iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XI **Month of publication:** November 2021

DOI: https://doi.org/10.22214/ijraset.2021.39036

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue XI Nov 2021- Available at www.ijraset.com

Microservices Software Architecture: A Review

Chitrak Vimalbhai Dave[1], Abhishek Patel[2], Utkarsh Keshri[3]
[1, 2, 3]*Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India*

Abstract: ***Architecting is an indispensable activity in all spheres and paradigms of Software Engineering. DevOps, a portmanteau of Development and Operations, has a major adoption challenge in context to Microservices Architecture. Architecture refers to the most important aspect of internal design of a software. Architecture must be good otherwise; it will become slow and much more expensive to add new proficiencies in future. This paper has presented a review of microservices architecture and implementation patterns. Microservice approach is a new concept in software architecture patterns and has leapt up over past few years for describing a certain method to design software applications as collections of individualistically deployable services. In this paper we are looking to discover the role of microservices software architecture in DevOps. It is found that adopting DevOps may carry many challenges/issues like organizational values, immaturity of the tools and infrastructural support with it for architecting with it.***

Keywords: Microservices Architecture, Microservice, Service Oriented Architecture (SOA), Software Architecture, DevOps

I. INTRODUCTION

Microservice software architecture is a design pattern of SOA (Service oriented architecture) which are more independent as it shares fewer resources, and communicate with each other. In SOA, individual components are tightly coupled and communicate through a piece of specialized software called an enterprise storage bus. Microservice architecture is a successor design pattern of SOA. Microservice is a process of implementation of Software-oriented architecture by dividing the complete application into interconnected services and each service will serve particular business need [1]. Research on Microservices has taken into account both the architectural aspects and support as well as application of architectural pattern in cloud. In cloud, Microservices are linked to containers which area virtualization mechanism used for application packaging, distribution at PaaS layer. In Software-oriented architecture, complete software packages are divided into different small business units which are interconnected. Each unit will communicate with one another using some protocols in order to deliver successful business to the client. In this paper, we have discussed about Microservice software architecture and how it differs from Software oriented architecture. Software oriented architecture is a design pattern and Microservice is an implementation methodology. The different Software architecture patterns are given beneath.

1) Layered architecture
2) Event driven architecture
3) Microkernel architecture
4) Microservices architecture
5) Space based architecture

Microservices architecting style can be better explained by its comparison with Monolithic architecting style. [11] In Monolithic style, all the request handling logic runs in a single process, it allows the application to be divided into classes, functions and namespaces using the elementary features of your language. An application can be run and tested on a developer's laptop with care, and make use of delivery pipeline to make sure that improvements are thoroughly tested and then put into production. Microservices architecture has certain common features, such as Services componentization, is constructed around business skills, products and not projects, decentralized Governance, decentralized governance, decentralized data management, evolutionary design. [12] Microservice architecture methodology uses the service-oriented architecture with the best practices and developments in software virtualizations to overcome the issues. Microservices have some benefits like strong module boundaries Microservices support modular structure, that is specifically essential for bigger teams. When team size increases, keeping it balanced becomes tighter, in such a case retaining module boundaries, separate deployment becomes essential Simple systems are easier to manage and less likely to trigger malfunctions as and when error arises in them, as they are autonomous. With a lot of microservices that need to be deployed frequently, it is essential that you act together on deployment. That is why the rapid deployment of applications and the rapid provision of infrastructure are prerequisites for Microservice.

You need to do a continuous delivery for anything beyond the basics, technology diversity with Microservices multiple languages can be mixed with software frameworks and data storage technologies with microservices. Microservices can be written in various languages, using various libraries and using separate data warehouses. It permits teams to select a suitable tool for certain job, some languages and libraries are best suited to some type of issues. Microservices comes with limitations as well such as distributed programs are firmer to program, as remote calls are sluggish and always face failure, maintaining many programming languages and frameworks poses a challenge, existing methods are likely to be inconsistent with new dependence on operation, every service requires individualized testing and monitoring, which ensures that the automation systems must also be taken into account by organizations, achieving data consistency is a real challenge, because each provider has its own database and management system for transactions**.** To better explain Microservices you need to get idea about Monolithic approach. Monolithic applications can be successful, but people are frustrated with them especially as more applications are implemented in the cloud. Change cycles are connected together, a change made to a specific part of the application, involves a redesign and implementation of the entire monolith. These concerns contributed to the architectural style of microservice: building applications as service suites. As well as being independently deployable and scalable, each service also offers a robust module boundary, thus allowing for the writing of different services in different programming languages. Any architectural style has trade-offs: strengths and weaknesses that we need to measure according to the context in which they are used. Of course, this is the case with microservices. Though it is useful architecture but many situations would do better with a Monolith.

II. DEVOPS AND MICROSERVICES

DevOps is set of activities that integrate the creation of application [13]. Dev and Ops together seek to minimize the life cycle of application development and provide a better quality software. Technology teams of all sizes are following the DevOps concept by implementing the microservice model to produce applications rapidly. [14] DevOps is a progress of agile movement. Trends in industry clearly show that adoption of microservices is expanding very fast. Microservices aren't without obstacles, though. Large scale implementation calls for autonomous concepts and strategies to be applied, that leads to increased complexity and risk of loss. We would like to make clear why Microservices are important for the future of DevOps. [15] Adopting DevOps "ideology," different software departments are responsible for different facets of the end application, allowing both the department and the software kernel to develop, test, handle errors and evolve individually.

III. LITERATURE REVIEW

Dmitry Namiot et al., [2] [2016] has discussed the basic principles behind this approach, its pros and cons This paper has presented the overview of the architecture and deployment patterns for micro-services. The authors have concluded that a Microservice is a lightweight, independent service which uses a well-defined interface to perform single functions and collaborate with other similar services. Claus Pahl and Pooyan Jamshidi [3] [2016] have intend to describe, taxonomically characterize and systematically compare the current body of research on microservices and their cloud application. They have performed a systematic mapping study of 21 selected articles, reported from the advent of the microservices model until the end of 2015 over the last two years. Authors have categorized and compared the selected studies based on a structure for characterization. This resulted in a review of the accepted and evolving issues within the architectural style of microservices, placing them within a framework of continuous development but also bringing them closer to cloud and container technology. Nabor C. Mendonc et al., [4] [2019] have suggested ways to address several of these challenges, based on existing and emerging microservice trends and technologies. In this paper, authors have taken a thorough look at the interaction between self-adaptive systems and microservices to recognize the challenges as self-adaptive systems when designing microservice applications. This article has addressed challenges in designing microservice applications as self-adaptive systems, using as a motivating example a cloud based, intelligent video surveillance device. It also proposes potential new paths by incorporating current microservice activities and technologies to address most of the listed challenges. Lianping Chen [5] [2018] has acknowledged new challenges related to the growing number of services, changing service contracts, technology diversity and testing. He has shared the realistic approaches which can be used to tackle these challenges, explore circumstances that might not be a decent choice for Microservices, and identifying the areas that need to be studied more. In this article, He has reported the main advantages of using Microservices, explained new challenges that emerged from its use, and outlined the realistic approaches that can be taken dealing with these challenges, addressing circumstances for which Microservices may not be a decent choice and identifying the areas for future study. Trihinas, Demetris et al. [6] [2018] authors have indicated that through embracing microservices, teams are able to build and deploy distributed, cloud - based applications fast. The use of Microservices for cloud computing industry is a new standard which is also becoming a trend in IoT.

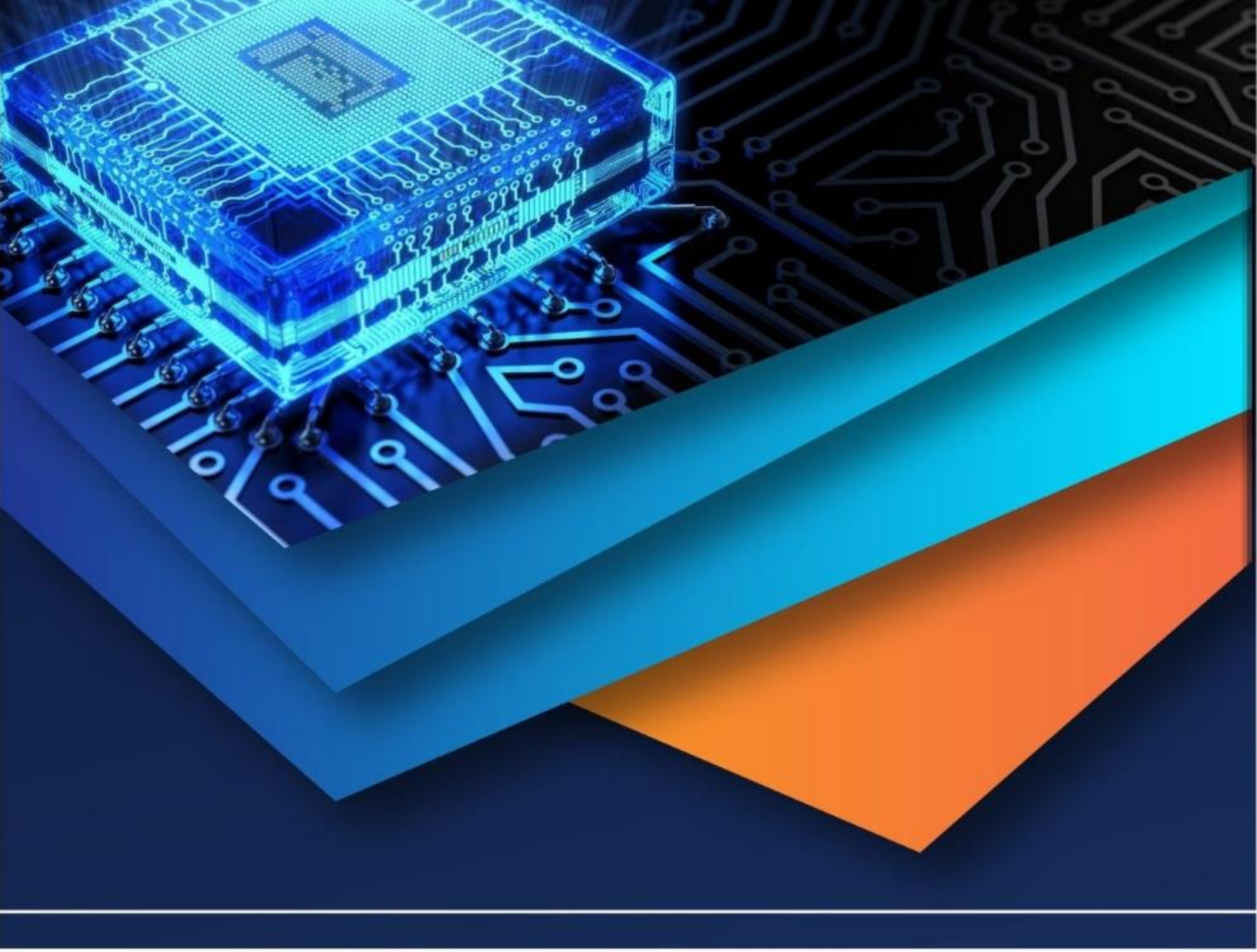
As many companies accept the microservice model and move their systems into the cloud. Len Bass [7] [2018] has focused on the technical aspects of the software architect with respect to activities include four main concerns that are entering into development easily, finding faults by automated checking, reducing or removing errors that occur during deployment, recovery and repair of system faults quickly. G. Márquez, F. Osses, and H. Astudillo [8] [2018] have reviewed that profits 44 patterns of architecture in academic sources and 74 in industrial patterns of architecture, as well as some architectural tactics that are proposed to address related problems. This paper has described an organized review of academic and industrial literature regarding architectural patterns and architectural tactics of Microservices. Armin Balalaie and Abbas Heydarnoori [9] [2016] have described the insights and lessons learnt as a service to Microservices architecture during steady migration and technical refactoring of a commercial mobile back end. Article has described how the researchers supported DevOps and how a migration was facilitated smoothly. Muhammad Waseem and Peng Liang [10] [2017] has aimed to get a thorough understanding of the challenges and the solutions, definition of design, migration, refactoring, and evolution in Microservices software architecture with DevOps. Problems of using Microservices Architecture (MSA) in DevOps were discussed in this study.

IV. CONCLUSION

After analyzing various software architectures, we have concluded that Microservices and DevOps are the two trends that are emerging in enterprise. Both the approaches are designed to offer great agility & efficiency. According to the architects, DevOps is an important component of microservices. Microservices approach of architecting from a set of DevOps that came into existing at some companies like Amazon, Facebook, Netflix, Google and many more. Microservices software architecture adds further productivity to DevOps by embracing few toolsets, which are used for the development and operations. That toolset consists of common terminologies, processes for the requirements, dependencies & issues which make it easier for the Devs & Ops to work with each other. DevOps and Microservices are better together.

REFERENCES

[1] Pahl, Claus, and Pooyan Jamshidi. "Microservices: A Systematic Mapping Study." In CLOSER (1), pp. 137- 146. 2016.
[2] Namiot, Dmitry, and Manfred Sneps-Sneppe. "On micro-services architecture." International Journal of Open Information Technologies 2, no. 9 (2014): 24-27.
[3] Pahl, Claus, and Pooyan Jamshidi. "Microservices: A Systematic Mapping Study." In CLOSER (1), pp. 137- 146. 2016.
[4] Mendonça, Nabor C., Pooyan Jamshidi, David Garlan, and Claus Pahl. "Developing Self-Adaptive Microservice Systems: Challenges and Directions." IEEE Software (2019).
[5] Chen, Lianping. (2018). Microservices: Architecting for Continuous Delivery and DevOps. 10.1109/ICSA.2018.00013.
[6] Trihinas, Demetris, Athanasios Tryfonos, Marios D. Dikaiakos, and George Pallis. "Devops as a service: Pushing the boundaries of microservice adoption." IEEE Internet Computing 22, no. 3 (2018): 65-71.
[7] L. Bass, "The Software Architect and DevOps," in IEEE Software, vol. 35, no. 1, pp. 8-10, January/February 2018.
[8] Marquez, Gaston, Felipe Osses, and Hernan Astudillo. "Review of architectural patterns and tactics for microservices in academic and industrial literature." IEEE Latin America Transactions 16, no. 9 (2018): 2321- 2327.
[9] Balalaie, Armin, Abbas Heydarnoori, and Pooyan Jamshidi. "Microservices architecture enables devops: Migration to a cloud-native architecture." Ieee Software 33, no. 3 (2016): 42-52.
[10] Waseem, Muhammad, and Peng Liang. "Microservices Architecture in DevOps." In 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), pp. 13-14. IEEE, 2017
[11] Ahmadvand, Mohsen, and Amjad Ibrahim. "Requirements reconciliation for scalable and secure microservice (de) composition." In 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), pp. 68-73. IEEE, 2016.
[12] Butzin, Björn, Frank Golatowski, and Dirk Timmermann. "Microservices approach for the internet of things." In 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1- 6. IEEE, 2016.
[13] Jha, Pratibha & Khan, Rizwan. (2018). A Review Paper on DevOps: Beginning and More to Know. International Journal of Computer Applications. 180. 16-20.
[14] Leite, Leonardo, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. "A Survey of DevOps Concepts and Challenges." ACM Computing Surveys (CSUR) 52, no. 6 (2019): 1-35.
[15] Bucena, Ineta, and Marite Kirikova. "Simplifying the DevOps Adoption Process." In BIR Workshops. 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY