# DIVERGENCES AND CONVERGENCES BETWEEN SOFTWARE ARCHITECTURE AND CLOUD ARCHITECTURE: A CRITICAL ANALYSIS

https://doi.org/10.56238/sevened2025.008-033

**Cláudio Filipe Lima Rapôso[1].**

## ABSTRACT
This paper explores the divergences and convergences between software architecture and cloud architecture, highlighting how these two areas interrelate and influence the development of modern systems. Through a detailed literature review, we identify gaps in the current literature and propose a replicable methodology to investigate these relationships. Key findings include identifying challenges and benefits in integrating cloud solutions into software architectures, as well as the need for new architectural approaches to maximize efficiency and security. Additionally, we analyze case studies that illustrate the practical application of these architectures in various business contexts, highlighting the structural changes required and the challenges faced by companies. We conclude that while cloud computing offers significant advantages in terms of scalability and flexibility, effectively integrating these solutions requires a careful reassessment of existing software architectures to ensure security and operational efficiency.

**Keywords:** Software Architecture. Cloud Architecture. Cloud Computing. System Integration.

[1] Bachelor in Production Engineering from Faculdade Estácio do Recife, Master in Computer Science, Master in Business Administration, and Doctoral Student in Business Administration at Atlanta College of Liberal Arts and Sciences, and Master of Science in Business Administration Student at Must University. Hon. D. Sc in Computer Science at Graham Internacional University E-mail: engcfraposo@outlook.com.br

## INTRODUCTION

Cloud computing has significantly transformed the way software systems are developed, deployed, and managed. The flexibility and scalability offered by the cloud require software architectures to adapt to new technological demands (Rapôso, Costa Junior, & Ferreira, 2024). In this context, it is essential to understand how software and cloud architectures converge and diverge in the practice of developing modern systems. The main question this study seeks to answer is: how do software and cloud architectures interrelate and influence the development of modern systems?

The general objective of this study is to analyze the divergences and convergences between software architecture and cloud architecture. To achieve this, the main structural changes in software architectures due to the adoption of cloud services are identified, the benefits and challenges faced by companies in integrating cloud solutions are analyzed, and case studies demonstrating the evolution of architectures in this context are presented. Software architecture is defined as the organizational structure of a software system, including its components and the relationship between them (Bass, Clements, & Kazman, 2012), while cloud architecture is the structure that defines how the components of a system are distributed and managed in a cloud computing environment (Erl, 2013).

Understanding the interactions between software architecture and cloud architecture is crucial for the development of efficient and secure systems, especially in a context where the demand for scalable and flexible solutions is growing (Richards, 2015).

## THEORETICAL FOUNDATION

### CLOUD *COMPUTING*

Cloud computing refers to the delivery of computing services such as servers, storage, databases, networking, software, analytics, and intelligence over the internet. This approach enables faster innovation, flexible resources, and economies of scale (Erl, 2013). The main characteristics of cloud computing include elasticity, scalability, and on-demand access. These characteristics are fundamental for the development of modern systems, as they allow companies to adjust their resources according to demand, optimizing costs and improving operational efficiency (Richards, 2015).

In this context, cloud computing offers a pay-as-you-go model, which can be highly beneficial for companies of all sizes. This eliminates the need for large initial investments in IT infrastructure, allowing companies to scale their resources according to their needs. However, cloud adoption also brings challenges, such as the need to ensure data security and compliance with specific regulations (Rapôso, Costa Junior, & Ferreira, 2024).

Cloud computing also facilitates collaboration and mobility, allowing employees to access data and applications from anywhere, at any time. This is particularly important in an increasingly digital and remote world, where the ability to work flexibly is essential. However, this flexibility can also introduce security vulnerabilities that need to be carefully managed (Erl, 2013).

Another important aspect of cloud computing is disaster recovery and business continuity. The cloud enables companies to implement robust backup and recovery solutions, ensuring that data and systems can be quickly restored in case of failures or disasters. This is crucial to minimize downtime and ensure the continuity of operations (Richards, 2015).

Cloud computing is also driving innovation in various industries, enabling the development of new applications and services that were previously impractical due to infrastructure limitations. Emerging technologies such as artificial intelligence and machine learning are being widely adopted thanks to the cloud's ability to provide the computational resources needed to process large volumes of data (Rapôso, Costa Junior, & Ferreira, 2024).

Thus, cloud computing is transforming business models, allowing companies to offer new products and services more agilely and efficiently. Subscription-based business models, for example, are becoming increasingly common, allowing companies to generate recurring revenue and provide continuous value to customers (Erl, 2013).

## CLOUD ARCHITECTURE

Cloud architecture involves the structure and design of systems that use cloud computing services. It focuses on the distribution and management of system components in a cloud environment, ensuring that resources are used efficiently and securely (Erl, 2013). Cloud architecture should be designed to maximize the advantages of the cloud, such as scalability and flexibility, while addressing challenges such as security and integration (Zimmermann, 2017).

Reference models are often used to specify the main activities and functions of cloud architecture, aligning business and service architectures for the cloud. These models help standardize practices and ensure that cloud solutions are implemented consistently and effectively. Cloud architecture should also consider aspects such as resilience and disaster recovery, ensuring that systems can continue operating even in case of failures (Erl, 2013).

Cloud architecture should be designed to support automation and orchestration of resources, allowing companies to manage their cloud environments more efficiently.

Automation tools can help reduce operational complexity and improve efficiency, enabling IT teams to focus on higher-value activities (Zimmermann, 2017).

Another important aspect of cloud architecture is security. Cloud security involves implementing controls and practices to protect data, applications, and infrastructure against threats and vulnerabilities. This includes data encryption, identity and access management, and the implementation of robust security policies (Erl, 2013).

Cloud architecture should be flexible enough to support different deployment models, such as public, private, and hybrid clouds. Each deployment model has its own advantages and challenges, and the right choice depends on the specific needs of the company and security and compliance requirements (Zimmermann, 2017).

Cloud architecture should be designed to support horizontal scalability, allowing systems to increase or decrease capacity as needed. This is particularly important for applications that face significant variations in demand, such as e-commerce sites during peak sales periods (Erl, 2013).

## SOFTWARE ARCHITECTURE

Software architecture is the organizational structure of a software system, including its components and the relationship between them (Bass, Clements, & Kazman, 2012). It defines the standards and guidelines for system development, ensuring that components work cohesively and efficiently. Software architecture must be adaptable to incorporate new technologies and practices, such as cloud computing, which require significant structural changes (Rapôso, Costa Junior, & Ferreira, 2024).

Integrating cloud solutions into existing software architectures can be complex and require significant restructuring, but it also offers benefits in terms of scalability and efficiency. The adoption of microservices, for example, is an approach that facilitates integration with the cloud, allowing systems to be more modular and flexible. However, this integration also brings challenges, such as the need to manage communication between services and ensure data security (Richards, 2015).

Software architecture should also consider aspects such as maintainability and extensibility, ensuring that systems can evolve and adapt to changes in business and technology needs. This includes implementing design patterns and coding practices that facilitate software maintenance and evolution over time (Bass, Clements, & Kazman, 2012).

Another important aspect of software architecture is performance. The architecture should be designed to ensure that systems can meet performance requirements, even under high workloads. This may involve implementing performance optimization techniques such as caching and load balancing (Richards, 2015).

Software architecture should also be designed to support interoperability, allowing systems to integrate with other applications and services. This is particularly important in complex IT environments, where systems need to communicate and collaborate with a variety of other solutions (Rapôso, Costa Junior, & Ferreira, 2024).

Finally, aspects such as usability and user experience should be considered, ensuring that systems are intuitive and easy to use. This includes implementing well-designed user interfaces and considering the needs and expectations of end users (Bass, Clements, & Kazman, 2012).

## METHODOLOGY

This study focuses on the analysis of companies that have adopted cloud computing solutions in the last five years, highlighting architectural changes and challenges faced during integration (Rapôso, Costa Junior, & Ferreira, 2024). The methodology used includes a detailed literature review, identification of gaps in the current literature, proposition of clear hypotheses, and analysis of case studies that illustrate the practical application of architectures in different business contexts (Zimmermann, 2017).

To conduct the literature review on software architecture, cloud architecture, and cloud computing, a systematic and integrative approach was used, applying specific search filters in various reliable databases. First, inclusion and exclusion criteria for studies were defined, such as publication date, language, type of study, and relevance to the research question. Then, comprehensive and reliable databases such as Google Scholar, PubMed, IEEE Xplore, Scopus, and Web of Science were selected, offering access to a wide range of scientific articles, theses, dissertations, and books.

Keyword searches were conducted using relevant terms such as "software architecture," "cloud architecture," and "cloud computing." Boolean operators (AND, OR, NOT) were used to combine terms and refine the search. For example, in Google Scholar, keyword combinations such as "software architecture AND cloud computing" were used to refine the results. In IEEE X plore, terms such as "cloud architecture AND microservices" were searched to find studies on the integration of microservices in cloud environments. In Scopus, terms such as "cloud computing AND software architecture patterns" were used to identify architectural patterns in cloud integration. In Web of Science, terms such as "cloud computing AND enterprise architecture" were searched to explore the relationship between enterprise architecture and cloud services. In PubMed, terms such as "cloud computing

AND healthcare software architecture" were used to find studies on the application of cloud computing in healthcare.

In addition to databases, relevant books and book chapters were consulted, such as "Software Architecture in Practice" by Bass, Clements, and Kazman (2012) and "Cloud Computing: Concepts, Technology & Architecture" by Erl (2013). Specific chapters directly addressing the topics of interest were selected. Previous systematic reviews on related topics were also consulted to identify potential studies and relevant terms for the search, ensuring that the current review was aligned with existing literature and leveraged previous findings.

Articles from renowned conferences such as ICSE (International Conference on Software Engineering) and CloudCom (IEEE International Conference on Cloud Computing Technology and Science) were included, focusing on articles with empirical results and case studies. Dissertations and theses available in institutional repositories and databases such as ProQuest were selected, including academic works directly addressing the topics of interest. Technical reports from research institutions and technology companies were also included, providing practical insights and case studies.

The critical evaluation of pre-selected studies was conducted to ensure they met quality and relevance criteria, analyzing the methodology, results, and conclusions of the studies. All stages of the search and selection of studies were recorded and documented, including search strategies, databases used, and inclusion and exclusion criteria, ensuring transparency of the process and allowing the review to be reproduced.

## RESULTS

The analysis of divergences and convergences between software architecture and cloud architecture revealed a series of challenges and opportunities that directly impact the development and implementation of modern systems. The main results obtained from the literature review and analyzed case studies are presented below.

### COMPLEXITY IN INTEGRATION

Integrating cloud computing solutions into existing software architectures is a complex process that may require significant restructuring. The adoption of microservices, for example, facilitates the modularity and flexibility of systems but also introduces challenges related to communication between services and dependency management (Richards, 2015). Companies that adopted microservices faced difficulties in coordinating and monitoring services, especially in hybrid cloud environments (Zimmermann, 2017).

Cloud architecture provides the necessary infrastructure to support the distribution and scalability of microservices. It allows services to be deployed and managed efficiently in a distributed environment, leveraging the scalability and flexibility resources of the cloud (Erl, 2013). Automation and orchestration tools are essential to manage the complexity of integration and ensure that services work cohesively.

On the other hand, software architecture defines standards and guidelines to ensure that components work cohesively. It must be adapted to incorporate new technologies and practices, such as cloud computing, which require significant structural changes. Software architecture must ensure that microservices are designed modularly and that communication between them is efficient and secure (Bass, Clements, & Kazman, 2012).

## SECURITY

Migrating to the cloud imposes new security challenges that need to be carefully managed. Cloud security involves implementing controls and practices to protect data, applications, and infrastructure against threats and vulnerabilities (Erl, 2013). The literature review highlighted the importance of data encryption, identity and access management, and the implementation of robust security policies. Companies that adopted cloud security practices effectively mitigated risks and protected their digital assets (Rapôso, Costa Junior, & Ferreira, 2024).

Cloud architecture must ensure that security mechanisms are effectively applied in a distributed environment. This includes implementing access controls, data encryption in transit and at rest, and continuous threat monitoring. Cloud architecture must also be designed to support compliance with specific regulations and ensure data privacy (Zimmermann, 2017).

Software architecture, in turn, must incorporate robust security mechanisms such as encryption and authentication. It must ensure that system components are designed to withstand attacks and that security practices are integrated into the development process. Software architecture must also facilitate the implementation of security policies and identity and access management (Bass, Clements, & Kazman, 2012).

## SCALABILITY

While the cloud offers scalability, adapting software architectures to fully leverage this feature can be challenging. Horizontal scalability, which allows increasing or decreasing capacity as needed, is essential for applications that face significant demand variations, such as e-commerce sites during peak sales periods (Erl, 2013). However, implementing horizontal scalability requires a well-designed software architecture and the use of automation and orchestration tools (Zimmermann, 2017).

Cloud architecture provides the resources and tools needed to implement scalability efficiently. This includes the ability to quickly provision and deprovision resources, load balancing, and performance monitoring. Cloud architecture must be designed to support horizontal scalability and ensure that systems can increase or decrease capacity as needed (Erl, 2013).

Software architecture must be designed to support scalability, ensuring that system components can function efficiently under variable workloads. This includes implementing performance optimization techniques such as caching and load balancing and using design patterns that facilitate scalability. Software architecture must also ensure that systems can adapt to changes in business and technology demands (Bass, Clements, & Kazman, 2012).

## MAINTAINABILITY AND EXTENSIBILITY

Software architecture must be designed to ensure the maintainability and extensibility of systems, allowing them to evolve and adapt to changes in business and technology needs (Bass, Clements, & Kazman, 2012). The literature review highlighted the importance of design patterns and coding practices that facilitate software maintenance and evolution over time. Companies that adopted these practices reduced maintenance time and costs and improved software quality (Richards, 2015).

Cloud architecture complements this need by offering a flexible infrastructure that can be easily adjusted and expanded as demands change. This includes the ability to quickly provision new resources and adjust system configurations as needed. Cloud architecture must be designed to support maintainability and extensibility, ensuring that systems can evolve efficiently (Erl, 2013).

Software architecture must ensure that system components are designed modularly and that coding practices facilitate maintenance and evolution. This includes implementing design patterns that support extensibility and using refactoring techniques to improve code quality. Software architecture must also ensure that systems can adapt to changes in business and technology demands (Bass, Clements, & Kazman, 2012).

## PERFORMANCE

Software architecture must be designed to ensure the maintainability and extensibility of systems, allowing them to evolve and adapt to changes in business and technology needs (Bass, Clements, & Kazman, 2012). The literature review highlighted the importance of design patterns and coding practices that facilitate software maintenance and evolution over time. Companies that adopted these practices reduced maintenance time and costs and improved software quality (Richards, 2015).

Cloud architecture complements this need by offering a flexible infrastructure that can be easily adjusted and expanded as demands change. This includes the ability to quickly provision new resources and adjust system configurations as needed. Cloud architecture must be designed to support maintainability and extensibility, ensuring that systems can evolve efficiently (Erl, 2013).

Software architecture must ensure that system components are designed modularly and that coding practices facilitate maintenance and evolution. This includes implementing design patterns that support extensibility and using refactoring techniques to improve code quality. Software architecture must also ensure that systems can adapt to changes in business and technology demands (Bass, Clements, & Kazman, 2012).

## INTEROPERABILITY AND USABILITY

Software architecture must support interoperability, allowing systems to integrate with other applications and services. This is particularly important in complex IT environments, where communication and collaboration between different solutions are essential (Rapôso, Costa Junior, & Ferreira, 2024). Additionally, usability and user experience are fundamental aspects that must be considered in software architecture. Companies that adopted user-centered design practices improved user satisfaction and efficiency (Bass, Clements, & Kazman, 2012).

Cloud architecture complements these aspects by providing an infrastructure that supports integration and collaboration between different systems. This includes the ability to connect different services and applications efficiently and securely, ensuring that users have an intuitive and efficient experience. Cloud architecture must be designed to support interoperability and usability, ensuring that systems can communicate and collaborate effectively (Erl, 2013).

Software architecture must ensure that system components are designed to support interoperability and usability. This includes implementing well-designed user interfaces and considering the needs and expectations of end users. Software architecture must also ensure that systems can integrate with other applications and services efficiently and securely (Bass, Clements, & Kazman, 2012).

## CONCLUSION

This study analyzed the divergences and convergences between software architecture and cloud architecture, highlighting the main challenges and opportunities that arise in integrating these two areas. The literature review and case studies revealed that complexity in integration, security, scalability, maintainability, performance, interoperability, and usability are critical aspects that need to be carefully managed to ensure the success of modern systems.

Software architecture plays a fundamental role in defining standards and guidelines that ensure the cohesion and efficiency of system components. It must be adaptable to incorporate new technologies and practices, such as cloud computing, and ensure that systems can evolve and adapt to changes in business and technology demands. Cloud architecture, in turn, provides the necessary infrastructure to support the distribution, scalability, and security of systems, allowing companies to fully leverage the advantages of the cloud.

Effectively integrating cloud computing solutions into existing software architectures requires a careful and well-planned approach. Understanding the interactions between these two areas is crucial for developing efficient, secure, and scalable systems. Companies that adopt user-centered design practices, performance optimization techniques, and robust security mechanisms can mitigate risks and improve the quality and efficiency of their systems.

The convergence between software architecture and cloud architecture offers numerous opportunities for innovation and improvement of systems but also presents significant challenges that need to be addressed with strategies and effective design practices. Collaboration between these two areas is essential for developing technological solutions that meet the growing demands for flexibility, scalability, and security.

To deepen the understanding of interactions between software architecture and cloud architecture, future work can explore the application of artificial intelligence and machine learning in optimizing cloud architectures. Empirical studies investigating the effectiveness of different design patterns in integrating microservices in cloud environments would also be valuable. Additionally, research analyzing the implementation of advanced security practices, such as zero trust and blockchain, in cloud architectures can provide important insights for improving data protection and regulatory compliance. Finally, case studies documenting the evolution of software architectures in companies that have adopted the cloud can offer practical examples and lessons learned for other organizations considering this transition.

## REFERENCES

1. Bass, L. (2012). Software architecture in practice. Pearson Education India.

2. Erl, T., Puttini, R., & Mahmood, Z. (2013). Cloud computing: concepts, technology & architecture. Pearson Education.

3. Fowler, M. (2012). Patterns of enterprise application architecture. Addison-Wesley.

4. Gorton, I. (2006). Essential software architecture. Springer Science & Business Media.

5. Hohpe, G., & Woolf, B. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley Professional.

6. Kruchten, P. (2004). The rational unified process: an introduction. Addison-Wesley Professional.

7. Lewis, J., & Fowler, M. (2014). Microservices: a definition of this new architectural term. MartinFowler. com, 25(14-26), 12.

8. Rapôso, C. F. L., da Costa Junior, O. G., & Ferreira, A. S. (2024). IMPACTOS DA COMPUTAÇÃO EM NUVEM NA ARQUITETURA DE SOFTWARE: UMA ANÁLISE DE LITERATURA. Revista Tópicos, 2(14), 1-12.

9. Richards, M. (2015). Software architecture patterns (Vol. 4, p. 1005). 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated.

10. Zimmermann, O. (2017). Microservices tenets: Agile approach to service development and deployment. Computer Science-Research and Development, 32, 301-310.