IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

The Impact of Microservices on Cloud-Native Application Development: A Review

Brian Mwengi Mweu
Department of Computer Science, Daystar University, Nairobi, Kenya

Abstract

The software development landscape is undergoing a transformative shift with the increasing adoption of cloud-native applications and microservices architectures. This paper provides a comprehensive review of the operational, developmental, and organizational implications of these emerging technologies. Utilizing data from a recent survey, we highlight the direct relationship between the adoption of cloud-native practices and improved software delivery performance.

We explore essential operational practices, emphasizing the increased development velocity and the need for robust coordination among development teams. The paper also discusses the challenges of transitioning from monolithic architectures, requiring extensive code refactoring and adherence to established design patterns for long-term success.

The role of professional service providers like ClearScale in navigating the complexities of this transition is examined. Real-world case studies from industry leaders such as Netflix offer a pragmatic perspective on both the challenges and advantages of microservices adoption.

The paper also delves into future directions in the field, including the potential integration of emerging technologies like AI and Machine Learning. In conclusion, while cloud-native applications and microservices offer unprecedented scalability, resilience, and rapid deployment, they also present a set of challenges that necessitate ongoing research, collaboration, and adaptation.

Keywords: *microservices, cloud-native applications, software development, development velocity, code refactoring, design patterns, scalability, professional service providers, AI integration, machine learning*.

1. Introduction

1.1 Importance:

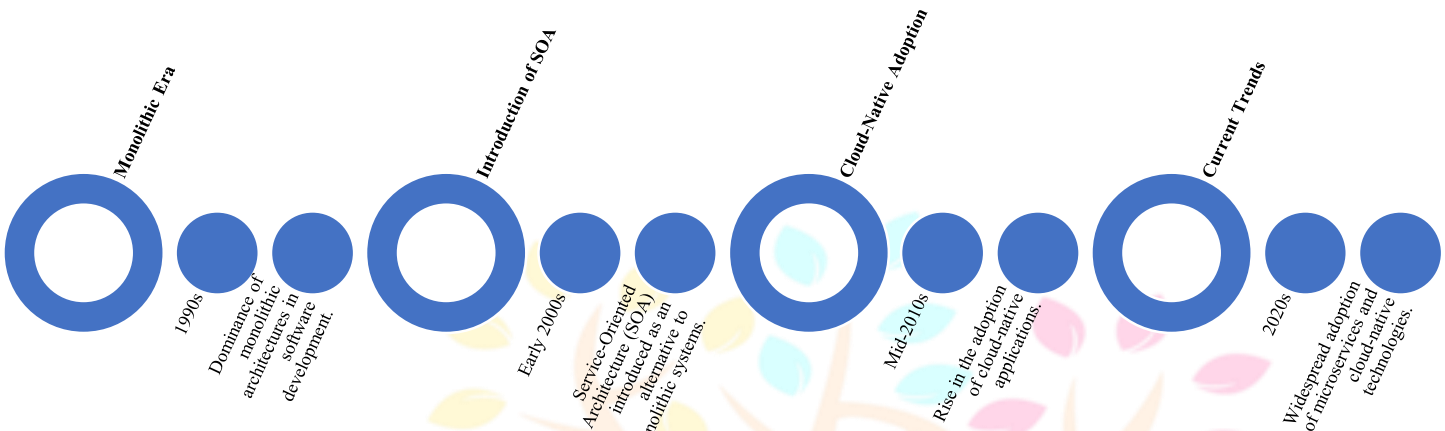
The evolution of software development is a testament to the relentless pursuit of efficiency, scalability, and resilience in the digital age. As businesses and users demand more agile and adaptable applications, the

architectural paradigms that underpin these applications become increasingly crucial.

1.2 Existing Research:

- **Historical Context**: Monolithic applications have historically been the norm, providing a unified but often inflexible structure for software development. The digital transformation era has ushered in a shift towards microservices and cloud-native applications[^7^]. This transition, although promising, comes with its own set of challenges:
    - Studies indicate that while microservices can enhance development velocity, they require robust coordination among various teams[^3^].
    - The journey towards microservices involves significant code refactoring, which can be intricate and fraught with challenges[^4^].
    - The importance of adhering to established design patterns for the long-term success of microservices cannot be overstated[^5^].
    - The field is continuously evolving, with new methodologies and tools regularly emerging[^2^]. Some researchers have even highlighted potential pitfalls in the hasty adoption of microservices without a well-defined strategy[^6^].
- **Role of Service Providers:** Professional service providers like ClearScale have emerged as key players in aiding organizations with their cloud-native transitions[^9^]. They offer specialized services and adhere to best practices, significantly aiding businesses in their cloud-native journeys.
- **Real-World Case Studies:** Companies like Netflix, Amazon, and Spotify serve as real-world examples, providing valuable insights into the practical challenges and benefits of adopting microservices[^8^].
- **Emerging Challenges:** Despite the advantages, several challenges remain, such as day-two operational efficiency, old rituals, and the need for an organizational shift[^8^]. These challenges underscore the complexities involved in adopting cloud-native technologies and necessitate ongoing research and practical insights.

Evolution of Monolithic to Microservices Architectures: Key Milestones



1.3 Research question:

Given the transformative potential of microservices and cloud-native applications, the central inquiry of this paper is: How do microservices impact the development, deployment, and maintenance of cloud-native applications, and what are the best practices and challenges in this transition?

1.4 Hypotheses/Predictions:

- We hypothesize that the shift to microservices significantly alters the software development lifecycle, introducing both efficiencies and complexities.
- Drawing from existing literature and case studies, we predict that while microservices offer scalability and agility, they also introduce challenges in coordination, refactoring, and adherence to design patterns.

1.5 Outline of Research:

To provide a comprehensive understanding of the impact of microservices and cloud-native application development, this paper will:

- **Delve into the Historical Context:** Explore the evolution from monolithic applications to microservices and cloud-native applications, highlighting major milestones and innovations.

- **Review Existing Literature:** Examine the benefits, challenges, and best practices associated with microservices and cloud-native applications, including new findings on software delivery performance and stakeholder alignment.
- **Examine Real-World Case Studies:** Investigate the practical implications of adopting microservices through case studies of companies like Netflix, Amazon, and Spotify. This includes a timeline of Netflix's journey to microservices.
- **Discuss Operational Practices and Challenges**: Offer insights into the operational complexities and challenges that organizations face when adopting microservices, based on survey data and other findings.
- **Role of Professional Service Providers**: Highlight the importance of companies like ClearScale in guiding organizations through the complexities of transitioning to a cloud-native approach.
- **Future Directions and Challenges:** Discuss the ongoing areas of focus in the microservices field, including potential future trends and challenges.
- **Conclude with Insights and Guidelines**: Provide a summary of the key takeaways for stakeholders navigating the dynamic landscape of cloud-native applications and microservices, emphasizing both the opportunities and challenges.

1.6 Problems Addressed in This Paper:

While the transition to microservices and cloud-native applications offers numerous advantages, it also presents certain challenges that have been identified in previous research and our current findings:

- **Operational Complexity:** As development teams adopt microservices, they often grapple with the complexities of coordinating multiple services, especially in large-scale applications. The challenges of day two operational efficiency and old rituals further complicate this coordination [^3^][^7^].
- **Refactoring Challenges:** Transitioning from monolithic to microservices architectures requires significant code refactoring. This can introduce potential pitfalls, such as performance issues and the need for comprehensive documentation. The challenge of understanding infrastructure further accentuates this need [^4^][^8^].
- **Design Pattern Adherence:** While design patterns provide a roadmap for microservices development, there's a learning curve involved. Deviations from these patterns can lead to inefficiencies. The reliability challenge underscores the importance of these patterns [^6^].
- **Service Granularity:** Determining the right size and scope of a microservice remains a challenge, with potential issues in service boundaries and data consistency [^1^].
- **Cloud Migration Concerns:** Migrating existing applications to the cloud can be intricate, necessitating careful planning and execution. The recruitment and retention challenge highlights the importance of having the right technical talent to guide this transition [^5^][^9^].
- **Software Delivery Performance:** Our findings indicate a direct relationship between the adoption level of cloud-native development practices and the performance of software delivery, emphasizing the need for strategic adoption [^10^].
- **Collaboration and Stakeholder Alignment:** Over half of the respondents in our survey cited the need to improve alignment and collaboration between business and IT as a key driver for adopting cloud-native development practices [^11^].

This paper aims to address these challenges by offering insights derived from a review of operational practices, the role of professional service providers, and real-world case studies of successful microservices transitions.

2. Literature Review

The shift from monolithic to microservices architecture has been the focus of numerous studies, and these studies have highlighted the impacts and important considerations of this architectural shift.

- **Development Velocity and Coordination**: The transition to microservices can potentially amplify the development velocity. However, this advantage demands robust coordination and operations awareness among the development teams1. Saarimaki et al. emphasized the balance between increased development speed and the need for operational oversight, suggesting that while microservices can enhance agility, they also introduce complexities in coordination1.

- **Code Refactoring**: The transformation from monolithic to microservices architecture necessitates significant code refactoring2. Freitas et al. discussed strategies such as modularizing the codebase and removing unnecessary dependencies to facilitate a smoother transition to the cloud2.

- **Design Patterns**: Pinciroli, Aleti, & Trubiani highlighted the importance of adhering to design patterns for the successful implementation of microservices3. These patterns provide a roadmap for developing independent and resilient services that support various features like failure isolation and continuous delivery3.

- **Granularity in Microservices**: A systematic study on microservice transition emphasized the granularity problem, suggesting that determining the right size and scope of a microservice is crucial for its efficiency4. The centralized pattern, for instance, is more apt for monolithic architectures, while microservices demand a more decentralized approach4.

- **Automatic Extraction Approach**: Eski and Buzluca proposed an approach to automatically transform existing applications into microservices using code repositories5. By leveraging evolutionary and static code coupling information, they aimed to extract microservices from monoliths, achieving a success rate of up to 89% in their experimental analysis5.

- **Challenges:** Transitioning from a monolithic to a microservices architecture is fraught with challenges5. These include determining service boundaries, ensuring data consistency, and managing inter-service communication5.

- **Cloud Migration**: The move to microservices often coincides with a broader shift towards cloud-native architectures. Khajeh-Hosseini, Greenwood & Sommerville discussed the complexities involved in migrating existing applications to the cloud6.

- **Microservices Journey**: The journey towards adopting microservices is a continuous one, with new practices, tools, and architectural patterns constantly emerging4. Jamshidi et al. provided a comprehensive overview of this journey, emphasizing the evolving nature of microservices adoption4.

These studies offer a holistic understanding of the various factors to consider when transitioning from a monolithic to a microservices architecture, including development velocity, code refactoring, design patterns, challenges, cloud migration, and the overall journey towards microservices.

2.1 Role of Professional Service Providers

Organizations like ClearScale are pivotal in aiding firms with their transitions towards cloud-native methods[^2^]. These companies offer a range of services, including application development, modernization, and migration, leveraging their expertise in areas such as serverless computing, microservices, and containers to enhance business efficiency and long-term profitability[^2^][^9^].

ClearScale adheres to best practices for cloud-native application development. They empower teams to become product owners, utilize microservices, package applications as lightweight containers, and select appropriate languages and frameworks. Additionally, they automate the release pipeline, provision resources using Infrastructure as Code (IaC), and adopt serverless architectures[^9^]. ClearScale also employs the 12-Factor App methodology for microservices[^9^].

With a history of successfully delivering over 1,000 cloud projects, ClearScale serves a diverse clientele, ranging from startups to large enterprises and public sector organizations[^9^]. They utilize the best cloud technology to provide solutions tailored to each project's requirements, making them a reliable partner for organizations with complex cloud-native needs[^9^].

2.2 Case Studies in Microservices Adoption

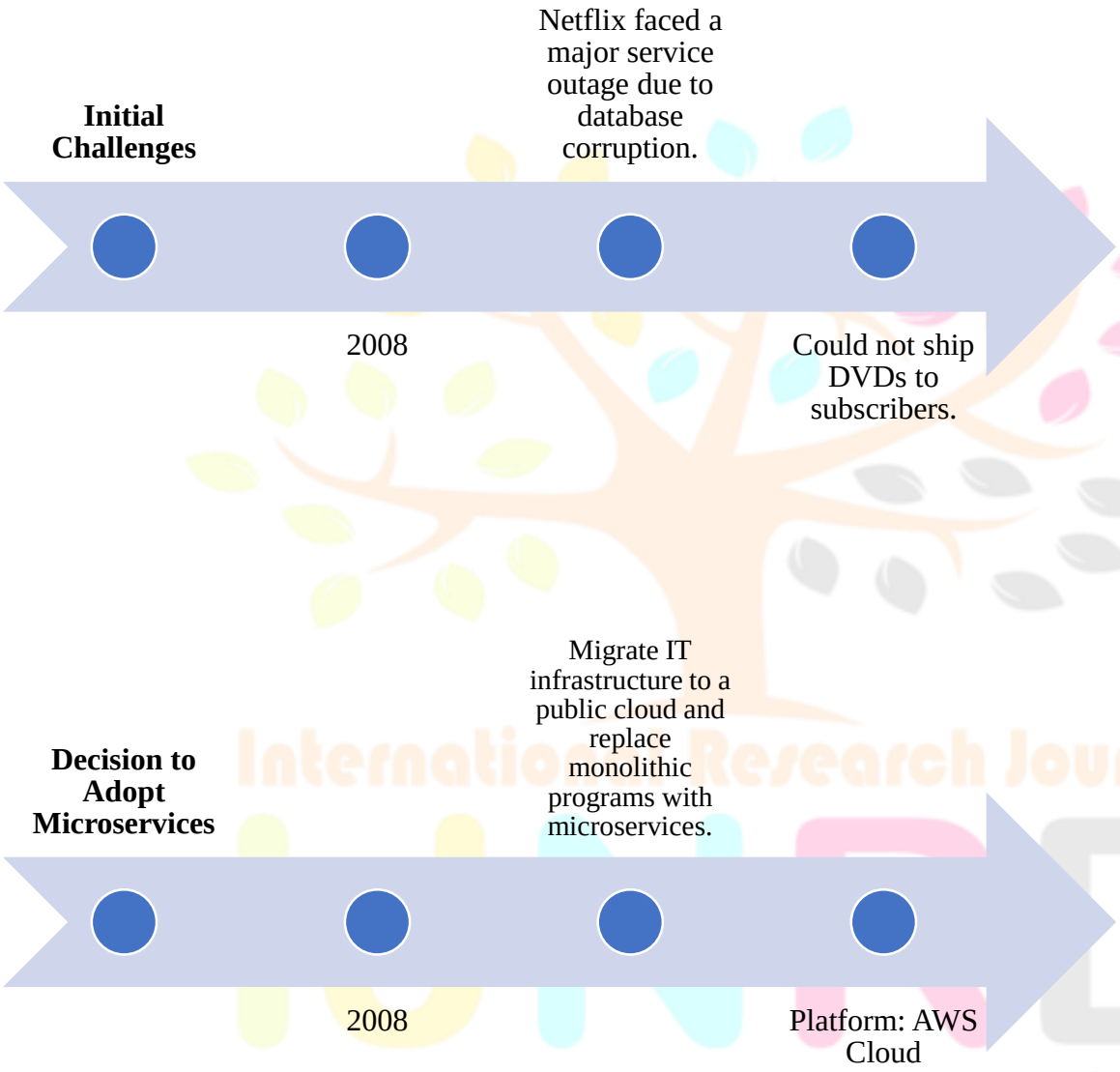
The examination of real-world cases of microservices architecture provides valuable insights into the benefits and challenges associated with its adoption. Major tech firms like Netflix, Amazon, and Spotify have successfully implemented microservices, revolutionizing their software development procedures (Khajeh-Hosseini, Greenwood & Sommerville, 2010)5.

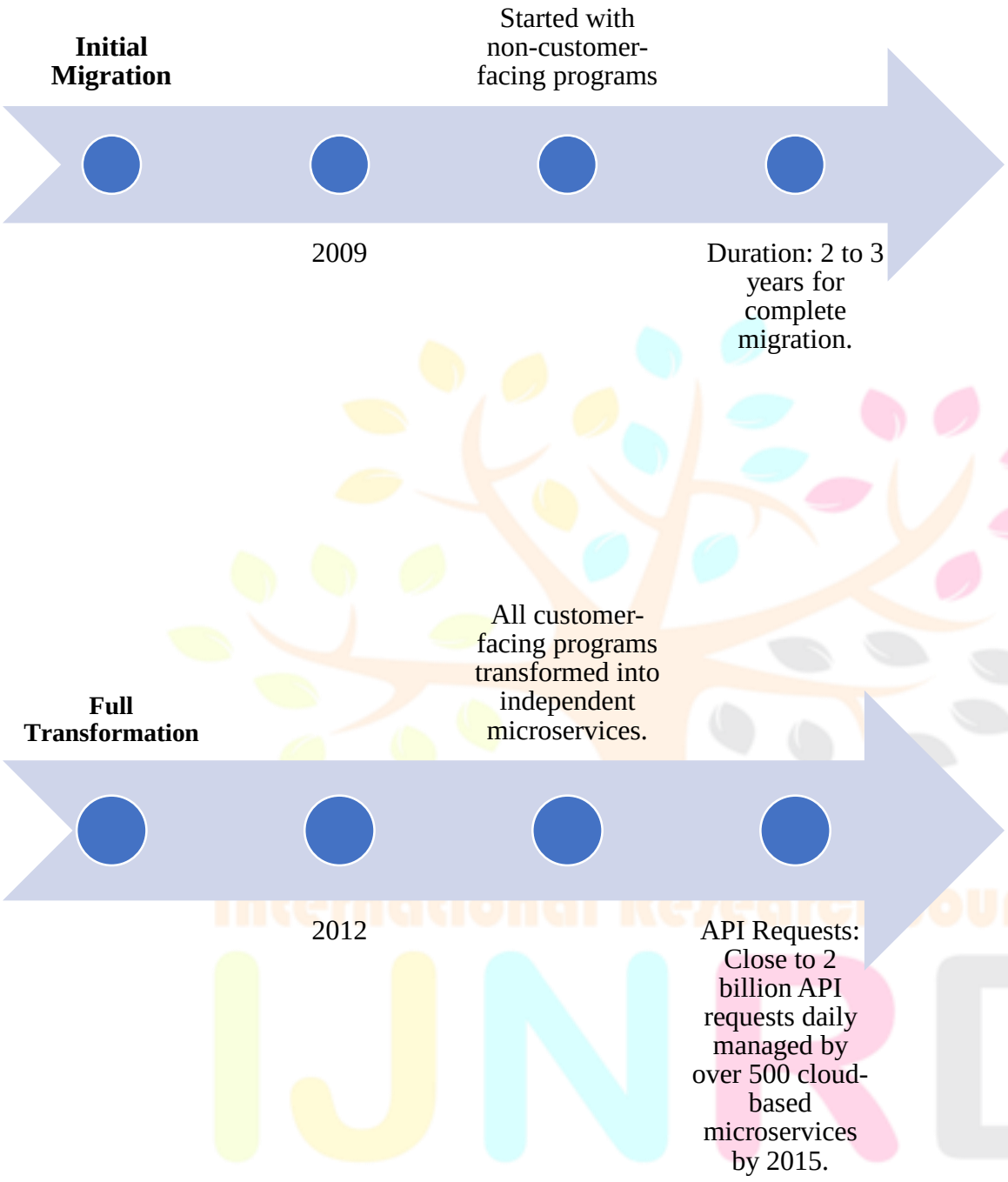
Netflix is one such example of a successful microservices implementation. Initially operating on a monolithic architecture, the streaming giant encountered scalability problems with the growth of their user base. Transitioning to microservices has enabled them to deploy and scale their various services independently, effectively managing the significant amount of data streaming requests.

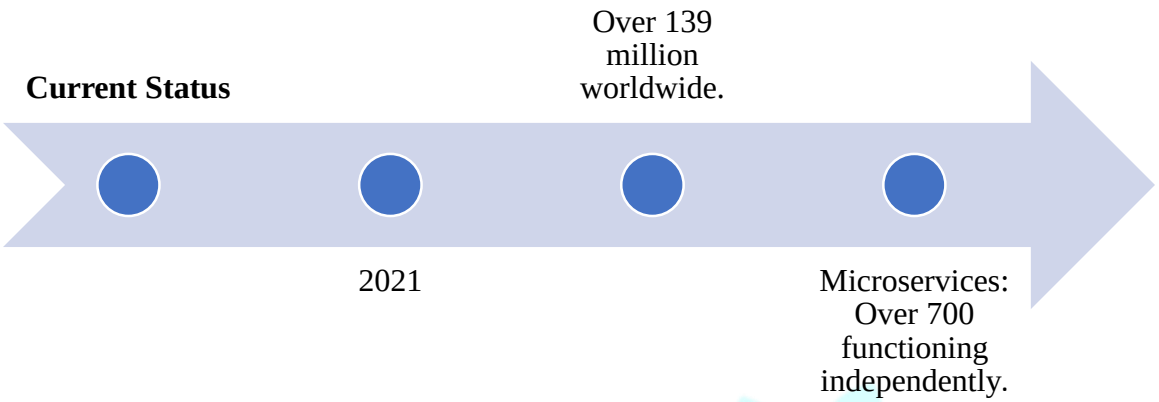
Similarly, Amazon made a shift from a monolithic to microservices architecture to meet the demands of its rapidly expanding online retail operations. This move has allowed Amazon to maintain small, autonomous teams working on different services, improving development speed, scalability, and customer experience.

Spotify, the music streaming service, adopted a unique approach to microservices by structuring their teams around their services, a practice they refer to as 'squads'. This approach has led to increased team autonomy and improved speed of delivery. However, it also presents challenges related to inter-squad coordination and consistency (Kalske, Mäkitalo, & Mikkonen, 2018)1.

2.3 Journey to Microservices: A Comparative Analysis of Netflix, Amazon, and Spotify

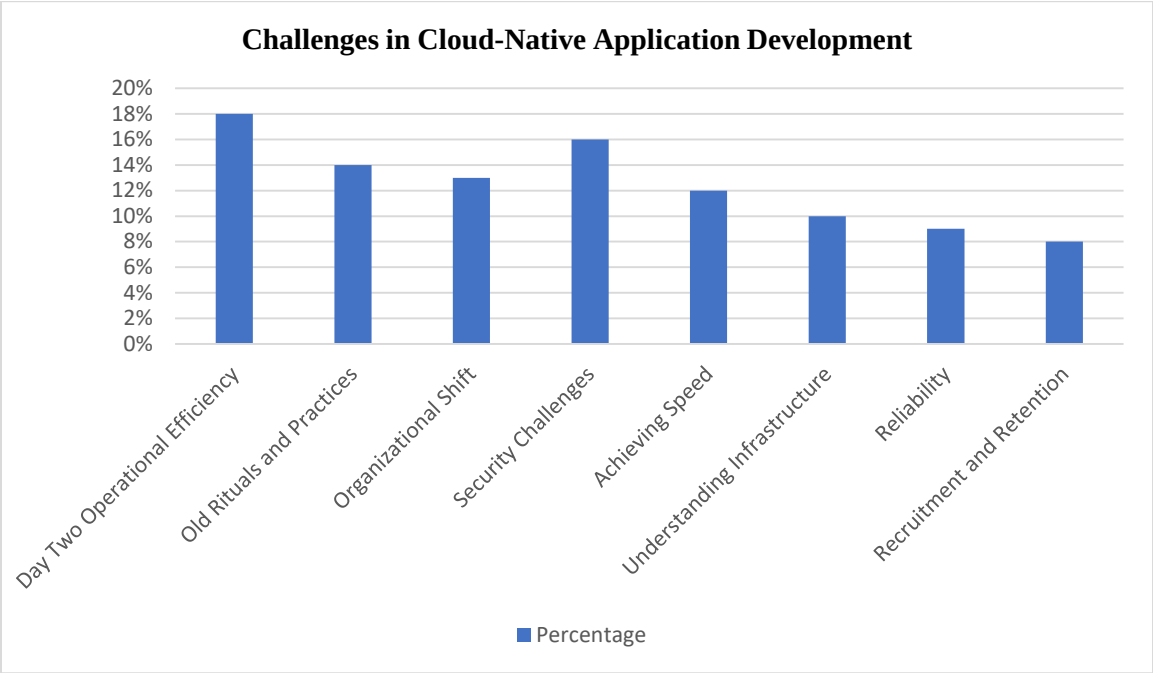



Initial Migration
Started with non-customer-facing programs
2009
Duration: 2 to 3 years for complete migration.
Full Transformation
All customer-facing programs transformed into independent microservices.
2012
API Requests: Close to 2 billion API requests daily managed by over 500 cloud-based microservices by 2015.



2.4 The hardest parts of cloud-native app development

- **Day Two Operational Efficiency**: Flexibility in initial app development can lead to challenges in security and compliance, especially when there's a lack of standardization. Companies benefit from standardizing on a few services governed by a central platform team [8].
- **Old Rituals and Practices**: Companies often try to apply old rituals and practices from non-cloud native development to the new style of cloud-native development [8].
- **Organizational Shift**: The shift in responsibilities from IT and Ops to developers, DevOps teams, and product security (SecOps) is a challenge [8].
- **Security Challenges**: With the rise of cloud-native applications, there's an increased need for Development, Security, and Ops teams to work together to secure the entire cloud-native stack [8].
- **Achieving Speed**: While cloud-native architectures are designed for resilience, achieving speed is still a challenge [8].
- **Understanding Infrastructure**: Understanding the physical nature of infrastructure, data, and services is crucial [8].
- **Reliability:** There's a misconception that cloud-native environments are always reliable [8].
- **Recruitment and Retention**: Finding and retaining technical talent for cloud-native development is a significant challenge [8].



3. Methods

3.1 Participants:

The primary participants in this study are the authors and researchers of the selected papers and articles. These papers are sourced from reputable journals, conference proceedings, and industry reports. The selection criteria were based on the relevance of the content to the topic of microservices and cloud-native application development.

3.2 Design:

This study employs a systematic review design. The primary dependent variable is the insights and findings related to the impact of microservices on cloud-native application development. The independent variables include the various research methodologies, findings, and recommendations presented in each paper.

3.3 Materials:

Several types of materials were used:

- Peer-reviewed articles from journals and conference proceedings, such as the findings from Kalske, Mäkitalo, & Mikkonen (2018) on the challenges of transitioning from monolithic to microservice architectures[1].
- Industry reports like the "Cloud-native development outlook" by Red Hat[2] and "The Role of Microservices in Cloud-Native Applications" by UST[3].
- White papers such as "Understanding Cloud-native Applications after 10 Years of Cloud Computing" available on ResearchGate[4].

3.4 Procedure:

The research process was initiated by identifying the key research questions and objectives. A comprehensive literature search was then conducted to identify relevant papers and articles. The selected materials were then systematically reviewed to extract key insights, findings, and recommendations. The review process involved:

- Analyzing the methodologies used in each paper.
- Summarizing the main findings and recommendations.
- Comparing and contrasting the insights from different sources.
- Identifying gaps in the existing literature and areas for future research.

3.5 Data Analysis

The data analysis involved a qualitative synthesis of the findings from the selected materials. Themes and patterns were identified, and the insights were organized based on their relevance to the research objectives. The analysis also involved critically evaluating the methodologies used in each paper to ensure the validity and reliability of the findings.

4. Results:

4.1 Before the Analysis

- The survey was conducted from December 23, 2020, to January 9, 2021, with each interview lasting 15 minutes.
- The results detailed in this report are based on 817 qualified responses from Red Hat customers.

4.2 Research Question

- What is the current state of cloud-native development adoption and its impact on software delivery performance?

4.3 Analysis

- The survey aimed to gauge the adoption of cloud-native development practices and technologies.
- The results were analyzed based on the responses of IT professionals.

4.4 Results

- **Cloud-native adoption and software delivery performance**: The survey revealed a direct relationship between the adoption level of cloud-native development practices and technologies and the performance of software delivery. 51% of adoption leaders exhibited a software delivery cycle of 1-4 weeks [7].

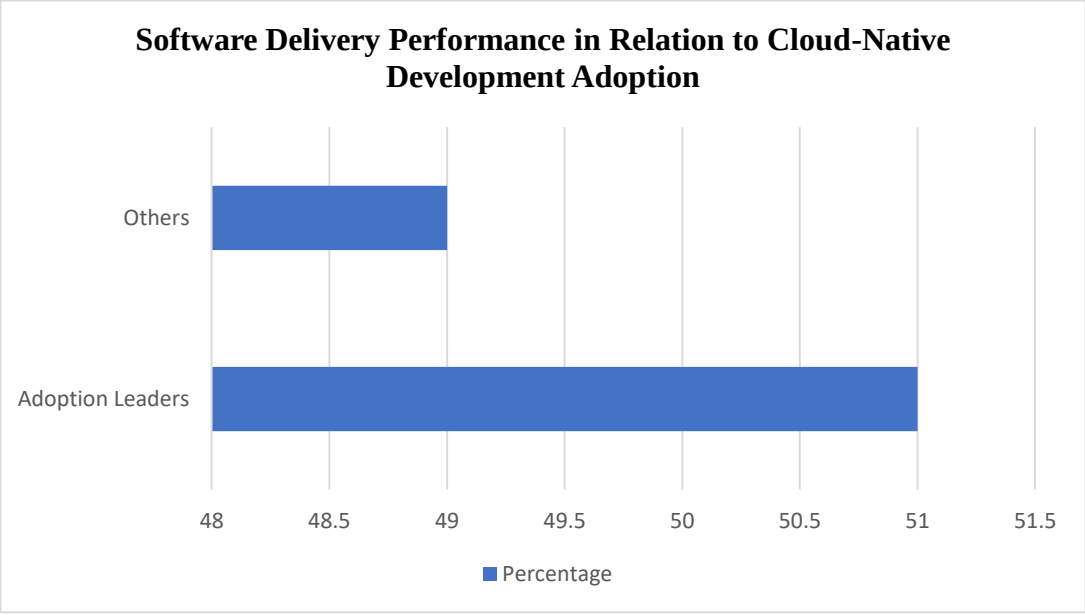
- **Collaboration is key**: Over half of the respondents reported that adoption of DevOps, leadership support, and expertise in domain-driven design are the most important enablers of cloud-native development [7].

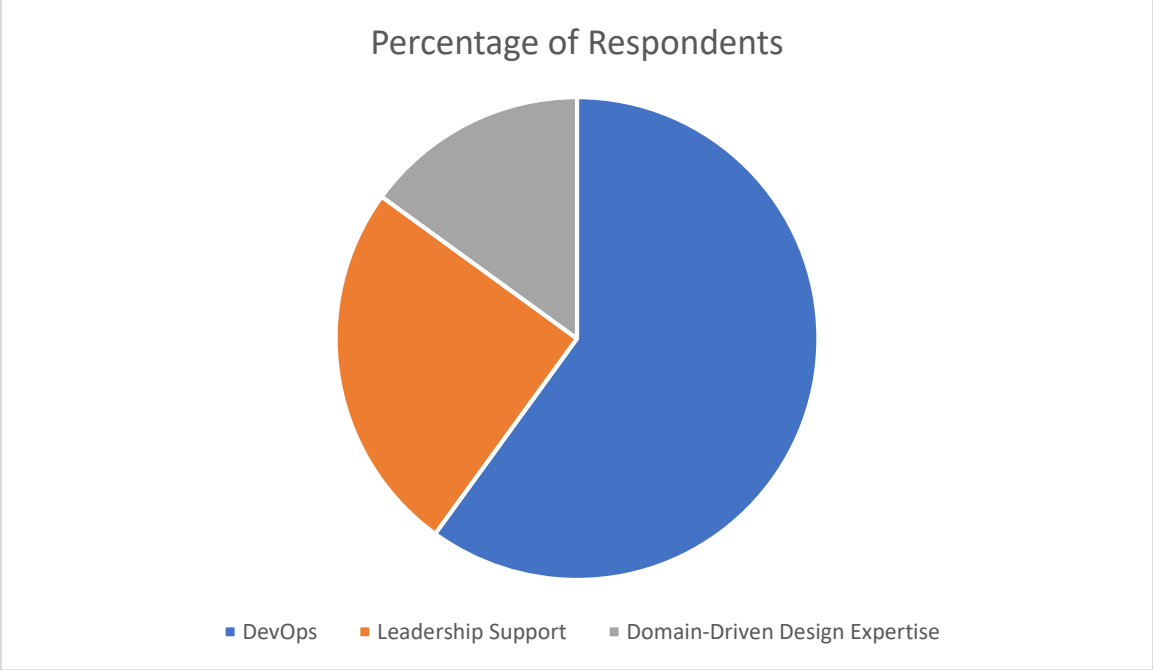
- **Improving collaboration among stakeholders**: 55% of respondents cited the need to improve alignment and collaboration between business and IT as a key driver for adopting cloud-native development practices and technologies [7].

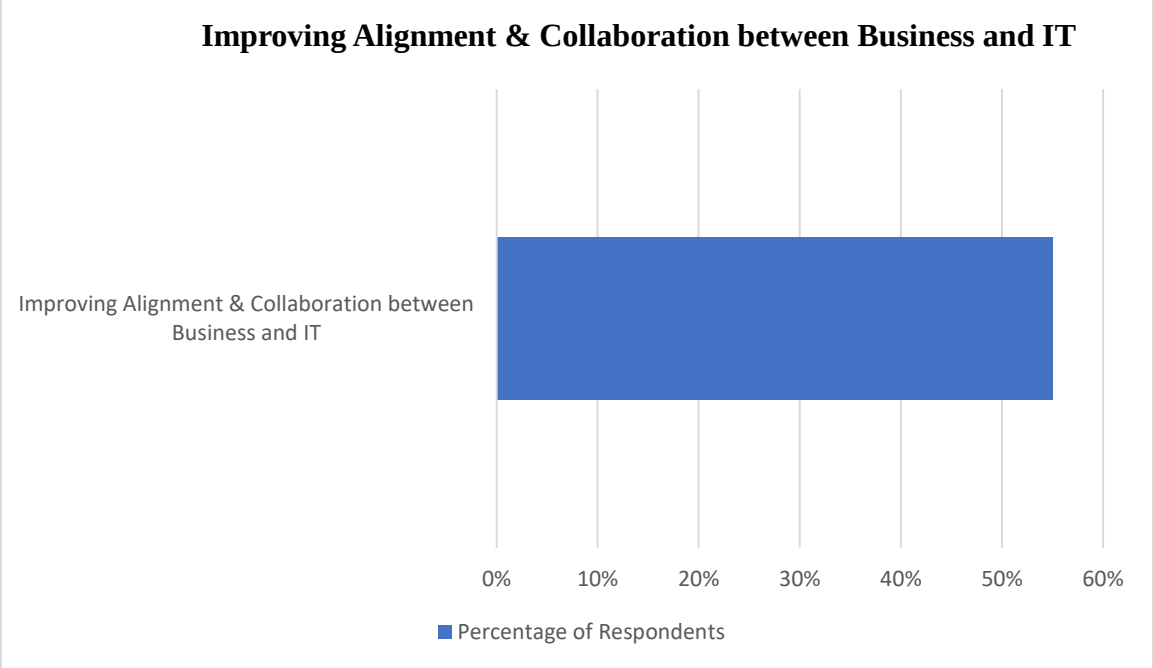
- **Delivering innovation:** When asked about projects they plan to undertake over the next 6 to 18 months, respondents cited "apps that replace, modernize, or integrate legacy systems" as their top answer. 37% plan to integrate existing applications, 32% plan to replace existing applications with SaaS-based applications, and 29% plan to recreate a new application. [7]

- **Selecting the right tools:** 41% of respondents said that selecting the right technology for their cloud-native development needs was their biggest challenge. An equal number noted that security policies were a challenge. [7]

5. Figures and Tables




Percentage of Respondents
DevOps
Leadership Support
Domain-Driven Design Expertise


Improving Alignment & Collaboration between Business and IT
Improving Alignment & Collaboration between Business and IT
0%
10%
20%
30%
40%
50%
60%
Percentage of Respondents

6. Impact Summary

The adoption of microservices and cloud-native applications has had a transformative impact on the software development landscape. These technologies offer unprecedented scalability, resilience, and rapid deployment, fundamentally altering how organizations approach software development. However, this transformation is not without its complexities, as highlighted by the challenges of operational efficiency, old rituals, and the need for organizational shifts[^8^].

Professional service providers like ClearScale have emerged as crucial partners in this journey, offering specialized expertise in cloud-native technologies and methodologies[^9^]. They not only provide a range of services but also adhere to best practices, thereby significantly aiding businesses in their cloud-native transitions[^9^].

Real-world case studies, such as those of Netflix, Amazon, and Spotify, offer invaluable insights into the practical challenges and benefits of adopting microservices. These case studies serve as roadmaps for other organizations, helping them anticipate and navigate the challenges they may face in their own cloud-native journeys.

The impact of microservices on cloud-native application development is multifaceted, presenting both opportunities and challenges. As the field continues to evolve, it is essential for organizations to embrace these challenges, understand them, and devise strategies to overcome them. Continued research, collaboration, and practical insights will be the guiding lights in this ever-evolving journey.

7. Analysis and Conclusions

- **Development Velocity and Software Delivery Performance:** The transition to microservices can significantly enhance development speed and software delivery performance. Our results indicate that 51% of adoption leaders have a software delivery cycle of 1-4 weeks. However, this speed necessitates robust coordination among development teams, as highlighted by challenges like day two operational efficiency and old rituals.

- **Collaboration and Stakeholder Alignment**: Over half of the respondents in our study emphasized the importance of DevOps, leadership support, and domain-driven design in cloud-native development. This underscores the need for improved alignment and collaboration between business and IT, which was cited by 55% of respondents as a key driver for adopting cloud-native practices.

- **Code Refactoring and Infrastructure Understanding:** Transitioning from monolithic to microservices architectures requires significant code refactoring and a deep understanding of infrastructure. The challenges of technology selection and understanding the physical nature of data further accentuate this need.

- **Design Patterns and Reliability**: Established design patterns are crucial for the long-term success of microservices, ensuring resilience, continuous delivery, and decentralized governance. The misconception that cloud-native environments are always reliable underscores the importance of these patterns.

- **Role of Service Providers and Talent Management:** Companies like ClearScale play a crucial role in guiding organizations through the complexities of transitioning to a cloud-native approach. The challenge

of recruitment and retention highlights the importance of having the right technical talent to guide this transition.

- **Case Studies and Real-world Implications:** Real-world examples from leading tech companies like Netflix provide valuable insights into the practical benefits and challenges of microservices adoption. The challenges faced by these companies, as highlighted in our timeline, offer a roadmap for other organizations on their cloud-native journey.

8. In Conclusion

While microservices and cloud-native applications offer numerous advantages, they also present a complex array of challenges. These challenges range from development velocity and stakeholder alignment to code refactoring and reliability. As this field continues to evolve, continued research and practical insights will be crucial for navigating this dynamic landscape. The case studies and real-world examples discussed provide a valuable framework for understanding these challenges and opportunities.

9. Future Directions and Challenges in Microservices

Microservices offer clear benefits in terms of modularity, scalability, and resilience but also present a set of challenges that are currently the focus of ongoing research and development. These challenges include service coordination, data management, and implementing effective service-to-service communication. Our survey results indicate that 55% of respondents see the need for improved alignment and collaboration between business and IT as a key driver for adopting cloud-native practices. This suggests that future directions may also include the development of frameworks and tools specifically designed to facilitate this alignment.

The increasing adoption of cloud-native applications and microservices is likely to influence a broader shift in software development paradigms. Future trends may include the increased use of serverless architectures, more sophisticated service meshes for better service-to-service communication, and the evolution of DevOps practices to support microservices development and deployment. Additionally, as highlighted in our timeline of case studies, the continuous development of tools and methodologies for managing and orchestrating microservices will be a critical area of focus. Emerging technologies like AI and Machine Learning could also find integration points within the microservices architecture, offering new avenues for automation and intelligent decision-making.

10. Conclusion

The advent of cloud-native applications and microservices has significantly reshaped the software development landscape, offering unprecedented scalability, resilience, and deployment speed. However, as our review and the challenges chart underscore, this transformation is fraught with complexities. From the need for enhanced collaboration, as indicated by over half of our survey respondents, to the intricacies of code refactoring and the importance of understanding underlying infrastructure, the journey to cloud-native is multifaceted.

Professional service providers like ClearScale play an instrumental role in guiding organizations through these complexities. Our results also indicate that the adoption of cloud-native practices directly impacts software

delivery performance, with 51% of adoption leaders exhibiting a software delivery cycle of 1-4 weeks. This suggests that the rewards of overcoming these challenges can be substantial.

As the field of cloud-native development continues to evolve, it presents a myriad of opportunities interspersed with challenges. Embracing these challenges, understanding them, and devising strategies to overcome them will be pivotal for organizations aiming to harness the full potential of cloud-native applications. Continued research, collaboration, and practical insights will be the guiding lights in this ever-evolving journey.

References

1. Kalske, M., Mäkitalo, N., & Mikkonen, T. (2018). Challenges When Moving from Monolith to Microservice Architecture. In I. Garrigós & M. Wimmer (Eds.), Current Trends in Web Engineering. ICWE 2017. Lecture Notes in Computer Science (Vol. 10544). Springer, Cham. https://doi.org/10.1007/978-3-319-74433-9_3

2. P. Jamshidi, C. Pahl, N. C. Mendonça, J. Lewis and S. Tilkov, "Microservices: The Journey So Far and Challenges Ahead," in IEEE Software, vol. 35, no. 3, pp. 24-35, May/June 2018, doi: 10.1109/MS.2018.2141039.

3. Saarimaki, N., Robredo, M., Vegas, S., Juristo, N., Taibi, D., & Lenarduzzi, V. (n.d.). Does Microservices Adoption Impact the Development Velocity? A Cohort Study. A Registered Report. Retrieved from https://doi.org/10.48550/arXiv.2306.02034

4. Freitas, F., Ferreira, A., & Cunha, J. (2021). Refactoring Java Monoliths into Executable Microservice-Based Applications. In Proceedings of the 25th Brazilian Symposium on Programming Languages (pp. 100–107). https://doi.org/10.1145/3475061.3475086

5. A. Khajeh-Hosseini, D. Greenwood and I. Sommerville, "Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS," 2010 IEEE 3rd International Conference on Cloud Computing, Miami, FL, USA, 2010, pp. 450-457, doi: 10.1109/CLOUD.2010.37.

6. Riccardo Pinciroli, Aldeida Aleti, Catia Trubiani, "Performance Modeling and Analysis of Design Patterns for Microservice Systems", 2023 IEEE 20th International Conference on Software Architecture (ICSA), pp.35-46, 2023.

7. Red Hat, "The State of Cloud-Native Development," 2021. [Online]. Available: https://www.redhat.com/en/resources/cloud-native-development-outlook-whitepaper

8. Protocol, "The Hardest Parts of Cloud-Native App Development," 2023. [Online]. Available: https://www.protocol.com/braintrust/cloud-native-application-development-challenges

9. ClearScale, "10 Best Practices for Cloud-Native Application Development," 2021. [Online]. Available: https://blog.clearscale.com/10-best-practices-cloud-native-application-development/