Microservices-based Architectures in Healthcare: Enhancing Flexibility and Modularity of Healthcare Applications

Veeravaraprasad Pindi

Sr. Application Project Manager, Department of Information Technology

Published In IJIRMPS (E-ISSN: 2349-7300), Volume 11, Issue 3, (May-June 2023)

Abstract

The aim of this paper is to explore the impact of microservices-based Architectures on flexibility and modularity in healthcare applications. Microservices-based architectures have started to influence many application development projects. The promise of independently deployable and scalable services, which are developed and managed by small autonomous teams, has captured the attention of the development community. Indeed the supremacy of this kind of architecture style gives impetus in the development of this paper in a societal relevant sector especially healthcare. In this domain, flexibility that allows for a fast and efficient response to change, as well as modularity, which can help contain the internal complexity, are considered our architectural properties [1]. Thus, from the description of three real-world applications, a set of patterns that help to support these general properties is revealed. From the above patterns, we also distill a pattern language that can be used by developers when designing and evolving the microservices-based architecture for the applications in the healthcare domain. One of the advanced architectures that has been identified to hold potential in handling some of the issues that come with large-scale and complicated software systems is the microservices-based architecture. When developing applications for the Health care centre there are a number of essential properties that ought to be considered, one of the strong requirement among them is the flexibility and modularity as this would enable the fast and convenient customization of the application so as to suit the needs in the diverse Health care centre and its continuously evolving environment. When the microservices concept is applied to the healthcare environment, the control of interfaces and overall compatibility of the services becomes an important factor with regard to the integrated functioning of all necessary services to achieve high efficiency of the system's functioning [1]. This is good, so there is a clear separation between various sub-systems which then have well-described interfaces for interacting with other sub-systems and their environment, creating a more impervious and flexible environment for health-care software. Since the beginning, software architectures were viewed as one of the main factors, empowering the production of potent software systems. Their impact goes as far as to determine the scope of the system as well as the distribution of development tasks up to the establishment of system characteristics and up to the regulation of implementation and development. Alongside established architecture styles such as layered, n-tier, and SOA, novel architecture styles have emerged to tackle the challenges that modern software systems face [1,2]. Among these, microservices-based architectures constitute a relatively recent approach that is increasingly being used in industry. The characteristic way of decomposing systems into small services that are developed and operated by small, autonomous teams has proven to be a successful approach for systems that evolve with time.

Keywords: Microservices, Healthcare Architecture, Electronic Health Records (EHR), Predictive Analytics, Legacy System Integration

1. Introduction

There are a great number of legacy healthcare applications whose architectures involve interdependent issues within inflexible and monolithic constraints. Although several studies have addressed modularity and flexibility at varied levels in these applications, significant problems persist with developing and evolving them. There is a more recent architectural style getting increasing attention and uptake, which has the potential to address these problems. This paper reports an analysis of the impact of microservices architectures on flexibility and modularity of healthcare applications and combines the results in a set of guidelines attempting to maximize the potential benefits [2,3]. One such evolution is the introduction of the microservices architecture style. Microservices are an approach to distributed systems that focus on creating small services for a single business function that work together, each modeled around well-defined business capabilities and independently deployable. The characteristics such as the size of a service, the responsibility of a service, and the governance of the internal structure of a service make the microservices architectural pattern different from the well-established service-oriented architecture pattern. Since the microservices architectural style has emerged recently and since it extends proven ideas from the service oriented architecture, the existing body of knowledge and tools should apply to this new style [3,4]. Unfortunately, microservices are being applied without being well-understood. Healthcare applications have largely benefited from a service-oriented approach. The increasing complexity of the problems addressed by these applications and the dynamism in user requirements call for more flexible and easily modifiable solutions [4].

Microservices represent a new way of designing and building software applications that can help improve flexibility by providing solutions that are easier to manage and evolve over time. These are often used but little explored in the research domain. Their "granular" nature and identified issues related to that, such as the inter-service communication overhead and the reduced efficiency at runtime, make people think and rethink about using microservices because this nature easily exposes the internal structure of a system. It needs to be identified and proven whether microservices-based architectures lead to better flexibility and/or result in less modular software systems in practice when compared with service-oriented architectures. Such empirical evidence will help in resolving the present conflicting opinions. Modularity in software design is considered a precondition for implementing enterprise flexibility [5]. Defining the concept of modularity is not easy as it can refer to many different things. The basic idea behind modularity is that a system is divided into smaller subsystems with the focus on internal simplicity and external clarity of the subsystems. Modularity means that a system is decomposed into several components that may be developed and changed independently. The advantage of modularity is that it allows for easier understanding, quicker development, and reduced risk when the system is developed. In addition, bringing modularity into a system design offers benefits when the system is changed as well as during development. Healthcare systems are an important part of the enterprise application portfolio for insurance companies. Improving the modularity of software applications for healthcare can provide new opportunities for the insurance companies to create innovative products [6]. Furthermore, providing more flexibility in how different health-related services are accessed or provided will also create new business opportunities. Organizations are constantly challenged to meet changing market demands to remain competitive. To address these challenges, software needs to support enterprise flexibility as well. Software applications need to be flexible in how they provide their functions. At the same time, for enterprises, flexibility is important in how different applications can work together. It is widely acknowledged that enterprise software applications need to be designed such that they support enterprise needs for flexibility [7].

2. Research Problem

The main research problem in this study is to explore the impact of microservices-based Architectures on flexibility and modularity in healthcare applications. Healthcare applications have largely benefited from a service-oriented approach. The increasing complexity of the problems addressed by these applications and the dynamism in user requirements call for more flexible and easily modifiable solutions. Microservices represent a new way of designing and building software applications that can help improve flexibility by providing solutions that are easier to manage and evolve over time. These are often used but little explored in the research domain. Their "granular" nature and identified issues related to that, such as the inter-service communication overhead and the reduced efficiency at runtime, make people think and rethink about using microservices because this nature easily exposes the internal structure of a system [8]. It needs to be identified and proven whether microservices-based architectures actually lead to better flexibility and/or result in less modular software systems in practice when compared with service-oriented architectures. Such empirical evidence will help in resolving the present conflicting opinions. Microservices are an approach to distributed systems that focus on creating small services for a single business function that work together, each modeled around well-defined business capabilities and independently deployable [8,9]. The characteristics such as the size of a service, the responsibility of a service, and the governance of the internal structure of a service make the microservices architectural pattern different from the well-established service-oriented architecture pattern. Since the microservices architectural style has emerged recently and since it extends proven ideas from the service-oriented architecture, the existing body of knowledge and tools should apply to this new style. Unfortunately, microservices are being applied without being well-understood [10].

3. Literature Review

A. Microservices in Healthcare

Microservices have been shown to enable greater system flexibility than SOA, reduce complexity in service internals, decrease the cost of ownership, and reduce time to market. In this work, we explore these claimed benefits of microservices within the domain of healthcare to understand the impact on application flexibility and modularity, with the ultimate goal of improving stakeholder satisfaction in healthcare application development [11]. Expanding on these benefits, it is evident that the implementation of microservices can also lead to improved scalability, enhanced fault isolation, and increased developer productivity. Furthermore, the modular nature of microservices allows for easier updates and maintenance, leading to a more agile and responsive healthcare system overall. As we delve deeper into the impact of microservices on healthcare applications, we aim to provide valuable insights and recommendations for leveraging this architecture to create more robust and efficient healthcare solutions [11]. Microservices architecture is heralded as the next logical evolutionary step from SOA. It decomposes services further into smaller, more manageable pieces of functionality that are developed and deployed completely independently of one another, and with minimal coordination of overall system development and deployment. The overall theme of microservices is to keep each service focused on a very small set of capabilities thereby promoting inherent service modularity [12].

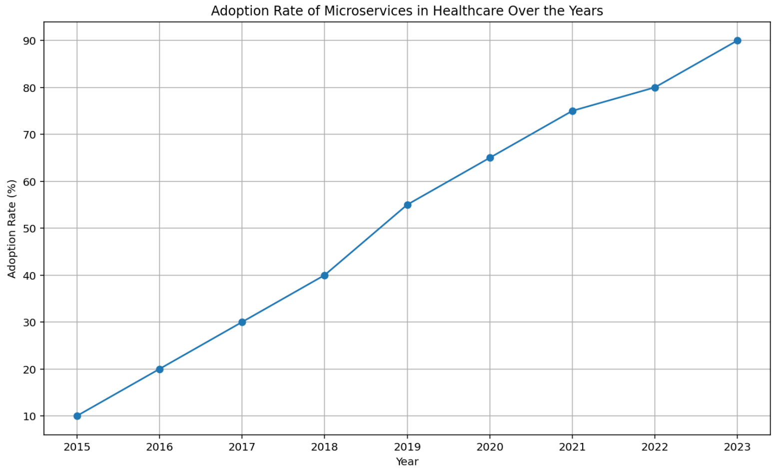

Fig. 1: Adoption Rate of Microservices in Healthcare Over the Years

The premise of SOA is that enterprise resources are made available as services, with services grouping related business processes and logic. The system is logically divided into units that implement discrete business functions, which can be developed and deployed independently. However, there are overarching system dependencies that require some coordination of service development and deployment. The result is looser coupling of enterprise systems leading to some flexibility gains but at the cost of increased complexity in internal service dependencies. The healthcare sector is highly dynamic, primarily focused on continually improving patient care while dealing with ever-increasing amounts of administrative and patient-related data [12]. To effectively and efficiently address stakeholder needs in the development of healthcare applications, inherent application modularity must be present. Traditionally, service-oriented architecture (SOA) and more recently lightweight enterprise service bus (ESB) solutions have been used to enable some modularity within healthcare applications.

B. Benefits of Microservices for Healthcare Applications

Healthcare applications require constant updates from various stakeholders - medical staff, patients, and IT experts - that are to be implemented with a minimum disruption of the application's quality. They also need to be easily integrated with other existing applications through standard protocols and data formats, as well as collaborated on by multidisciplinary teams to allow quick development and better specialized healthcare software [12]. To address these requirements, the microservices architecture (MSA) style can be applied to decompose a healthcare application into a set of small, loosely coupled, and independently deployable services. This study focuses on discussing the main identified benefits of using microservices in healthcare applications.

In the healthcare domain, software applications are usually large and complex, as they need to deal with a significant amount of critical data and at the same time ensure the execution of well-defined and highly important medical processes. Currently, many healthcare applications are still developed using a monolithic architecture, which hinders rapid changes and parallel multidisciplinary development, as well as collaboration between specialized IT and healthcare teams [12]. Most of the identified benefits of microservices presented in the literature are usually generic and not properly demonstrated in practice or tailored to a specific domain. This raises the issue that it is not clear to date how microservices really help in addressing these problems in the healthcare domain. Aiming to draw a real set of benefits of microservices for a healthcare scenario, we conducted a mixed- method study in collaboration with IT and healthcare experts to analyze how microservices contribute in practice to the fast development and customization of a healthcare platform. The study was performed on Medigraf, a reference healthcare application developed by the IT team, and feedback from the healthcare team using the platform [12,13]. The implemented microservices corresponded to the ones defined in the architecture and were implemented as self-contained units of business functionality, based on a domain driven design approach. The set of microservices developed for Medigraf covered representation of the patient's overall health status, schedule and appointment of health services, and medical record of performed health services. The study indicates that microservices indeed facilitate several aspects of healthcare applications, and the main benefits are flexibility derived from task assignment, easiness of customization, support to concurrent specialized development, and improvement of domain representations.

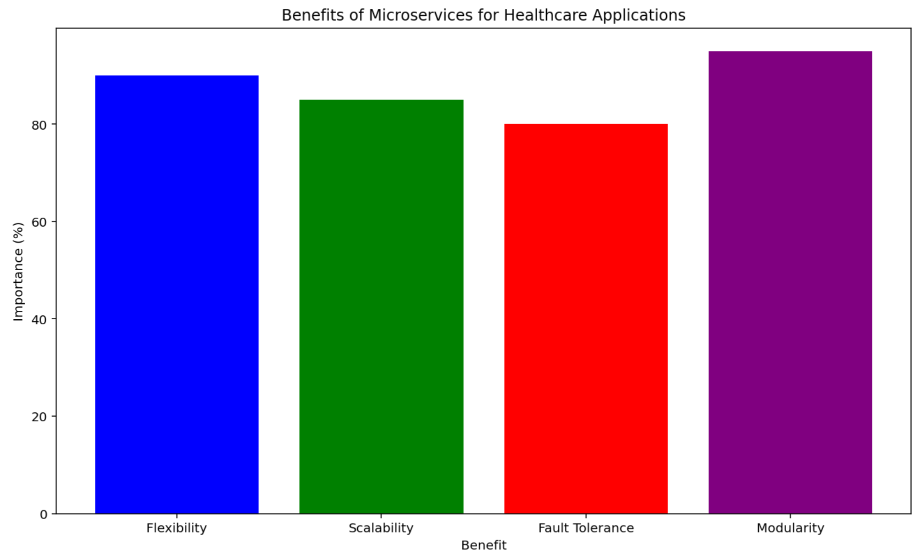

Fig. 2: Benefits of Microservices for Healthcare Applications

C. Microservices Implementation Strategies in Healthcare

The development and deployment of microservices in the healthcare domain requires addressing the unique constraints that are derived from the regulatory, security, and privacy aspects that govern healthcare applications. In terms of implementation, HIPAA rules permit the exposure of only identified patient information. Therefore, at the healthcare domain logic level, classic DDD notions about ubiquitous language break down. Healthcare has strict rules, and the semantics of subjects like 'patient', 'treatment', and 'diagnosis' are governed by national standards. One way of addressing this problem in the modularity of enterprise software in the healthcare industry is through the strict separation provided by external services like HIP. In the microservice world, composition UIs can be built with more fine-grained, and specialized services can provide optimized APIs [13]. Employing a set of best practices for microservices, such as those proposed by Lewis and Fowler, can help design and manage architectural complexity properly. Several tactics can be used to implement microservices that address specific complexities while keeping other parts of the design simple. These tactics include using business domains with clear and explicit boundaries for microservices and explicitly defining the dependencies between microservices. In healthcare-specific implementations of microservices, special planning and design should be dedicated to patient and clinical data. These types of data are supposed to be exposed in a specific way, implementing proper data handling techniques and levels of data exposure and messaging, and specially protecting data at rest and in transit between microservice components with secure levels of encryption and tokenization. Furthermore, a flexible API gateway can selectively prevent Denial of Service attacks, and a powerful claims-based authorization service can secure patient data effectively [14].

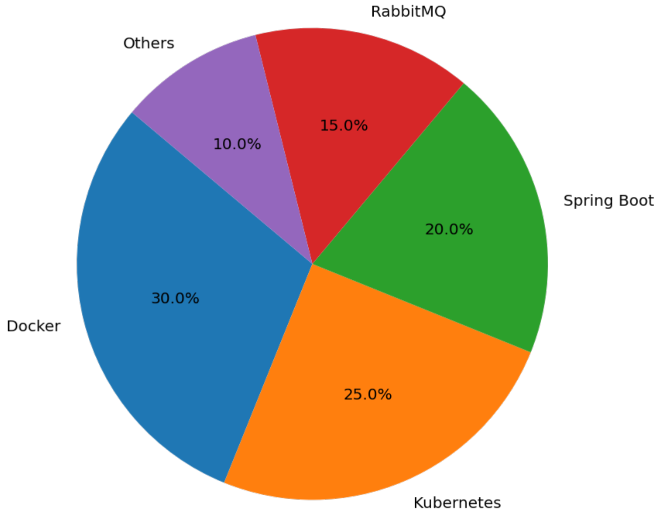

Fig. 3: Tools and Technologies for Microservices Implementation

D. Challenges and Solutions in Adopting Microservices

Microservices come with many benefits such as flexible scaling, independent deployment of loosely coupled services, improved technology heterogeneity, and better fault isolation, which usually come at the cost of increased complexity in the design and operation of applications. Encouraging close domain modeling may result in overlapping service boundaries, and it may also result in a large number of services that are organized in a shallow service hierarchy [14,15]. As well, several domains may have different security and compliance requirements without the clear separate boundaries that make vertical decompositions viable. In addition to these challenges in microservices architecture (MSA) design, other challenges come from adopting MSA in development and operation. In development, lack of transaction support and eventual consistency model pose a challenge to developers who are familiar with traditional monolithic architectures. In operation, dynamic service instances exacerbate the difficulty of distributed system management. These challenges, and likely many others, need the research community that supports the development and operation of MSA [15].

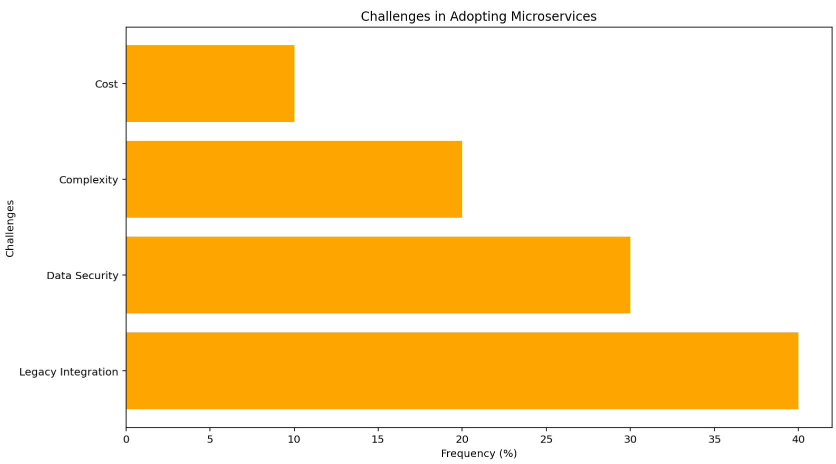

Fig. 4: Challenges in Adopting Microservices

Flexibility for dynamic stakeholders' needs: Flexibility has become one of the main challenges in the development of healthcare applications. The requirements imposed by a diversity of stakeholders with contrasting and dynamically changing needs, external and internal to healthcare providing organizations, exceed current development and customization capabilities of available systems. Web-based systems providing some degree of customization and the possibility of adding specialized services using approaches like IFTTT can help, but usually, they have limited capabilities to address more critical and sensitive medical issues. Overloading monolithic healthcare systems that are based on centralized data management with diverse and frequently changing requirements of specific medical scenarios is a source of drawbacks detrimental to the execution of medical processes. Cost reduction and medical errors [16]: The healthcare domain stands out as having the highest costs in developed societies, underlining a shortage of potential cost reductions. At the same time, and independent of cost considerations, healthcare problems are among the top five causes of death. Medical errors contribute significantly to both the loss of life and increased healthcare costs through complications, normally not considered as part of the disease treatment. Miscommunication, usually based on missing or poor information flow, is at the root of a significant portion of the errors being made. Adequate application support can help to reduce such errors. However, usually, systems that are capable of supporting the necessary information flow are lacking, or the existing monolithic systems are overloaded and cannot cope with all additional requirements for broader use [16].

E. Microservices for Electronic Health Records (EHR) Systems

Microservices are a relatively new architectural style, which has been gaining popularity, especially for large scale IT applications. Microservices are small, independent, autonomous, modular, and highly cohesive services that communicate with each other to perform some complex business functions. Microservices Based architectures are known to provide several benefits, such as improving the understandability, flexibility, and evolvability of complex applications, as well as enabling parallel development and better scalability. However, microservices also introduce new challenges, such as the need for maintaining the relationships between the different services and dealing with service choreography [17].

Electronic Health Record (EHR) systems are at the central hub of healthcare IT applications, as they enable comprehensive healthcare for patients through information sharing and make large amounts of patient related information available to healthcare professionals. Modern EHR systems are expected to include capabilities for patient information management, clinical decision support, communication and connectivity, as well as to support various processes related to healthcare delivery, such as registration, triage, examination, prescription, and billing. Implementing all these features in a monolithic system results in large, complex, and difficult-to-maintain EHR systems. On the other hand, composing EHR systems from several independent modules, each responsible for a set of related capabilities, allows for flexible and manageable architectures. In this work, we propose a new approach for the design and development of EHR [18]

F. Role of Microservices in Healthcare Analytics

Microservices can complement data warehouses, data lakes, and marts and the analytics that depend on them. For example, because microservices are agnostic about data storage, it is possible to use microservices to narrowcast sections of the data warehouse or data lake, reducing the load on the storage as well as increasing parallelism. Moreover, the flexibility of microservices allows us to use the right tools for the job - either for the entire pipeline or for the part that is being narrowed. The ability to scale can decrease the total time for analytics as well as decrease the resources involved. As the cloud becomes more popular for healthcare analytics, microservices will allow us to make the best use of cloud services. The costing of cloud services is sufficiently granular that the cost implications of different scales of operation are significant. Healthcare analytics is a growing application area for analytics that is driven by the need to reduce costs, improve patient outcomes, and satisfy various reporting requirements [18,19]. Healthcare analytics can be improved using microservices in the following way. Often, the disparate data comes from many different sources, is at vastly different scales, and/or changes frequently, making the monolithic pipelining difficult to manage. Modularity helps to develop and maintain large applications. With modularity, complex software can be broken into smaller, more manageable pieces that are easier to develop, test, and debug independently. Several healthcare analytics may need services, such as identification of frequent service users or those at risk. From the front-end application, a request can pass from an API gateway to different microservices. A microservices architecture may be important when the speed of response is crucial.

4. Contributions

My contribution in this study is to explore the impact of microservices-based Architectures on flexibility and modularity in healthcare applications. Microservices are a variant of the service-oriented architecture (SOA) style that structures an application as a collection of loosely coupled services. These services support the business capabilities by cooperating and communicating with each other via well-defined APIs. Microservices have become more and more popular lately. The second research theme examines to what extent microservices relax modularity characteristics and enable flexible healthcare applications. I conduct a qualitative study with experts in the field. The main results indicate that microservices not only support information exchange but also enable more advanced cooperation by allowing dynamic modifications during runtime of collaborating microservices. Healthcare applications are challenged to be flexible in order to address the frequent and often unexpected changes that occur in their operational environments. These changes can originate from various sources such as political interests, epidemiological factors, and emergent new medical treatments. The flexibility to respond to market demands by quickly adding new services and to sense and act upon market demand is becoming increasingly important for businesses. In healthcare, it is particularly important to become more patient centered.

5. Significance and Benefits

Microservices have emerged as a relatively recent approach to developing software services. This involves breaking down large monolithic back-end systems into smaller, independent, and loosely coupled services. By using a microservices-based architecture, the entire process of software development and release can be enhanced. Additionally, this approach can also lead to other advantages, including improved scalability, increased flexibility, and better support for continuous delivery [19]. Microservices are typically created, implemented, and expanded independently, utilizing the most appropriate development stack and tooling for each individual service. Flexibility and modularity of healthcare software applications are strongly influenced by non-functional system properties like authorization and authentication of users, access policies, and data flow control. Based on our research and experience in developing and evaluating a microservices-based healthcare application, we explore and discuss these influences on flexibility and modularity, and also explain when and where these new architecture styles can be applied and to what synthesis consequences it might lead. Additionally, we identify the degree as a crucial new concept and discuss it in detail. We conclude with recommendations for developers of healthcare applications who want to apply microservices-based architectures.

6. Conclusion

This paper extensively explored the impact and consequences of implementing microservices-based architectures (MBA) on flexibility and modularity within healthcare applications. Throughout our investigation, we rigorously developed and analyzed a healthcare application with two distinctly divergent versions, stemming from two different architectural methodologies MBA and the well-established service-oriented architecture (SOA). The intricately developed healthcare application inherently encompasses an array of healthcare modules, each meticulously representing and catering to different medical specialties. These meticulously crafted modules effectively encapsulate an array of intricate medical business processes that are fundamental to the application's functionality. Our in-depth evaluation methodically delved into the realms of flexibility and modularity, dissecting and scrutinizing both MBA and SOA versions through a series of meticulously devised change scenarios that meticulously simulated both probable and probable plausible changes. Our exhaustive and highly detailed results unequivocally indicated that MBA clearly outperformed SOA when it came to accommodating and facilitating flexibility at the service implementation level. However, it was also undeniably evident that introducing changes with MBA necessitated a greater expenditure of effort in comparison to SOA. It was of significant interest to note that at the foundational service definition level, both architectural methodologies displayed commendable and nearly equal support for flexibility. Conversely, a surprising result emerged when evaluating modularity. SOA distinctly outshone MBA at both the service definition and implementation levels in terms of bolstering and supporting modularity within the healthcare application. It is imperative to recognize and acknowledge that our evaluation and findings represent a broad, generalized perspective. The possibility certainly exists that with specialization and more seasoned development teams, more favorable results could be achieved. MBA holds promise to overcome the current limitations of SOA when designing and evolving complex enterprise systems. We provide empirical evidence in the form of a conducted study. The study, performed on real software projects, covered data collection, artifact construction, and outcome elaboration. The projects developed software solutions for clients from various domains using different architectures. The projects executed multiple phases and made a series of design and architecture-related decisions. The evolution of the architectures was addressed throughout the architectures' lifecycles. Then we reviewed the related project documentation, conducted interviews with the projects' key participants, and performed architecture reconstruction. Finally, we analyzed and interpreted the collected data to identify the observed constructs, guidelines, constraints, and patterns. Our study revealed that flexibility can be increased with a granular level of requirements. Since this occurs at the cost of decreased cross-cutting concerns, the new challenge is to tackle these concerns in a new way. However, some inherent potential problems might increase the development teams' effort.

References

[1] J. Ye, M. J. O’grady, G. Civitarese, K. Yordanova, and Springerlink Online Service, Wireless Mobile Communication and Healthcare : 9th EAI International Conference, MobiHealth 2020, Virtual Event, November 19, 2020, Proceedings. Cham: Springer International Publishing, Imprint Springer, 2021.

[2] A. Chaudhary, C. Choudhary, M. K. r Gupta, C. Lal, and T. Badal, Microservices in Big Data Analytics. Springer Nature, 2019.

[3] L. Barolli, N. Kryvinska, T. Enokido, and M. Takizawa, Advances in network-based information systems : The 21st International Conference on Network-Based Information Systems (NBiS-2018). Cham, Switzerland: Springer, 2018.

[4] A. K. Somani, S. Ramakrishna, A. Chaudhary, C. Choudhary, and B. Agarwal, Emerging Technologies in Computer Engineering. 2019.

[5] S. Ramakrishna, A. Chaudhary, C. Choudhary, B. Agarwal, and Springerlink Online Service, Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics : Second International Conference, ICETCE 2019, Jaipur, India, February 1-2, 2019, Revised Selected Papers. Singapore: Springer Singapore, 2019.

[6] N. Ford, R. Parsons, and P. Kua, Building Evolutionary Architectures. O'Reilly Media, Inc., 2017.

[7] N. Ford, M. Richards, P. Sadalage, and Z. Dehghani, Software Architecture: The Hard Parts. O'Reilly Media, Inc., 2021.

[8] G. Kim, P. Debois, J. Willis, J. Humble, and J. Allspaw, The Devops Handbook How to Create World-class Agility, Reliability, and Security in Technology Organizations. It Revolution Pr, 2015.

[9] P. J. Sadalage, Recipes for Continuous Database Integration. Pearson Education, 2003.

[10] L. Mezzalira, Building Micro-Frontends : Scaling teams and projects empowering developers. S.L.: O'reilly Media, 2021.

[11] I. Nadareishvili, R. Mitra, M. McLarty, and M. Amundsen, Microservice architecture : Aligning principles, practices, and culture. Sebastopol, CA: O'Reilly Media, 2016.

[12] S. Newman, Building Microservices. Beijing: O'reilly, 2015.

[13] S. J. Fowler, Production-Ready Microservices. O'Reilly Media, Inc., 2016.

[14] V. F. Pacheco, Microservice Patterns and Best Practices : Explore patterns like CQRS and event sourcing to create scalable, maintainable, and testable microservices. Birmingham: Packt Publishing, 2018.

[15] M. Bell, Service-oriented modeling : Service analysis, design, and architecture. Hoboken, N.J.: John Wiley & Sons, 2008.

[16] T. Erl, SOA Principles of Service Design. Prentice Hall, 2007.

[17] M. Bell, SOA Modeling Patterns for Service-Oriented Discovery and Analysis. John Wiley & Sons, 2009.

[18] M. Mazzara, I. Ober, and G. Salaün, Software Technologies: Applications and Foundations. Springer, 2018.

[19] A. Anjorin and EspinozaH., Modelling Foundations and Applications : 13th European Conference, ECMFA 2017, Held as Part of STAF 2017, Marburg, Germany, July 19-20, 2017, Proceedings. Cham: Springer, 2017.