



ISSN: 0975-833X

Available online at <http://www.journalcra.com>

*International Journal of Current Research*  
Vol. 12, Issue, 05, pp.11805-11808, May, 2020

DOI: <https://doi.org/10.24941/ijcr.38524.05.2020>

**INTERNATIONAL JOURNAL  
OF CURRENT RESEARCH**

## RESEARCH ARTICLE

### A WORLD BETWEEN COMPONENTS AND MICROSERVICES – DIFFERENT POINTS OF VIEW OF THE RELATION BETWEEN THEM

**\*Radu Irina-Miruna**

The Bucharest University of Economic Studies, Bucharest, Romania

#### ARTICLE INFO

##### Article History:

Received 10<sup>th</sup> February, 2020  
Received in revised form  
19<sup>th</sup> March, 2020  
Accepted 27<sup>th</sup> April, 2020  
Published online 31<sup>st</sup> May, 2020

##### Key Words:

Component-Based Software Development,  
Component Software Engineering, CBSD,  
CBD, Microservices.

#### ABSTRACT

When talking about a complex application, the first thought of the majority of developers is splitting the logic into pieces. For building the smaller parts, nowadays micro services architecture is a fast-growing popular choice. But there was a time when people used to rely on a component-based development approach to satisfy the need for building components. Something happened during this interval that shifted the point of view. This article proposes to find out the relation between micro services architecture and component-based architecture. The focus of this research is towards finding how micro services and components are perceived in the opinions of scientists and IT professionals as they are exposed online.

*Copyright* © 2020, **Radu Irina-Miruna**. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Citation:** **Radu Irina-Miruna**. 2020. "A world between components and microservices – different points of view of the relation between them ", *International Journal of Current Research*, 12, (05), 11805-11808.

## INTRODUCTION

The evolution of technology has brought about many breakthroughs that are changing the way we live. There is a continuous need to innovate and technologize, using less time and less effort. Due to the need to be closer to and cover more of the needs of the user, the applications are becoming exponentially more complex than the ones already available on the market. In order to simplify the process of development it is necessary to split the bigger systems into smaller parts. These can cover smaller logics, are more easily unit tested and help the development team to work at the same time on different topics without having to understand the whole system when handling changes of certain parts. There are a few solutions available for this problem but the ones that make the subject of this research paper are component based-software architecture and microservices architecture. The above analysis was obtained by conducting a search on Google Trends regarding how the topics of component-based software engineering and microservices were covered between 2004 and May 2019 (Fig. 1). The results show that in 2004 component-based software engineering was a frequently looked up term in Google's search engine.

During the subsequent years the interest in this topic dropped and by July 2014 it was surpassed by the rising interest in microservices. Today we can observe that the topic of microservices has overtaken the world's attention from the component-based software approach. Starting from this discovery the aim of the current paper is to find the correlation between the disappearance of component-based software architecture and the emergence of microservices. To achieve the goal of this quantitative study the paper is structured into four main parts that cover the way components are viewed in regards to microservices. The first two parts present details regarding components and microservices, shedding light on the principles behind them. The third part describes the way data was gathered for this quantitative study, offering insight into the points that were taken into consideration into this analysis. The last part outlines the results of the research aggregated and presented along with some mentionable examples that illustrate them. The study will be an eye-opener towards how the relationship between components and microservices is seen through the eyes of the specialists as their opinions emerge from scientific articles, online opinions and answered questions.

**Understanding Component-based architecture:** A component is a piece of software that encapsulates certain functionalities to fulfill a clear purpose. Just like a Lego piece, a component should be able to interact according to a well-defined interface and behavior to other components within the system (Petritsch, 2019)

**\*Corresponding author: Radu Irina-Miruna,**  
The Bucharest University of Economic Studies, Bucharest, Romania.

The idea of components appeared more than fifty years ago when McIlory (1976) suggested the concept of commercial software component production. After this, the methodology of component-based software development emerged encouraging the building of software systems by combining existing developed components. By using the component way of thinking you must also use a standardized way of developing. The components are units that have well-defined interfaces, with a specific behavior, following clear specifications (Washizaki, 2003). They are built with the purpose of reusability in mind, having to think everything from the perspective of families of systems and no longer as single systems. A component is developed to achieve a purpose, to cover a well-specified user need.

**Understanding Micro services architecture:** Microservices architecture refers to the structuring of an application into separate pieces that are loosely coupled. By loose coupling we understand the idea that every part has little to no knowledge about how other parts, that help build the whole system, are designed. In building microservices for a software application the programming language and the frameworks do not represent certain standards. Microservices are independently running which means they can use their own database, functions and language independently from the other parts that make up the application. To summarize, microservices are an architecture in which the process of software development is done by using autonomous components that isolate fine-grained business functionalities and communicate with each other through standardized interfaces (Hassan, 2017).

The idea behind microservices is to benefit from isolated units that are autonomous by coordinating them into distributed infrastructure by lightweight container technologies, such as Docker or Kubernetes (Rocha, 2018) Usually, when adopting this model of architecture, the developers also adopt an agile practice, such as DevOps, to be able to reduce the time between implementing a change in the system and transferring it to the production environment (Aderaldo, 2017).

## RESEARCH METHODS

The main purpose of this paper is to highlight the relation between component-based software development (CBS) architecture and microservices architecture based on various sources to present an overview of this dilemma. The main question that this research proposes to answer is: are microservices seen as components?

To be able to have a clear overview of the above-mentioned question, it was necessary to keep in mind some other questions that will help with our process:

- What are the differences between CBS and microservices?
- What are the similarities between CBS and microservices?
- How do people see these two concepts: different, complementary, swappable?

**Paper selection:** The search process involved an online search on three levels to provide a rich overview of how this topic is perceived through academic groups, company people and other IT professionals:

- Scientific papers: Google Scholar, Research Gate, Science Direct
- Online articles: Medium, private blogs, etc.
- Question and answer platforms for IT professionals: Stackoverflow.com, Softwareengineering.stackexchange.com etc.

The terms used for the gathering of the resources were: "Microservices vs. Components", "Microservices vs CBS", "Differences between CBS vs Microservices". Based on the first results it was observed that SOA vs Components and SOA vs Microservices was a better-researched topic and some of these results were included in this study by doing the extrapolation from SOA to Microservices by previously acquired knowledge.

**Process:** For filtering the information a three-step process was used. In the first step, 157 of scientific articles, 54 of informal articles and 32 of questions were reviewed. In the second step, the summaries were analyzed and the findings were narrowed to 57- 31- 10. In the final step, the selections were read and analyzed. The ones that were not relevant to the study were removed. In the end the number of sources was: 13 scientific articles, 10 online sources and 4 questions with 9 answers in total. For each study the idea behind it was extracted by keeping in mind the questions to be answered. Next, the data were aggregated into an Excel file, structured on levels (points of view of the scientific publications, opinions of the non-scientific articles, answers of the IT professionals on online platforms) and altogether to have a deeper understanding on sources point of view. This way, a quantification of different opinions: how many studies see the components and microservices as swappable, how many see them as different, or how many see them as complementary was obtained. The resulted numbers are further discussed in the next paragraph.

## RESULTS

The results of the research are presented at the source level and as a whole to paint a picture close to reality. This way we can have a better understanding of how different types of professionals are tackling the same subject. After quantifying the different opinions exposed in the selected scientific studies on their view of the concepts of components and microservices, Fig 2 was obtained. The obtained graphic (Fig.2) shows that more than 60% of the authors consider that components and microservices are a result of evolution through the idea that concepts can grow together. Only 8% of the articles perceived microservices as components with no important impact on the logic of the application when switching them from one to another. More than 30% of the articles portrayed clear differences between the two concepts. When looking through the internet for answers to questions regarding how microservices and components are seen the different opinions seem to be shared equally. The quantified answers are shown in Fig 3. So in the analysis of the relevant answers to questions on topics regarding the relation between microservices and components, a smaller number of perspectives (45%) than in the case of scientific articles considered that they are complementing each other, while 22% of the respondents considered that they are different architectures that do not intersect. For 33% of the responders the concept can be easily swappable as they bring the same benefits without having much trouble in interchanging them.



Figure 1 Google Trends analysis

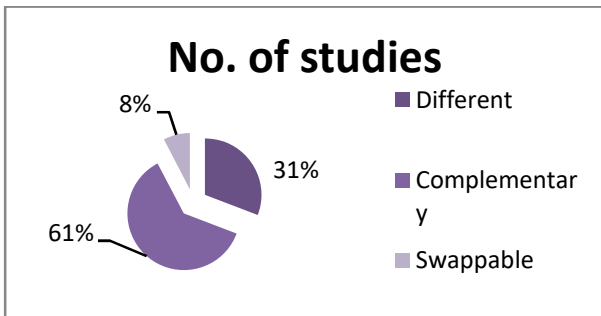


Figure 2 Perspectives from the scientific studies

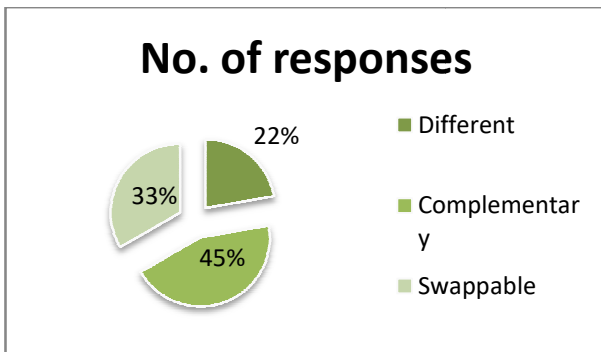


Figure 3 Perspectives from the answers to the questions

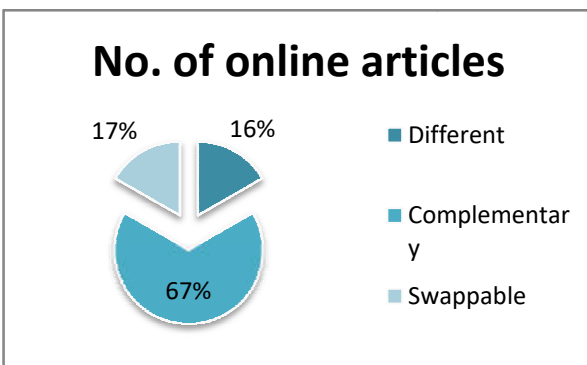


Figure 4 Perspectives from the online sources

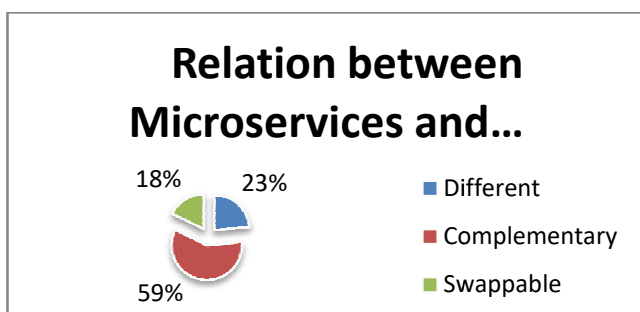


Figure 5 Overall perspectives

When analyzing online articles, that are published on different technical blogs by IT professionals, the opinions have a different spread than the other two sources reviewed before. The quantified points of view are shown in percentages in Fig 4. So, in the case of the online sources (Fig. 4), only 67% of the opinions consider that the two concepts complement each other, while 16% see them as different and 17% consider that they can be easily swapped. An overall perspective of the opinions no matter the source is shown in Fig 5. Further, the three perspectives of the relation between microservices and components are discussed in relation to the graphic.

**Complementary:** As we can see from Fig. 5, most of the analyzed points of view perceive microservices as an evolution from the component-based development. They consider that the development process can grow significantly by taking advantage of both architectures by using one on top of each other, combining them. A noticeable mention for exemplifying this perspective is the study published by R. M. Parizi "Microservices as an Evolutionary Architecture of Component-Based Development" (Parizi, 2019). The article focuses on the advantages of microservices that could benefit from a component-based approach. The observations highlight that the market is expecting from the microservices the part of software reusability and from the components the part of scalability. The proposal of the author is using microservices as components as a promising way and a good mechanism to redesign applications that were used to be developed using component-based architecture. In the article entitled "Towards an understanding of microservices" (Shadija, 2017) component-based development is seen as a point of start for evolving to microservices architecture. With the passing of the years, the newer concept proposes a higher level of granularity that is not forced to respect a certain level of standardization in the matter of programming language or behavior.

**Swappable:** As the numbers show, some perspectives see no impediments in switching components for microservices. For them there are no noticeable differences, advantages or challenges that could stop the replacement of one component with the other. In the first step of selection there were a lot of articles that did not have a clear distinction between the term "component" and "microservice", addressing statements like "Microservices are components ...", or using components as a synonym for microservices. These studies were not taken into consideration. One example in underlining the swappable aspect is the article of Bogner *et al.*, "Microservices in Industry: Insights into Technologies, Characteristics, and Software Quality" (Bogner, 2019). This study shows how microservices are used by various software companies. To have a real overview, the author interviewed a couple of IT& software companies. Through its cases there were three online shops where the component-based architecture was replaced with microservices architecture.

**Different:** Although at a lower rate there are still points of view that consider that microservice architecture is quite different from component-based software engineering. One example is the article of Bridgwater, "Microservices are not the same thing as components" (Bridgwater, 2016). In this study the author draws some fine lines of the distinction between microservices and components. He underlines that, even though there are studies that stipulate that a microservice should be structured to run independently from the platform, whereas the components are bound to the context of the software system structure they are used for, there are some

components that also run independently and are transferrable. The author's remarks regarding the peculiarity of these two are about the fact that the microservices can be written in different programming languages, with no restriction on the data storage, while the components are bound to the application structure (programming language, data storage). He concludes that microservices should be seen "not as a distinctly different "thing" in relation to components, but as an architectural style in its own right"(Bridgwater, 2019). In the same article there are some additional points of view. Dan Scholnick, an investor in Docker, Code Climate and New Relic and microservices enthusiast, expresses how he sees components as a general term for a single microservice within a microservices architecture, making a point in favor of the swappable aspect. On the other hand, Mark Davis, chief executive of ClusterHQ, argues that microservices are specifically styled-components and so comparing them is "like comparing an electric four-wheeled automobile with a mode of transportation" (Bridgwater, 2019), making a point in favor of complementary.

### Conclusion

With the rise of the impact of technology over human lives and the increase of complexity for software applications it becomes more clear the reason why people of today are interested in splitting the systems into smaller parts. This paper focuses its attention on two of these approaches: microservices architecture and the component-based architecture. The paper attempts to demystify the ambiguous usage of term components in a microservices interested world by analyzing different perspectives. The existing literature is the starting point. By investigating and reviewing different opinions in regards to the relation between components and microservices some conclusions are drawn. Throughout the study, the differences and similarities between the two concepts (which define the way they are seen) were the main concerns in conducting the study towards the main question: are microservices seen as components?. Different opinions from scientific articles, online sources and public answers to different questions were analyzed from a critical point of view and sorted into categories. The categories were in relation to how microservices and components are perceived as different, complementary or swappable. In order to lead to a quantitative overview the numbers were aggregated. All of the three perspectives showed up that when thinking about microservices and components we talk about a complementary relation no matter if the microservices are improving the components or the components are enhancing the microservices. Some opinions argued that components and microservices are "not the same thing" and that their differences cannot be overlooked. In smaller percentage components and microservices were seen as similar and easy to be swapped one with the other. As a general result, scientists and IT professionals embraced the fact that microservices architecture is an evolution from the components-based software development, but also the idea that either one of these concepts can still learn something from the other.

### Acknowledgments

This paper was co-financed from the Human Capital Operational Program 2014-2020, project number POCU / 380/6/13/125245 no. 36482 / 23.05.2019 "Excellence in interdisciplinary PhD and post-PhD research, career

alternatives through entrepreneurial initiative (EXCIA)", coordinator The Bucharest University of Economic Studies".

### REFERENCES

- Aderaldo, C. M., Mendonc, N. C., Pahl A. C. and Jamshidi, P. 2017. "Benchmark Requirements for Microservices Architecture Research.," in 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE).
- Bogner, J., Fritzsich, J., Wagner S. and Zimmermann, A. 2019. "Micro services in Industry: Insights into Technologies, Characteristics, and Software Quality," in IEEE International Conference on Software Architecture Workshops (ICSAW), Hamburg, Germany.
- Bridgwater, A. 2016. "Micro services are not the same thing as components," 6 January. (Online). Available: [https://www.theregister.co.uk/2016/01/06/inside\\_microservices/](https://www.theregister.co.uk/2016/01/06/inside_microservices/). (Accessed 19 May 2019).
- Hassan, S., Ali, N. and Bahsoon, R. 2017. "Microservice ambients: An architectural meta-modelling approach for microservice granularity," in IEEE International Conference on Software Architecture (ICSA), pages 1-10.
- McIlroy, M. D. 1976. "Mass-produced software components," in Software Engineering Concepts and Techniques (1968 NATO Conference on Software Engineering), New York, Van Nostrand Reinhold, pp 138-155.
- Parizi, R. M. 2018. "Microservices as an Evolutionary Architecture of Component-Based Development: A Think-aloud Study," May. (Online). Available: [https://www.researchgate.net/publication/325464032\\_Microservices\\_as\\_an\\_Evolutionary\\_Architecture\\_of\\_Component-Based\\_Development\\_A\\_Think-aloud\\_Study](https://www.researchgate.net/publication/325464032_Microservices_as_an_Evolutionary_Architecture_of_Component-Based_Development_A_Think-aloud_Study). (Accessed 19 May 2019).
- Petrtsch, H. 2019. "Service-Oriented Architecture (SOA) vs Component Based Architecture," (Online). Available: [http://www.petrtsch.co.at/download/SOA\\_vs\\_component\\_based.pdf](http://www.petrtsch.co.at/download/SOA_vs_component_based.pdf). (Accessed 19 May).
- Rocha, M. V. T. T., Figueiredo, E., Vigiato, M., Terra R. and Valente, M. T. 2018. "Microservices in Practice: A Survey Study," in Conference: VI Workshop on Software Visualization, Evolution and Maintenance, São Carlos - Brazil.
- Shadija, D., Rezai M. and Hill, R. 2017. "Towards an understanding of microservices," in 23rd International Conference on Automation and Computing (ICAC), University of Huddersfield, Huddersfield.
- Washizaki, H., Yamamoto H. and Fukazawa, Y. 2003. "A Metrics Suite for Measuring Reusability of Software Components," in Proceedings of the 9th International Symposium on Software Metrics, Sydney, Australia, September, pp 211-223.