laying a solid foundation for scalability and rapid market entry [15, 17].

The decision to migrate to a microservice architecture should be based on a comprehensive analysis of key parameters: business value, reflected in accelerated feature delivery and flexible scaling; the technical readiness of the team for cloud technologies, containerization and orchestration; and an organizational culture that supports cross-functional product teams and "infrastructure as code."

The diagram in Figure 2 illustrates the key control points: from assessing the current architecture and DevOps maturity to launching a pilot microservice and progressively extending the approach across the entire system.

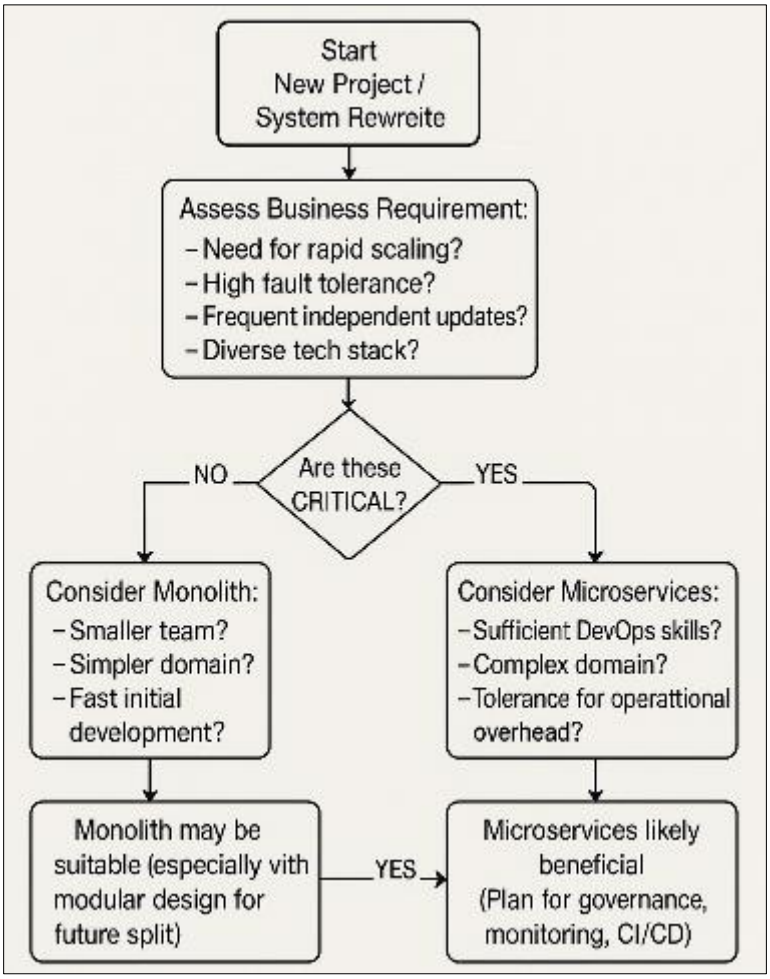

Figure 2 Simplified Decision Diagram: Monolith vs Microservices [5, 6, 11, 13]

Thus, microservice architecture enables autonomous horizontal scaling of individual services, directing additional resources only to overloaded components and thereby reducing redundant cloud infrastructure costs while minimizing downtime and latency in other subsystems. Circuit Breaker and Bulkhead patterns localize failures, preventing them from propagating across the platform, and cloud environments provide automatic monitoring, restart and scaling of instances for rapid recovery and high SLA. At the same time, independent teams—empowered to choose polyglot technology stacks and backed by a mature DevOps culture with continuous integration, delivery and containerization (Docker, Kubernetes) can accelerate feature releases but must address increased complexity in request routing, service-contract management and distributed data consistency, making the adoption of service meshes, centralized logging, tracing and large-scale testing indispensable.

4. Conclusion

The research demonstrates that implementing a microservice architecture within cloud infrastructures delivers enhanced scalability, improved fault tolerance, and substantial flexibility in development, alongside the ability to leverage diverse technology stacks. However, adopting this approach demands meticulous system design, significant capital investment in specialized orchestration and observability tools, and organized training of staff to manage complex distributed services effectively. Despite the higher initial investment and increased operational overhead, a strategic move to microservices is warranted for organizations seeking long-term adaptability, technological leadership, and a competitive edge.

Promising directions for future work include the creation of detailed total cost of ownership (TCO) models for microservice ecosystems and the examination of their impact on the organizational structures of enterprises of various sizes.

References

[1] Statista. Public Cloud – Worldwide [Internet]. Hamburg: Statista GmbH; © 2025. Available from https://www.statista.com/outlook/tmo/public-cloud/worldwide.

[2] Newman S. Building microservices: designing fine-grained systems. Sebastopol (CA): O'Reilly Media, Inc.; 2021. 578 p.

[3] Velepucha V, Flores P. A survey on microservices architecture: principles, patterns and migration challenges. IEEE Access. 2023;11:88339-88358. DOI: 10.1109/ACCESS.2023.3305687.

[4] Tapia V., Gaona M. Research opportunities in microservices quality assessment: a systematic literature review. Journal of Advances in Information Technology. 2023;14(5):991-1002.

[5] Ramamoorthi V. Machine learning models for anomaly detection in microservices. Quarterly Journal of Emerging Technologies and Innovations. 2020;5(1):41-56.

[6] Luo Y. et al. Integrating system state into spatio temporal graph neural network for microservice workload prediction. Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024:5521-5531. DOI: 10.1145/3637528.3671508.

[7] Zhong C. et al. Domain-driven design for microservices: an evidence-based investigation. IEEE Transactions on Software Engineering. 2024. DOI: 10.1109/TSE.2024.3385835.

[8] Aksakalli IK, et al. Deployment and communication patterns in microservice architectures: a systematic literature review. Journal of Systems and Software. 2021;180:111014. DOI: 10.1016/j.jss.2021.111014.

[9] Blinowski G, Ojdowska A, Przybyłek A. Monolithic vs. microservice architecture: a performance and scalability evaluation. IEEE Access. 2022;10:20357-20374. DOI: 10.1109/ACCESS.2022.3152803.

[10] Tang B, et al. Cost-aware deployment of microservices for IoT applications in mobile edge computing environment. IEEE Transactions on Network and Service Management. 2022;20(3):3119-3134. DOI: 10.1109/TNSM.2022.3232503.

[11] Rahaman MS, et al. Static-analysis-based solutions to security challenges in cloud-native systems: systematic mapping study. Sensors. 2023;23(4). DOI:10.3390/s23041755.

[12] Söylemez M, Tekinerdogan B, Kolukısa Tarhan A. Challenges and solution directions of microservice architectures: a systematic literature review. Applied Sciences. 2022;12(11). DOI: 10.3390/app12115507.

[13] Laso S, et al. Human microservices: a framework for turning humans into service providers. Software: Practice and Experience. 2021;51(9):1910-1935. DOI: 10.1002/spe.2976.

[14] Gartner. Gartner forecasts worldwide public cloud end-user spending to reach nearly $600 billion in 2023 [Internet]. Stamford: Gartner; © 2022. Available from https://www.gartner.com/en/newsroom/press-releases/2022-10-31-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-600-billion-in-2023.

[15] Flexera. 2024 State of the Cloud Report [Internet]. McLean: Flexera LLC; © 2024. Available from https://info.flexera.com/CM-REPORT-State-of-the-Cloud.

[16] Grand View Research. Development To Operations Market Size [Internet]. San Francisco: Grand View Research; © 2025. Available from https://www.grandviewresearch.com/industry-analysis/development-to-operations-devops-market.

[17] PR Newswire. Global DevOps Market to Reach $17.8 Billion by 2026 [Internet]. New York: PR Newswire; © 2025. Available from https://www.prnewswire.com/news-releases/global-devops-market-to-reach-17-8-billion-by-2026--301323923.html.