

BUILDING MICROSERVICE ARCHITECTURES: LESSONS FROM DECOUPLING

*Hrishikesh Rajesh Mane¹, Sandhyarani Ganipaneni², Sivaprasad Nadukuru³, Om Goel⁴, Niharika Singh⁵ &
Prof.(Dr.) Arpit Jain⁶*

¹The State University of New York at Binghamton, Binghamton New York, US

²Scholar, Jawaharlal Nehru Technological University, Hyderabad, Telangana, India

³Andhra University, Muniswara Layout, Attur, Yelahanka, Bangalore,

^{4,5}ABES Engineering College Ghaziabad, India

⁶KL University, Vijaywada, Andhra Pradesh, India

ABSTRACT

The transition from monolithic architectures to microservice architectures has become a pivotal shift in software development, driven by the increasing demand for scalability, agility, and resilience in complex applications. Monolithic systems, characterized by tightly coupled components and a singular deployment strategy, often lead to significant challenges, including prolonged deployment cycles, reduced system reliability, and difficulties in scaling individual components. This paper explores the lessons learned from the process of decoupling monolithic systems into microservices, offering insights into best practices, challenges, and strategies for successful migration.

Through a comprehensive literature review, we identify key principles underlying microservice architecture, including the concepts of domain-driven design, decentralized data management, and the use of lightweight communication protocols. Our research synthesizes findings from various case studies that illustrate the transition from monolithic systems to microservices across different industries. By examining these real-world examples, we reveal common pitfalls and effective strategies for overcoming them, thereby providing a roadmap for organizations contemplating similar transformations.

The paper further elaborates on the architectural methodologies employed during the decoupling process, highlighting the importance of a gradual transition rather than a complete overhaul. We propose a phased approach that emphasizes identifying and isolating business capabilities, which can then be independently developed, deployed, and scaled as microservices. This method not only minimizes disruption to ongoing operations but also allows organizations to iteratively refine their microservice architecture.

Our results indicate that organizations adopting microservice architectures experience enhanced system performance, improved deployment speed, and increased flexibility in adapting to changing business needs.

However, the transition is not without challenges. Issues such as service orchestration, data consistency, and inter-service communication require careful consideration and robust solutions. Our analysis highlights the significance of implementing effective monitoring and logging strategies to manage these complexities and ensure system reliability.

KEYWORDS: *Microservices, Decoupling, Scalability, Resilience, Distributed Systems, API, Independent Deployment, Service Communication*

Article History

Received: 05 Feb 2020 | Revised: 14 Feb 2020 | Accepted: 18 Feb 2020
