DOI: 10.15514/ISPRAS-2024-36(1)-7

Research Trends and Recommendations for Future Microservices Research

[1] *Z. Stojanov*, ORCID: 0000-0001-6930-5337 <*zeljko.stojanov@uns.ac.rs*>
[2] *I. Hristoski*, ORCID: 0000-0002-9291-7873 <*ilija.hristoski@uklo.edu.mk*>
[1] *J. Stojanov*, ORCID: 0000-0002-2237-8026 <*jelena.stojanov@uns.ac.rs*>
[1] *A. Stojkov*, ORCID: 0000-0002-1078-7068 <*aleksandra.stojkov@tfzr.rs*>

[1] *University of Novi Sad, Technical Faculty "Mihajlo Pupin" Zrenjanin,*
BB, Djure Djakovica st., Zrenjanin, 23000, Serbia.
[2] *University "St. Kliment Ohridski" - Bitola, Faculty of Economics - Prilep,*
143, Prilepski Braniteli, Prilep, 7500, North Macedonia.

Abstract. Microservices are the most promising direction for developing heterogeneous distributed software systems capable of adapting to dynamic changes in business and technology. In addition to the development of new software systems, the migration from legacy monolithic systems to microservice architectures is also a prominent aspect of microservices use. These trends resulted in an increasing number of primary and secondary studies on microservices, stressing the need for systematization of research at a higher level. The objective of this study is to comprehensively analyze secondary studies in the field of microservices with objectives to inquire about publishing trends, research trends, domains of implementation, and future research directions. The study follows the guidelines for conducting a systematic literature review, which resulted in the findings derived from 44 secondary studies. The study findings are structured to address the proposed research objectives. Recommendations for further literature reviews relate to the improvement of quality assessment of selected studies to increase the validity of findings, a more detailed review of human and organizational factors through the microservices life cycle, the use of social science qualitative methods for more detailed analysis of selected studies, and inclusion of gray literature that will bring the real opinions and experiences of experts from industry.

Keywords: microservices; tertiary study; systematic literature review; research trends; recommendations.

For citation: Stojanov Z., Hristoski I., Stojanov J., Stojkov A. Research Trends and Recommendations for Future Microservices Research. *Trudy ISP RAN/Proc. ISP RAS*, vol. 36, issue 1, 2024. pp. 105-130. DOI: 10.15514/ISPRAS-2024-36(1)-7.

Full text: Stojanov Z., Hristoski I., Stojanov J., Stojkov A. A Tertiary Study on Microservices: Research Trends and Recommendations. *Programming and Computer Software*, 2023, Vol. 49, No. 8, pp. 796–821. DOI: 10.1134/S0361768823080200.

Направления будущих исследований и рекомендации по развитию микросервисной архитектуры

[1] *Ж. Стоянов*, ORCID: 0000-0001-6930-5337 <*zeljko.stojanov@uns.ac.rs*>
[2] *И. Христоский*, ORCID: 0000-0002-9291-7873 <*ilija.hristoski@uklo.edu.mk*>
[1] *Е. Стоянова*, ORCID: 0000-0002-2237-8026 <*jelena.stojanov@uns.ac.rs*>
[1] *А. Стойкова*, ORCID: 0000-0002-1078-7068 <*aleksandra.stojkov@tfzr.rs*>

[1] *Технический факультет "Михайло Пупин" университета Нови-Сада,
Сербия, 23000, Зренянин, ул. Джуре Джаковича, ББ.*
[2] *Университет Св. Климента Охридского - Битола, Экономический факультет - Прилеп,
Северная Македония, 7500, Прилеп, ул. Защитники Прилепа, 143.*

Аннотация. Микросервисы являются наиболее перспективным направлением для разработки разнородных распределенных программных систем, способных адаптироваться к динамическим изменениям бизнеса и технологий. В дополнение к разработке новых программных систем, переход от устаревших монолитных систем к микросервисным архитектурам также является важным аспектом использования микросервисов. Эти тенденции привели к увеличению числа первичных и вторичных исследований микросервисов, что подчеркивает необходимость систематизации исследований на более высоком уровне. Целью настоящего исследования является всесторонний анализ вторичных исследований в области микросервисов, который поможет выявить тенденции в направленности публикаций, исследований, уточнить области использования полученных результатов и перспективы будущих исследований. Представленное исследование следует рекомендациям по проведению систематического обзора литературы, в процессе его проведения были выявлены результаты 44 вторичных исследований. Эти результаты структурированы в соответствии с сформулированными авторами целями. Рекомендации для дальнейших обзоров литературы касаются улучшения оценки качества отдельных исследований для повышения достоверности результатов, повышения детализации обзоров человеческих и организационных факторов через жизненный цикл микросервисов, использования качественных методов социальных наук для более подробного анализа отдельных исследований, и включения в оборот литературы, обычно остающейся вне области внимания коммерческих и академических журналов, но содержащей реальные мнения и опыт промышленных экспертов.

Ключевые слова: микросервисы; третичное исследование; систематический обзор литературы; тенденции исследований; рекомендации.

Для цитирования: Стоянов Ж., Христоский И., Стоянова Е., Стойкова А. Направления будущих исследований и рекомендации по развитию микросервисной архитектуры. Труды ИСП РАН, том 36, вып. 1, 2024 г., стр. 105–130 (на английском языке). DOI: 10.15514/ISPRAS–2024–36(1)–7..

Полный текст: Стоянов Ж., Христоский И., Стоянова Е., Стойкова А. Третичное исследование микросервисов: направления исследований и рекомендации. Programming and Computer Software, 2023, т. 49, № 8, стр. 796–821 (на английском языке). DOI: 10.1134/S0361768823080200.

1. Introduction

Microservices have recently emerged as popular and widely used architectural model for cloud-based applications, representing a new trend in developing distributed software systems [1-2]. As small and independent services, they offer improved performance and support for continuous delivery [3]. Microservices based applications are in many cases built by breaking up monolithic applications, which assumes considering factors such as the number of objects owned by a service, the level of responsibility, and the team distribution [4]. In the beginning, microservices were adopted by large companies like Amazon, LinkedIn, and Netflix, and later by other companies [5-6], leading to an increasing trend in using microservices for developing cloud-based applications. The adoption of microservices in developing or reengineering software systems includes a new organizational and business culture in software organizations [7]. Adoption of DevOps in software companies is crucial for better integration of microservices-based applications throughout the system life cycle [8], leading to improvement their competitiveness [9].

Microservices-based applications consist of multiple components that collectively form the entire system. Each component performs a single task, with its boundaries shielding it from external knowledge, while the processed results can be shared and accessed by other microservices [2]. A system structure is stable even when upgrades or extensions are necessary. With microservices, clients can be confident that any changes or growth in their business will be implemented into software. Microservices show better performance than monolithic architectures, particularly in terms of meeting business requirements, ensuring systems reliability, enhancing maintainability, and bolstering infrastructure resilience [2]. Although microservices require a larger number of teams and greater effort, the long-term benefits make the investment worthwhile [10]. Migration of monolithic or legacy systems to service-oriented architectures is a common trend in contemporary software systems [11], particularly to microservice architectures, resulting with improved system performance [12-13]. Use of design patterns results in improvement of development practices and better fulfillment of various architectural quality attributes [14].

Microservices-based systems consist of individual microservices, each independently performing a specific functionality. Consequently, if one microservice fails, the entire system remains unaffected. The principle of *Autonomy* is responsible for this behavior, while other key principles are [10]: *Resilience* – ensuring that the application can continue providing services even if a specific microservice encounters failure; *Transparency* – exposing the necessary details and providing documentation for each microservice; *Automation* – employing tools that enhance the efficiency, reliability, and scalability of the microservices' building and maintenance processes; and *Alignment* – relating to harmonizing different microservices within the system.

Systematic Literature Reviews (SLRs) [15] and Systematic Mapping Studies (SMSs) [16] have recently been adopted by software engineering research community, for systematizing and analyzing the evidence on the practice and leading to Evidence-Based Software Engineering (EBSE) [17]. Review of academic literature, commonly referred to as "white literature", has recently been supplemented with "gray literature" sources such as blog posts, white papers, industrial magazines, and videos, introducing Multivocal Literature Reviews (MLRs) [18].

Based on the above discussion, the objectives of this study are: (1) to present the current publishing trends of secondary studies research, (2) to determine topics inquired in secondary studies, (3) to inquire in which domains are microservices commonly implemented, and (4) to present identified future research directions. A SLR based on the guidelines proposed in [15, 17] was performed, resulting in 44 secondary studies that were used for drawing research findings and recommendations for further research.

This paper is structured as follows. The second section presents related work on tertiary studies related to microservices. The third section outlines the research methods employed in the study, while the fourth section presents the research findings. Recommendations for future reviews are discussed in the fifth section. The last section contains conclusions.

2. Related work

Tertiary studies have been recently used in software engineering for reviewing secondary studies and conducting meta-analyses on specific research topics. Some of the tertiary studies relate to DevOps [19], architecting systems of systems [20], cloud computing [21], agile software development [22], variability in software product lines [23], or testing artifact quality [24].

Two tertiary studies on microservices were identified: "Research on Microservice Architecture: A Tertiary Study" by Liu et al. [25], and the second study titled "Microservice Architecture: A Tertiary Study" by Costa et al. [26]. Table 1 presents information on the period covered and the number of secondary studies included in identified tertiary studies. The review types encompass Systematic

Literature Reviews (SLRs), Systematic Mapping Studies (SMSs), Systematic Grey Literature Reviews (SGLRs), and Multivocal Literature Reviews (MLRs).

Table 1. Tertiary studies on microservices

Study reference	**Time span**	**SLRs**	**SMSs**	**MLRs**	**SGLRs**
Liu et al. (2022) [25]	2016-2021	17	20	0	0
Costa et al. (2020) [26]	2016-2019	5	14	2	1

Liu et al. [25] conducted a SLR and identified 37 secondary studies on microservices published in the period from 2016 to 2021. The authors formulated two research questions: (RQ1) What are the common topics addressed in secondary studies related to microservices architecture (MSA), and what are their findings? (RQ2) What are the potential areas for new research in the field of MSA? Quality of the secondary studies was assessed based on the DARE quality criteria [17].

Costa et al. [26] conducted a SLR and identified 22 secondary studies on microservices published in the period from 2016 to 2019. The original study was written in Portuguese, which required the translation of methodological issues and results into English. The authors addressed the following research questions: (RQ1) Which secondary studies have been published in the field of microservices? (RQ2) What research topics on microservices have been investigated? (RQ3) What emerging patterns have been identified? (RQ4) What solutions and support tools have been utilized to facilitate the development and operation of microservices architecture? (RQ5) In which areas, particularly in the industry, are microservices being applied? (RQ6) Which topics exhibit gaps and require further exploration in future microservices research? The quality of secondary studies was assessed by using the DARE quality criteria [17].

3. Research methods

This study is based on the guidelines for conducting SLRs proposed in [15, 17]. The research process contains the following main phases: (1) planning the review, (2) conducting the review, and (3) reporting the findings.

3.1 Planning the review

Justification of the need for a tertiary study, determining research questions, selecting digital databases, and defining the studies search and selection process with clearly stated keywords for searching, inclusion/exclusion criteria, and quality assessment criteria are described in this section.

3.1.1 Need to conduct a tertiary study on microservices

In the last decade, research on microservices has gained popularity by the researchers, resulting in an increasing number of empirical studies and leading to the execution of systematic reviews. These reviews were performed as SLRs [15], SMSs [16], and even MLRs [18].

During the search of literature on microservices, two tertiary studies on microservices were identified: a study titled “Research on Microservice Architecture: A Tertiary Study” by Liu et al. [25], and a study titled “Microservice Architecture: A Tertiary Study” by Costa et al. [26]. Insights into these two studies revealed a consistent increase in the number of secondary studies over the years, and it can be expected that this trend will continue in 2023 and beyond, highlighting the need for new reviews of recent secondary research.

3.1.2 Research questions

The following research questions (RQs) are proposed:

RQ1: What are the publishing trends observed in secondary studies?

RQ2: What are the predominant topics investigated in secondary studies?

RQ3: In which domains are microservices commonly implemented?

RQ4: What future research directions have been identified?

3.1.3 Search and selection process of secondary studies

Proposed research questions were used for selecting keywords for searching for secondary studies. Keywords were put into two groups: the first group includes the keywords "microservices" and "microservices architecture", while the second group comprises the keywords "SLR", "Systematic literature review", "SMS" and "Systematic mapping study". The following search strings were constructed for searching for secondary studies:

[1]: ("microservices architecture" OR "microservices") AND ("SLR" OR "Systematic literature review")

[2]: ("microservices architecture" OR "microservices") AND ("SMS" OR "Systematic mapping study")

The process of searching and selecting studies contains the following phases (Ph#No):

- Ph#1: Searching digital libraries using constructed search strings.
- Ph#2: Selecting specific studies based on their title, abstract, and keywords. This phase also involves removing duplicates (in case a study appears in multiple databases) and selecting the most recent version of the study (if there are multiple versions by the same authors).
- Ph#3: Using snowball search method [27] for finding additional studies and minimize the possibility of omitting relevant secondary studies.
- Ph#4: Applying inclusion/exclusion criteria to studies that passed phases Ph#2 and Ph#3.
- Ph#5: Conducting a detailed reading and analysis of the studies that passed Ph#4.

The digital libraries used for searching secondary studies are ACM Digital Library, IEEE Xplore, ScienceDirect, Springer, Wiley Online Library, and MDPI. These libraries were selected because they publish a majority of the leading journals and conference proceedings in the field.

Filtering of the studies identified during the search of digital libraries and snowball search was based on the inclusion and exclusion criteria. *Inclusion criteria* are: (I1) A study reviews relevant studies on microservices, (I2) A study follows guidelines for conducting SLR or SMS, (I3) A study answers research questions in the domain of microservices. *Exclusion criteria* are: (E1) Full text of a study is not available, (E2) A study is not peer-reviewed, (E3) The study is less than 6 pages, (E4) A study is not written in English, (E5) A review study that includes gray literature.

All selected secondary studies were evaluated against these inclusion and exclusion criteria, and if a secondary study failed to meet even one criterion, it was excluded from further analysis.

3.1.4 Quality assessment of secondary studies

The secondary studies were evaluated for quality based on guidelines proposed in [17]. The primary objective of the quality assessment was to identify and exclude low-quality studies from the detailed analysis and synthesis of review findings. Quality assessment was based on a three-point scale with values 1 (Yes), 0.5 (Partly), and 0 (No). This scale was based on the five questions (Q2-Q6) proposed in [17], while an additional question concerning the use of the review methodology (Q1) was added. The quality assessment questions are:

Q1: Is the review methodology clearly stated and appropriate?

Q2: Are the review's inclusion and exclusion criteria described and appropriate?

Q3: Is the literature search likely to have covered all relevant studies?

Q4: Did the reviewers assess the quality/validity of the included studies?

Q5: Were basic studies adequately described?

Q6: Were the extracted data from included studies synthesized in the findings?

Based on the authors' agreement, studies with an average quality score of less than 0.5 will be excluded from the detailed analysis.

3.2 Conducting the review

The search for secondary studies was performed in January 2023. The first phase resulted in the identification of 821 papers. The details of the search conducted in digital libraries are presented in Table 2.

Table 2. Total number of papers obtained through search in digital libraries

Library	**Number of search results for search string [1]**	**Number of search results for search string [2]**	**Totally for search strings [1] and [2]**
ACM Digital Library	68	90	158
IEEE Xplore	13	14	27
ScienceDirect	127	85	212
Springer	144	183	327
Wiley Online Library	31	58	89
MDPI	5	3	8
TOTAL	388	433	821

The phased process of selecting secondary studies is presented in Fig. 1, while the selection process of SLRs and SMSs throughout the phases is presented in Table 3.

Filtering in Ph#2 and Ph#3 resulted in the selection of 57 secondary studies. After implementing inclusion/exclusion criteria on each of the 57 studies, 44 were selected for further analysis and quality assessment. Three studies were excluded based on the E3 exclusion criterion, one study based on the E1 exclusion criterion, and nine studies based on the E5 exclusion criterion.

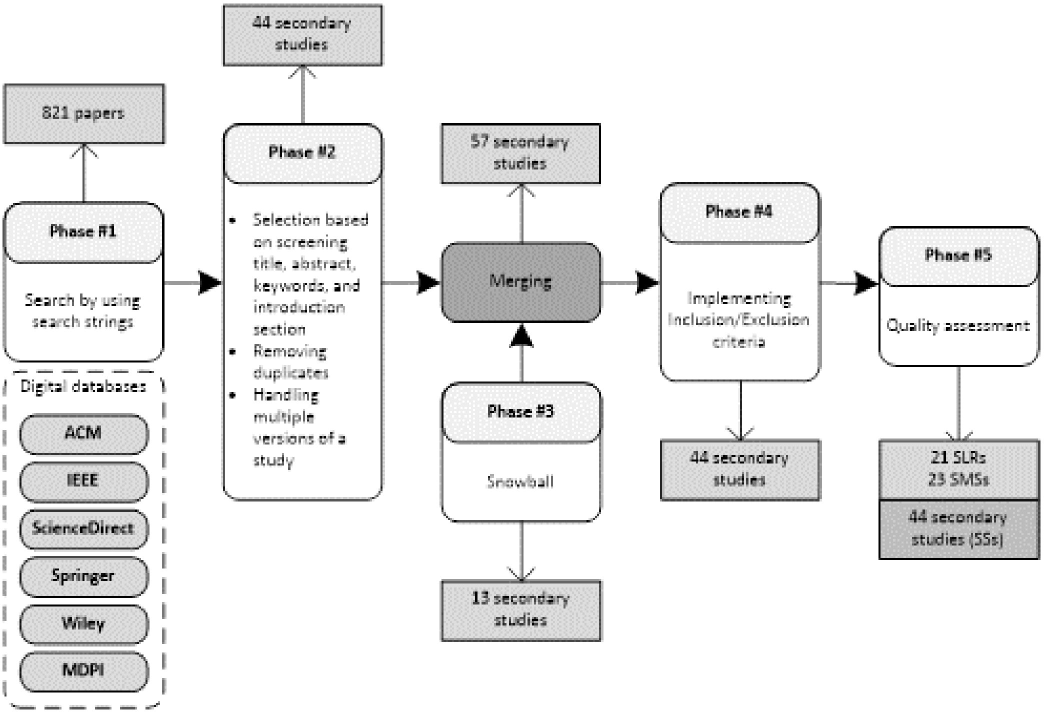

Fig. 1. The phased process of selecting secondary studies

Table 3. Details of the phased process for selecting secondary studies

	SLRs	SMSs	MVRs	Totally SSs
Ph#1: Selected studies after checking titles, abstracts, and keywords	19	23	14	56
Ph#2: Selected studies after removing duplicates	16	19	9	44
Ph#3: Snowball search for additional studies	6	7	0	13
Merging digital libraries and snowball search results	22	26	9	57
Ph#4: Selected studies after implementing Inclusion/Exclusion criteria	21	23	0	44
Ph#5: A final set of secondary studies after quality assessment	21	23	0	44

3.2.1 Quality assessment

The first, third, and fourth authors assessed the quality of secondary studies, while the second author reviewed the grades and calculated the average quality scores for all studies. Each evaluator assigned a mark from the three-point scale (0.0, 0.5, or 1.0) to each study for all quality assessment criteria. The average quality score for each study was then calculated as the mean of all average values assigned by each evaluator. Table 4 presents the average scores for the quality assessment of the selected studies. It is evident that all studies exceeded the minimum required quality threshold for further analysis (overall quality score greater than 0.50).

Table 4. Average scores for quality assessment of the selected secondary studies

ID	Year	Type	Q1	Q2	Q3	Q4	Q5	Q6	AVG
SS01	2020	SLR	0.67	0.67	0.83	0.33	0.83	0.83	0.69
SS02	2019	SLR	1.00	1.00	1.00	0.83	1.00	1.00	0.97
SS03	2019	SLR	0.83	1.00	1.00	0.83	0.83	0.83	0.89
SS04	2020	SLR	0.67	1.00	1.00	0.33	0.83	0.67	0.75
SS05	2018	SLR	0.50	0.50	0.67	0.33	0.83	0.83	0.61
SS06	2021	SLR	1.00	1.00	1.00	1.00	0.83	1.00	0.97
SS07	2021	SLR	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SS08	2018	SLR	0.50	0.67	0.67	0.50	1.00	1.00	0.72
SS09	2019	SLR	0.50	0.50	0.83	0.67	0.83	0.83	0.69
SS10	2020	SLR	1.00	1.00	1.00	0.33	1.00	1.00	0.89
SS11	2022	SLR	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SS12	2022	SLR	0.50	0.33	1.00	0.17	0.83	0.83	0.61
SS13	2022	SLR	1.00	1.00	0.83	0.00	0.83	0.83	0.75
SS14	2022	SLR	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SS15	2022	SLR	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SS16	2022	SLR	0.83	1.00	1.00	0.83	1.00	1.00	0.94
SS17	2021	SLR	1.00	1.00	0.83	0.83	1.00	1.00	0.94
SS18	2021	SLR	0.67	1.00	1.00	0.50	1.00	1.00	0.86
SS19	2018	SLR	1.00	1.00	0.67	0.83	1.00	0.83	0.89
SS20	2021	SLR	0.67	1.00	0.83	0.33	1.00	0.83	0.78
SS21	2021	SLR	0.67	1.00	1.00	0.00	0.83	0.83	0.72
SS22	2022	SMS	1.00	1.00	0.83	0.00	1.00	0.83	0.78
SS23	2017	SMS	1.00	0.83	1.00	0.00	0.67	0.83	0.72
SS24	2021	SMS	0.83	1.00	1.00	0.00	1.00	1.00	0.81
SS25	2020	SMS	1.00	1.00	1.00	0.00	0.67	1.00	0.78
SS26	2021	SMS	1.00	1.00	0.83	0.00	0.67	0.83	0.72
SS27	2017	SMS	1.00	1.00	1.00	0.00	0.67	1.00	0.78
SS28	2019	SMS	1.00	1.00	1.00	0.67	1.00	0.83	0.92
SS29	2016	SMS	1.00	1.00	0.83	0.17	1.00	1.00	0.83
SS30	2022	SMS	1.00	1.00	1.00	0.00	1.00	0.83	0.81
SS31	2022	SMS	0.83	0.83	1.00	0.75	0.50	0.83	0.79
SS32	2021	SMS	0.83	1.00	1.00	0.00	1.00	0.67	0.75
SS33	2020	SMS	1.00	0.83	1.00	0.33	1.00	1.00	0.86
SS34	2019	SMS	1.00	1.00	1.00	0.00	1.00	1.00	0.83

SS35	2021	SMS	1.00	1.00	1.00	1.00	1.00	0.83	0.97
SS36	2017	SMS	0.50	0.50	0.83	0.00	0.67	0.83	0.56
SS37	2019	SMS	1.00	1.00	1.00	0.00	0.83	0.83	0.78
SS38	2022	SMS	1.00	0.83	1.00	0.00	1.00	0.67	0.75
SS39	2021	SMS	0.83	1.00	1.00	0.00	1.00	1.00	0.81
SS40	2019	SMS	0.83	1.00	1.00	0.00	1.00	1.00	0.81
SS41	2019	SMS	0.83	1.00	1.00	0.00	0.17	1.00	0.67
SS42	2019	SMS	0.50	0.33	0.83	0.33	0.50	0.67	0.53
SS43	2016	SMS	1.00	1.00	1.00	0.17	1.00	1.00	0.86
SS44	2023	SMS	1.00	1.00	1.00	0.00	0.33	1.00	0.72
Average scores for all SSs			0.86	0.91	0.94	0.37	0.87	0.90	0.81
Average scores SLRs			0.81	0.89	0.91	0.60	0.93	0.91	0.84
Average scores SMSs			0.91	0.92	0.96	0.15	0.81	0.89	0.77

Based on the quality assessment results, the minimum score achieved was 0.53 (for study SS42), while the average quality score across all studies was 0.81. All 44 secondary studies passed quality analysis and were selected for in-depth analysis. Selected secondary studies (SSs) are listed in Appendix A.

3.2.2 Data extraction

The template presented in Table 5 is used for extracting data on secondary studies, encompassing general information about each study's publication, data relevant for quality assessment, and specific data relevant to each research question. The extracted data was organized in an Excel spreadsheet.

Table 5. Data extraction template

ID	**Explanation**	**Use**
D1	Study ID	Demography, RQ1
D2	Title	Demography, RQ1
D3	Year	Demography, RQ1
D4	Study type (SLR, SMS)	Demography, RQ1
D5	Venue type (conference, journal, book chapter)	Demography, RQ1
D6	Sample size (number of primary studies)	Demography, RQ1
D7	Used research methodology description	Quality assessment Q1
D8	Inclusion/Exclusion criteria	Quality assessment Q2
D9	Coverage of relevant studies	Quality assessment Q3
D10	Quality assessment questions	Quality assessment Q4
D11	Method for describing selected studies	Quality assessment Q5
D12	Data extraction methods and tools	Quality assessment Q6
D13	Research questions	Research topics, RQ2
D14	Research topics	Research topics, RQ2
D15	Technical implementation areas	Application area, RQ3
D16	Future research directions	Future research direction, RQ4

4. Research findings

The findings were derived from the data extracted from 44 selected secondary studies, 23 of them are SMSs, while the remaining 21 are SLRs (refer to Table 3). The findings will be organized and presented in alignment with the research questions.

4.1 Publishing trends for secondary studies (RQ1)

The selected secondary studies span the publication period from 2016 to 2023. Fig. 2 presents the publication trends of secondary studies based on the publishing venue.

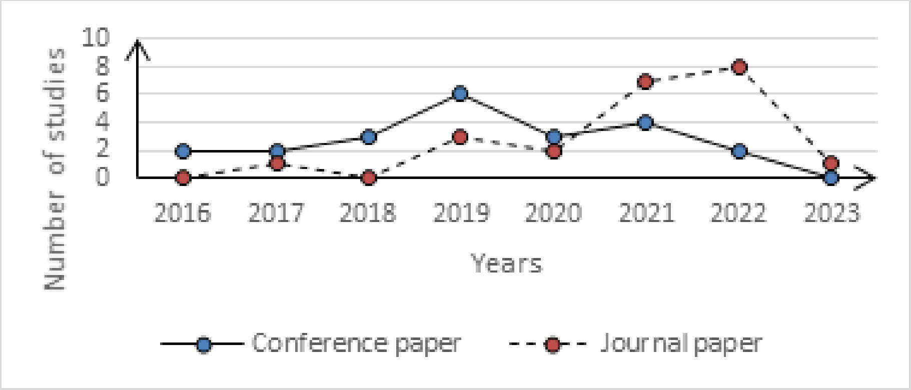

Fig. 2. Secondary studies publication trends by venue

It is obvious that the total number of studies has consistently grown over the years, and there has been a notable shift from primarily conference proceedings to journals. Out of the 44 studies, 22 were published in conference proceedings, and another 22 were published in journals.

4.2 Topics inquired in secondary studies (RQ2)

The identification of topic areas in the selected secondary studies is based on a comprehensive examination of each study. The extracted data corresponds to values in columns D13 (research questions) and D14 (research topics) in Table 5. A general overview of the topics investigated in the secondary studies is depicted in Fig. 3. It is worth noting that while most studies have a primary focus on a specific topic, they also touch upon other related topics.

4.2.1 Architecture

Architectural design is essential for the development of microservice-based software systems because it encompasses both the technical design of system functionalities and non-functional requirements, which are often referred to as quality attributes. Well-designed architecture is important for efficient development, operation, and maintenance of software systems. The main topics related to microservices architecture in the selected secondary studies include Analysis, Granularity, Patterns, Presentation, and Quality attributes (refer to Fig. 3). A more comprehensive examination of architecture topics, including identified subtopics and the distribution of secondary studies that mention them, is provided in Table 6.

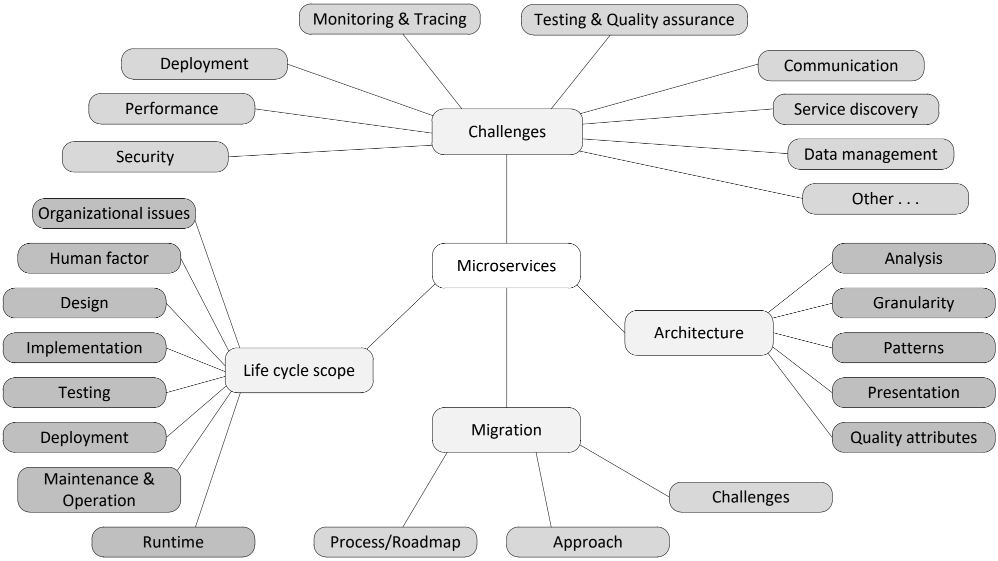

Fig. 3. General overview of microservices research topics

Analysis. The analysis of software architecture is crucial for understanding both new software systems and those that require migration to microservice architectures. It focuses on the following aspects:

(1) *Methods* - various methods can be distinguished, including Static Analysis, Dynamic Analysis, Combined Dynamic and Static Analysis, Model-Based Analysis, Graph-Based Analysis, and Pattern-Based Analysis;

(2) *Tools* - specific tools tailored to each analysis method, facilitating automated work; and

(3) *Challenges* – these encompass architectural analysis, software architecture reconstruction, technical debt analysis, quality attribute analysis, and fault analysis.

Granularity. Granularity refers to the size of individual microservices within a software system based on the microservices architecture. It plays a crucial role in determining the functioning of the system and its quality attributes, including performance, maintainability, data storage, and scalability. Determining the optimal granularity involves finding the right balance between the level of functionality encapsulated within each microservice and the need for modularity, maintainability, and scalability. Insights from the selected studies on granularity reveal the following main subtopics:

(1) *Methods or approaches* for defining granularity;

(2) *Metrics* used for evaluating granularity; and

(3) *Quality attributes* affected by granularity.

Table 6. Topics and subtopics related to microservices architecture

Topic	**Subtopic**	**Secondary studies**
Analysis	Methods	SS09, SS27, SS35, SS39, SS44
	Tools	SS30, SS39
	Challenges	SS30, SS39
Granularity	Approach	SS04, SS09, SS17, SS24
	Quality attributes	SS17
	Metrics	SS17
Patterns	Design	SS03, SS05, SS19, SS27, SS33, SS34, SS37, SS43

	Composition	SS02, SS09, SS19, SS33, SS43
	Communication	SS03, SS05, SS06, SS19, SS33, SS37
	Deployment	SS03, SS06, SS37, SS19, SS33, SS37
	Data storage	SS03, SS19, SS33, SS37
	Antipattern	SS01, SS44
Presentation	Languages	SS22, SS27, SS29, SS33
	Diagrams	SS27, SS29
	Visualization	SS30
Quality attributes	Reliability	SS03, SS08, SS17, SS27, SS33, SS34
	Security	SS03, SS07, SS08, SS17, SS27, SS29, SS33, SS40
	Compatibility	SS03, SS27, SS33, SS34, SS40
	Maintainability	SS03, SS08, SS17, SS27, SS29, SS33, SS34
	Performance	SS03, SS07, SS17, SS19, SS27, SS29, SS33, SS34, SS40
	Portability	SS03, SS27, SS33, SS34
	Testability	SS07, SS33
	Availability	SS07, SS08, SS17, SS33
	Monitorability	SS07, SS33
	Scalability	SS07, SS08, SS17, SS19, SS27, SS29, SS33, SS34, SS40
	Modularity	SS17, SS29
	Other …	SS29, SS33, SS40

Patterns. The identification and categorization of specific challenges and their corresponding solutions during software development, operation, and maintenance contribute to the recognition of recurring scenarios, commonly referred to as patterns. Incorporating patterns into the software life cycle enables developers to find reliable solutions to common problems, enhances communication among team members and with clients, and aids in meeting quality requirements. The following categories of patterns are identified:

(1) *Design* – patterns used for structuring and organizing microservices efficiently (API gateway, publish/subscribe, circuit breaker, proxy, and load balancer)

(2) *Composition* – patterns related to composing different microservices in a software system (semantic annotation, best-fitting, and workload-based approaches);

(3) *Communication* – patterns related to communication between multiple microservices (synchronous communication, publish/subscribe communication, combination of HTTP and message queue, communication using message-oriented middleware, asynchronous communication, point-to-point communication, and communication using binary protocols);

(4) *Deployment* – patterns related to the deployment or distribution of microservices to multiple resources for operational use (serverless deployment, service instances per VM, and service instances per container); and (5) *Data storage* – patterns oriented towards improving performances of data management systems (database-per-service pattern, the database cluster pattern, and the shared database server pattern). In addition, antipatterns refer to design and implementation choices that result in inadequate/poor software system design, leading to issues during operation and maintenance.

Presentation. Efficient design and utilization of microservice-based systems require the presentation of software architecture through various approaches that depict the system's structure and behavior. The secondary studies identified the following subtopics:

(1) *Languages* – specific languages for describing the architecture, such as RAML, YAML, Jolie, or various pseudocodes;

(2) *Diagrams* – visual diagrams, ranging from informal drawings to specialized and sophisticated diagrams such as Component/Container, Process/Behavior, Sequence, Execution Timeline, Deployment, Class, Use Case, Type Graph, Instance Graph, and Dependency Graph; and

(3) *Visualization* – dynamic analysis techniques, supported with specialized tools, for analyzing or recovering software architecture.

Quality attributes. Quality characteristics of software systems are typically reflected in the form of quality attributes, which are associated with non-functional requirements. While there are numerous quality attributes, it is often challenging to satisfy all of them simultaneously. In such cases, trade-offs must be made when considering quality attributes. The most common quality attributes mentioned in many studies are Reliability, Security, Compatibility, Maintainability, Performance, Portability, Testability, Availability, Monitorability, Scalability, and Modularity. These attributes have been extensively discussed in various sources and are relevant throughout the software life cycle, from design to operational use and maintenance. The selected secondary studies also mention additional quality attributes, such as Modifiability, Usability, Deployability, Flexibility, Reusability, Manageability, Independence, Traceability, Complexity, Load balancing, and Organizational alignment. To effectively address quality attributes, appropriate metrics need to be proposed, such as time, complexity, number of requests, or number of affected files. These metrics facilitate continuous improvements in quality attributes over time.

4.2.2 Life cycle scope

Microservice-based systems undergo various lifecycle phases that bring forth unique challenges, necessitating the utilization of specific methods, approaches, and tools. The primary topics identified in the selected secondary studies pertaining to the life cycle scope of microservices encompass Design, Implementation, Testing, Deployment, Maintenance and Operation, Runtime, Organizational issues, and the Human factor (refer to Fig. 3). A review of life cycle scope topics, subtopics and the distribution of secondary studies mentioning them is given in Table 7.

Table 7. Topics and subtopics related to the life cycle scope of microservices

Topic	**Subtopic**	**Secondary studies**
Design	Identification strategies	SS02, SS04
	Domain Driven Design	SS04, SS39
	Representations	SS04, SS34, SS36, SS39
	Design for failure	SS08, SS36
Implementation	Technology stack	SS08, SS36, SS37
	Supporting systems	SS08, SS34, SS36, SS37
	Services interfaces	SS08, SS34
Testing	Approaches	SS14, SS23, SS25, SS36, SS37, SS40
	Tools	SS25, SS37, SS40
Deployment	Platform	SS08, SS32, SS34, SS37
	Monitoring	SS08, SS14, SS23, SS32, SS34, SS36, SS39
	Approaches	SS06, SS23, SS32, SS37

Maintenance & Operation	Load balancing	SS11, SS34
	Fault diagnosis	SS11, SS23, SS36, SS37, SS39
	Autoscaling	SS11, SS12, SS21, SS34, SS37
	Anomaly detection	SS13, SS37
	Resource Scheduling	SS13, SS33
	Analysis	SS36, SS39
Runtime	Virtualization	SS08, SS43
	Discovery	SS14, SS43
	Control	SS08, SS34, SS39
	Verification and Validation	SS08, SS21
	Visualization	SS30, SS39, SS44
Organizational issue	DevOps	SS08, SS19, SS25, SS32, SS33, SS34, SS37, SS43
	Continuous processes	SS08, SS25, SS43
Human factor	Roles	SS04, SS17
	Skills	SS04, SS38

Design. The design phase of the lifecycle is crucial for achieving the desired system structure and fulfilling the proposed quality characteristics. The following subtopics are identified in the secondary studies:

(1) *Identification strategies* –focuses on the identification of services during the design of complex systems;

(2) *Domain-Driven Design* – relates to the use of principles, patterns, and domain-specific knowledge during system design;

(3) *Representations* –the use of various methods and tools for representing microservice system being developed; and

(4) *Design for failure* – relates to design principles and methods that enable the design of systems with increased fault tolerance, self-healing capabilities, and variability characteristics.

Implementation. The implementation phase utilizes the products and decisions from the design phase to create microservices and integrate them into a system. The following subtopics are identified:

(1) *Technology stack* –the use of various languages (formal, scripting, object-oriented), interaction models for communication flow, and protocols for data exchange (e.g., REST/HTTP, RPC-alike, message queues);

(2) *Supporting systems* –focuses on data storage systems for distributed microservices (e.g., SQL, graph-oriented, document-oriented) and systems for service discovery in a dynamic environment; and

(3) *Service interfaces* – concerns the specification of contracts for microservices communication.

Testing. The complex nature and dynamic behavior of microservice-based systems present several challenges in their testing. The following subtopics are identified:

(1) *Approaches* –encompasses various testing approaches employed during development, ranging from unit testing to integration testing (continuous testing as part of DevOps and continuous engineering practices, testing of microservices and system performance, testing during migration, and model-based testing); and

(2) *Tools* – relates to tools utilized in the testing process, with a preference for automated testing. It includes libraries and frameworks that enable specific types of tests.

Deployment. A deployment practice encompasses activities, methods, and tools necessary for the establishment of heterogeneous microservices to meet the requirements of contemporary businesses. Automated and continuous development and deployment processes are essential in ensuring the reliable and scalable delivery and operation of microservice-based systems. The following subtopics are identified:

(1) *Platform* – pertains to the selection of a hosting system for running microservices;

(2) *Monitoring* – relates to the activities performed to prevent or respond to failures or changes in the environment; and

(3) *Approaches* – addresses the various ways and strategies for facilitating the utilization of microservices-based systems.

Maintenance & Operation. The primary focus of maintenance and operation activities in the software life cycle is to ensure the usability and operability of the software. The following subtopics are identified:

(1) *Load balancing* – pertains to the coordination and management of a large number of service requests in systems with heterogeneous and distributed microservices;

(2) *Fault diagnosis* – involves improving the quality and efficiency of software operation by detecting faults (monitoring and localization of faults, identifying fault types, and fault modeling);

(3) *Autoscaling* – relates to the adjustment of system resources to meet changing needs and growing requirements (resource allocation, prediction and scheduling methods);

(4) *Anomaly detection* – focuses on identifying critical behaviors or abnormal states in application performance;

(5) *Resource scheduling* – involves the dynamic adjustments of system resources in response to the overall system state and workload.; and (6) *Analysis* – covers methods and tools for analyzing the states and behaviors of microservice-based systems during operational use.

Runtime. Analyzing the architecture, functioning, and performance of microservices-based systems requires extracting information from both static and dynamic sources during runtime, which is crucial due to the changes in structure and communication. The following subtopics are identified:

(1) *Virtualization* – pertains to different levels of platform abstraction, isolation, and sharing;

(2) *Discovery* – relates to identifying and finding appropriate services based on workload, scalability, and service quality considerations;

(3) *Control* – involves managing execution at both the local level of individual microservices and the system level as a whole;

(4) *Verification and Validation* – focuses on assessing the quality of microservices during runtime; and

(5) *Visualization* – relates to visually representing the microservices architecture during runtime, which covers techniques, tools, and types of information to be presented.

Organizational issue. Software development, operation, and maintenance take place within specific organizational contexts, including software development organizations and client organizations. These contexts have their own processes, procedures, challenges, and cultures that impact the technical and technological aspects of software processes. The following subtopics are identified:

(1) *DevOps* –encompasses the cultural and practical aspects of organizing stakeholders involved in the *development* and operation; and

(2) *Continuous Delivery, Integration, and Deployment* – focuses on continuous activities that facilitate a seamless and smooth transition between life cycle phases in microservices-based systems.

Human factor. Even though the literature primarily emphasizes technical and technological aspects, it is important to recognize that all activities in the software systems' life cycle are carried out and supervised by people. The following subtopics are identified:

(1) *Roles* – pertains to the various roles that individuals assume in the microservice life cycle; and

(2) *Skills* – focuses on the technical and soft skills that are necessary for individuals in different roles. It encompasses the specific knowledge, expertise, and abilities required to effectively perform their tasks.

4.2.3 Migration

One of the primary challenges with monolithic legacy software systems is the need for subsequent modifications to keep them operational and useful for end users. Frequent modifications can increase software complexity, reduce performance, and make maintenance challenging. A common solution is migrating software systems to microservices architectures. The main topics related to migration to microservices are Approach, Process/Roadmap, and Challenges (refer to Fig. 3). A more detailed review of migration topics, including identified subtopics and the distribution of secondary studies that mention them, is presented in Table 8.

Approach. In practice, different types of legacy systems require varying approaches for modernization and migration to microservice architectures. The identified subtopics are:

(1) *Strategy* – pertains to the overall strategy chosen in a migration project, such as clustering, candidate identification based on quality attributes, data-driven approaches, or bottom-up approaches;

(2) *Decomposition method* – focuses on the selection of the analysis method used to decompose the legacy system and identify microservices;

(3) *Unit level* – involves selecting the most suitable level of software artifacts during the decomposition of the old system and migration to microservices (business functions, database tables, classes, use cases); and

(4) *Evolution* – relates to supporting the scalability and maintainability features throughout the migration process.

Table 8. Topics and subtopics related to migration to microservices

Topic	Subtopic	Secondary studies
Approach	Strategy	SS24, SS13, SS26, SS38
	Decomposition method	SS09, SS24, SS13, SS26
	Unit level	SS09, SS24, SS38
	Evolution	SS09, SS24, SS38
Process/Roadmap	Definition	SS24, SS26, SS38
	Input information	SS24, SS26, SS38
	Output information	SS24, SS26
	Success factors	SS38
	Motivation	SS24
Challenges	Technical	SS24, SS26, SS38
	Organizational	SS26, SS38
	Knowledge and skills	SS04, SS26, SS38

Process/Roadmap. Every migration project follows a process or roadmap that guides the organization of activities and determines the input and output information. The identified subtopics are:

(1) *Definition* – pertains to the selection or proposal of different processes, guidelines, and roadmaps, supported by specific tools;

(2) *Input information* – focuses on identifying the required input information for the migration process;

(3) *Output information* – relates to the information produced upon completing the migration process (e.g., microservices candidates, communication approaches);

(4) *Success factors* – encompasses the factors that influence the successful execution and completion of the migration process; and

(5) *Motivation* – explores the motivations or driving factors (technical, operational, or organizational) behind organizing a migration project.

Challenges. The migration of existing systems to a new microservice-based architecture is a challenging project that poses various obstacles for organizations, teams, and individuals. The identified subtopics are:

(1) *Technical* – focuses on the introduction of new technologies and the selection of the most suitable tools for the migration process;

(2) *Organizational* – pertains to organizational changes within an IT company that undertakes a migration process; and

(3) *Knowledge and skills* – emphasizes the importance of selecting team members with the appropriate knowledge and a combination of technical and non-technical skills necessary for the successful implementation of the migration project.

4.2.4 Challenges

The adoption of microservices in industrial practice presents numerous challenges for both practitioners and researchers, given the inherent complexity and heterogeneity of microservice-based systems. Fig. 3 illustrates eight challenges that have been mentioned in at least three secondary studies, with additional challenges categorized under the shape “Other ...”. A comprehensive list of 21 identified challenges, along with the corresponding secondary studies that reference them, is presented in Table 9. By understanding and tackling these challenges, practitioners, and researchers can make significant strides in overcoming the obstacles inherent in microservice adoption.

Regardless of the topics covered, all studies identified specific challenges and proposed corresponding solutions. Among the challenges identified, Security emerges as the most crucial, with 13 occurrences across the studies. These discussions encompass a wide range of topics, needs, and scenarios, with proposed taxonomies or frameworks to address security issues. Following closely, Communication is mentioned in seven studies as the second most frequently cited challenge. Communication challenges may arise from remote calls, during replication of services or data, or service discovery. The third most frequently mentioned challenge pertains to Testing and quality assurance of microservice-based systems. This encompasses various specific challenges, including faster test feedback, automated testing, intercommunication testing, granularity testing, runtime testing, integration testing, and performance testing.

Table 9. Challenges related to microservices

Challenge	Secondary studies	Frequency
Security	SS14, SS15, SS16, S18, SS23, SS27, SS28, SS29, SS33, SS34, SS35, SS36, SS42	13
Communication	SS06, SS14, SS18, SS23, SS27, SS29, SS34	7

Testing & Quality assurance	SS14, SS23, SS25, SS27, SS34, SS36	6
Performance	SS14, SS23, SS29, SS33, SS36	5
Deployment	SS06, SS23, SS29, SS32, SS36	5
Monitoring & Tracing	SS14, SS23, SS29, SS32, SS36	5
Service discovery	SS14, SS23, SS29	3
Data management	SS14, SS27, SS34	3
Scalability	SS21, SS36	2
Migration	SS26, SS38	2
Complexity	SS27, SS34	2
Composition	SS27, SS34	2
Decomposition	SS14	1
Orchestration	SS14	1
Modeling	SS23	1
Context awareness	SS23	1
Integration	SS29	1
Fault tolerance	SS29	1
Publishing	SS32	1
Upgrading	SS32	1
Availability	SS36	1

4.3 Technical implementation and integration (RQ3)

The dynamics and the increasing need for integrating heterogeneous and complex systems in various sectors such as healthcare, industry, and transportation necessitate the adoption of service-oriented architectures, especially microservices-based architectures, to facilitate distributed processing capabilities and data integration. The identified topics are Service type, Service domain, and Industry adoption (refer to Fig. 3). Table 10 provides a detailed overview of the identified related subtopics and the distribution of secondary studies that mention them.

Table 10. Topics and subtopics related to technical implementation and integration of microservices.

Topic	**Subtopic**	**Secondary studies**
Service type	Functional	SS05, SS34
	Infrastructure	SS05, SS34
Domain	Smart systems	SS05, SS09, SS31, SS43
	Fog applications	SS42
	Big Data	SS20
	Blockchain	SS31
	Enterprise	SS31, SS33, SS43
Industry adoption	Readiness level	SS27, SS34
	Industry involvement	SS27, SS34
	Tools and system support	SS27, SS34
	Evaluation & Benchmarking	SS27, SS31, SS34

Service type. Microservices are deployed within complex and heterogeneous systems, requiring sophisticated infrastructures for hosting and execution. Apart from the functional services that cater to user needs, it is crucial to deploy services that support the infrastructure essential for the proper functioning of microservices. Subtopics in this context are:

(1) *Functional services* – are responsible for providing functionalities to users, enabling them to perform tasks or access features; and

(2) *Infrastructure services* – non-functional requirements, infrastructure, and service monitoring, as well as maintenance tasks that are not directly related to user functionalities.

Domain. The secondary studies have identified the following subtopics:

(1) *Smart systems* – focuses on the implementation of microservices for technical integration in smart systems, such as smart cities, smart transportation, and IoT applications;

(2) *Fog applications* – explores the use of microservices in fog computing applications;

(3) *Big Data* – examines the utilization of microservices in Big Data applications;

(4) *Blockchain* – investigates the combination of microservices with blockchain technologies; and

(5) *Enterprise* – focuses on the use of microservices in various business domains, such as healthcare, online commerce, supply chain management, financial systems, and telecommunications.

Industry adoption. The majority of studies report the widespread acceptance and adoption of microservices in various business scenarios, establishing them as a prominent software development approach in the software and IT industry. The identified subtopics are:

(1) *Readines*s l*evel* – pertains to the maturity level of specific methods, tools, and technologies intended for implementation in industrial projects;

(2) *Industry involvement* – explores the degree of engagement and participation of industry experts in research projects;

(3) *Tools and system support* – examines the utilization of specialized tools and systems that support the development and operation of microservice-based systems; and

(4) *Evaluation & Benchmarking* – emphasizes the use of evaluation and benchmarking tools to assess the quality and effectiveness of designed microservice-based architectures.

4.4 Directions for future research (RQ4)

Potential future research directions have been identified by analyzing the discussion sections and concluding remarks of each examined secondary study. Therefore, we searched within the secondary studies using keywords such as ‘future’ in combination with the words ‘research’, ‘work’, and ‘direction’, as well as the keywords ‘further’, ‘challenge’, and ‘gap’.

Notably, the majority of authors indicate future research directions in two main ways:

(a) by highlighting and elaborating on potential research gaps, open challenges, and issues encountered during their analysis, and using them as a basis to propose broader research directions; and

(b) by proposing specific research directions that are closely related to the research methodology employed in their study.

In the coming years, the research focus will center on addressing the following issues:

- Exploring various types of microservice architectures, their structure, and design aspects. This involves expanding the existing knowledge base in a systematic, structured, and consistent manner, (a) by including both theoretical and practical learning and exploration, (b) by investigating the application of microservices in specific domains or real-world scenarios, and (c) by considering organizational factors and addressing human-related issues in microservice architectures.

- To ensure the quality of microservice architectures, it is crucial to assess the associated quality attributes, while considering their complex interactions and trade-offs. By evaluating and addressing these attributes, researchers and practitioners can optimize the performance, robustness, and overall effectiveness of microservice architectures. This comprehensive assessment can help in developing resilient and adaptable systems.
- Focusing on the industrial adoption of microservices architecture, which involves several crucial phases, including design, implementation, validation, operation, deployment, maintenance, and testing of microservice architectural designs in practice. By emphasizing these essential phases, organizations can effectively adopt microservice architectures and reap the benefits of scalability, flexibility, and maintainability offered by this architectural style.
- Standardizing microservice architecture, interfaces, and related aspects such as load balancing, fault detection, and autoscaling.
- Designing fault-tolerant and event-driven/asynchronous microservices, particularly for smart systems, fog applications, and IoT applications.
- Transitioning from specific solutions and their related validation to more general solutions through fundamental research, reusable practices, and lessons learned.
- Addressing the complexities of the migration process from monolithic applications to microservice-based architectures systematically, as well as tackling challenges related to microservice identification, granulation, and proper design.
- Conducting additional systematic literature reviews that consider gray literature to compare findings and challenges identified in both white and gray literature, extending existing secondary studies to include the latest knowledge supplements, or exploring additional databases for comprehensive coverage, and enhancing literature review approaches by improving data extraction and synthesis methods, validity, and quality assessment of primary studies.

All these research directions aim to advance our understanding of microservices architecture research and facilitate the development of the best practices, standardized approaches, and improved methodologies in this rapidly evolving field.

The analysis of the directions for future research reveals that they are numerous and diverse. This can be attributed to the relative novelty of microservice architectures, and their unique, yet relatively unexplored nature, characterized by heterogeneity, decentralization, and independence. The breadth of future research directions signifies that the field of microservice analysis remains open to innovations and methodologies.

This study provides a valuable roadmap for researchers, highlighting areas that require further exploration. It also serves as a guide for practitioners, enabling them to assess the progress made thus far and determine which tools and approaches are suitable for practical implementation.

5. Recommendations for future literature reviews

This section provides comprehensive recommendations for further literature reviews in the field of microservices, encompassing primary, secondary, and tertiary studies. These recommendations are drawn directly from the research findings of this study and aimed at enhancing existing research and enriching the knowledge base on microservices. Recommendations relate to quality assessment of selected studies, microservices architecture, life cycle issues, migration, technical implementation in different domains, and further research directions.

Quality assessment of selected studies. According to guidelines for conducting literature reviews, quality assessment of included studies is typically considered a mandatory component of SLRs, whereas it may not be required for SMSs. Consequently, the quality assessment question Q4 received a low score of 0.37. Therefore, the primary recommendation for addressing this issue is to assess the quality of all the studies selected for inclusion in the review and to establish a minimum value threshold for study inclusion to maintain rigorous standards throughout the review process.

Architecture. Research about microservices architecture exhibits a significant level of research maturity, particularly in the areas of architecture analysis, pattern utilization, and the adoption of various methods for architectural representation. Moreover, considerable attention has been given to exploring quality attributes and their associated trade-offs. However, the research findings indicate that there is a need to assess quality attributes and metrics pertaining to granularity, runtime architecture visualization, as well as specific quality attributes such as context awareness, integrability, fault tolerance, upgrading, and availability.

Life cycle issues. Attention to life cycle issues, particularly the utilization of contemporary approaches like DevOps and continuous software engineering practices (continuous integration, continuous delivery, continuous deployment, and monitoring) increased in recent years. However, human factor issues, domain-driven design, and specific maintenance and operation concerns need more attention in further research.

Migration. The migration of legacy systems, typically characterized by monolithic architectures, to microservice architectures is a challenging research topic. Numerous methods and tools have been proposed and evaluated in real-world settings, forming a substantial knowledge base for practitioners and researchers. Based on the study findings, it is evident that motivation and success factors have not been sufficiently addressed in the existing research requiring further studies.

Technical implementation in different domains. Limited research has been documented in research studies and literature reviews concerning the technical implementation of microservices across various domains. Most of the research focuses on the initial development of microservice-based systems, with limited evaluations conducted in real-world settings. In addition, projects carried out by academic institutions often lack involvement of industry experts. There is an evident need to conduct implementation studies within real settings across different domains and to engage industry experts to gain a more comprehensive understanding of microservice-based system implementations.

Methodological issues for further research. Recommendations in the methodological areas are:

(1) Integration of gray literature aimed for complementing the insights obtained from white literature, as well as for increasing understanding of the industry experts' perspectives and the current state of practice;

(2) Utilization of data extraction and synthesis methods from social sciences that enable identification of patterns in practice and the development of theories necessary for building and extending a knowledge base in this area of software engineering.

6. Conclusions

This tertiary study focuses on publication trends, research topics, domains of implementation, and future research challenges in the domain of microservices. Conducted systematic literature review resulted in the selection of 44 secondary studies that are used for deriving findings. Based on the emerging findings, recommendations for further literature reviews are discussed. The main contributions of this study are detailed and structured recommendations for future literature reviews, which include improvement of quality assessment of analyzed studies, more detailed analysis of architecture quality attributes, analysis of implementation in various domains of business and human life, exploration of human factors and organizational issues, and addressing maintenance and operation challenges. In addition, the inclusion of domain experts in the preparation and implementation of these literature reviews is recommended for increasing the accuracy and validity of the findings. And finally, the creation of multidisciplinary teams with experts from social sciences (e.g., sociologists, psychologists, economists) will enable a more comprehensive approach to the analysis of human and organizational factors at different stages of the microservice life cycle, resulting in more comprehensive and reliable literature reviews.

From the methodological standpoint, it is recommended to use qualitative social science methods to obtain more structured findings and methodologically grounded data analysis of unstructured text in analyzed studies. This research recommendation will lead to the development of theories about the practice, increasing the knowledge base in this area of software engineering.

References

[1]. Cerny T., Donahoo M. J., Pechanec J. Disambiguation and Comparison of SOA, Microservices and Self-Contained Systems. In Proc. of the International Conference on Research in Adaptive and Convergent Systems, 2017. pp. 228-235. doi: 10.1145/3129676.3129682.

[2]. Newman S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Sebastopol, CA, USA, 2021.

[3]. Dragoni N., Giallorenzo S., Lafuente A. L., Mazzara M., Montesi F., Mustafin R., Safina L. Microservices: Yesterday, Today, and Tomorrow. In Mazzara M. and Meyer B. (eds.) Present and Ulterior Software Engineering. Springer International Publishing, Cham, Switzerland, 2017. pp. 195-216. doi: 10.1007/978-3-319-67425-4_12.

[4]. Alaasam A. B., Radchenko G., Tchernykh A. Refactoring the Monolith Workflow into Independent Micro-Workflows to Support Stream Processing. Programming and Computer Software, 47(8), 2021, pp. 591–600. doi: 10.1134/S0361768821080077.

[5]. Baškarada S., Nguyen V., Koronios A. Architecting Microservices: Practical Opportunities and Challenges. Journal of Computer Information Systems, 60(5), 2018, pp. 428-436. doi: 10.1080/08874417.2018.1520056.

[6]. Niño-Martínez V. M., Ocharán-Hernández J. O., Limón X., Pérez-Arriaga J. C. A Microservice Deployment Guide. Programming and Computer Software, 48(8), 2022, pp. 632-645. doi: 10.1134/S0361768822080151.

[7]. Larrucea X., Santamaria I., Colomo-Palacios R., Ebert C. Microservices. IEEE Software, 35(3), 2018, pp. 96–100. doi: 10.1109/MS.2018.2141030.

[8]. Ebert C., Gallardo G., Hernantes J., Serrano N. DevOps. IEEE Software, 33(3), 2016, pp. 94–100. doi: 10.1109/MS.2016.68.

[9]. Bosch J. Continuous Software Engineering. Springer International Publishing, Cham, Switzerland, 2014.

[10]. Bruce M., Pereira P. A. Microservices in Action. Manning Publications, Shelter Island, NY, USA, 2019.

[11]. Abdellatif M., Shatnawi A., Mili H., Moha N., El Boussaidi G., Hecht G., Privat J., Guéhéneuc Y-G. A taxonomy of service identification approaches for legacy software systems modernization. Journal of Systems and Software, 173, 2021, pp. 110868. doi: 10.1016/j.jss.2020.110868.

[12]. Henry A., Ridene Y. Migrating to Microservices. In Bucchiarone A., Dragoni N., Dustdar S., Lago P., Mazzara M., Rivera V., Sadovykh A. (eds.) Microservices: Science and Engineering. Springer International Publishing, Cham, Switzerland, 2020, pp. 45-72. doi: 10.1007/978-3-030-31646-4_3.

[13]. Stojkov A., Stojanov Z. Review of methods for migrating software systems to microservices architecture. Journal of engineering management and competitiveness, 11(2), 2021, pp. 152-162.

[14]. Valdivia J. A., Lora-González A., Limón X., Cortes-Verdin K., Ocharán-Hernández J. O. Patterns related to microservice architecture: a multivocal literature review. Programming and Computer Software, 46(8), 2020, pp. 594–608. doi: 10.1134/S0361768820080253.

[15]. Kitchenham B. Procedures for Performing Systematic Review. Technical report TR/SE-0401. Software Engineering Group. Department of Computer Science. Keele University, Keele, UK, 2004.

[16]. Petersen K., Feldt R., Mujtaba S., Mattsson M. Systematic Mapping Studies in Software Engineering. In Proc. of the 12th International Conference on Evaluation and Assessment in Software Engineering, 2008. pp. 68-77.

[17]. Kitchenham B., Budgen D., Brereton P. Evidence-based software engineering and systematic reviews. CRC Press, Boca Raton, FL, USA, 2016.

[18]. Garousi V., Felderer M., Mäntylä M. V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Information and Software Technology, 106, 2019, pp. 101-121. doi: 10.1016/j.infsof.2018.09.006.

[19]. Arvanitou E.M., Ampatzoglou A., Bibi S., Chatzigeorgiou A., Deligiannis I. Applying and Researching DevOps: A Tertiary Study. IEEE Access, 10, 2022, pp. 61585-61600. doi: 10.1109/ACCESS.2022.3171803.

[20]. Cadavid H., Andrikopoulos V., Avgeriou P. Architecting systems of systems: A tertiary study. Information and Software Technology, 118, 2020, pp. 106202. doi: 10.1016/j.infsof.2019.106202.

[21]. Delavari V., Shaban E., Janssen M., Hassanzadeh A. Thematic mapping of cloud computing based on a systematic review: a tertiary study. Journal of Enterprise Information Management, 33, 2019, pp. 161-190. doi: 10.1108/JEIM-02-2019-0034.

[22]. Hoda R., Salleh N., Grundy J., Tee H. M. Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology, 85, 2017, pp. 60-70. doi: 10.1016/j.infsof.2017.01.007.

[23]. Raatikainen M., Tiihonen J., Männistö T. Software product lines and variability modeling: A tertiary study. Journal of Systems and Software, 149, 2019, pp. 485-510. doi: 10.1016/j.jss.2018.12.027.

[24]. Tran H. K. V., Unterkalmsteiner M., Börstler J., bin Ali N. Assessing test artifact quality-A tertiary study. Information and Software Technology, 139, 2021, pp. 106620. doi: 10.1016/j.infsof.2021.106620.

[25]. Liu X., Li S., Zhang H., Zhong C., Wang Y., Waseem M., Babar M. A. Research on Microservice Architecture: A Tertiary Study. SSRN Electronic Journal, 2022, p. 23. doi: 10.2139/ssrn.4204345.

[26]. Costa D. I. C., Filho E. P. e S., da Silva R. F., Quaresma Gama T. D. de C., Cortés M. I. Microservice Architecture: A Tertiary Study. In Proc. of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse, 2020. pp. 61-70. doi: 10.1145/3425269.3425277.

[27]. Wohlin C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In Proc. of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014. pp. 38:1-38:10. doi:10.1145/2601248.2601268.

APPENDIX A. List of secondary studies

[SS01] Tighilt, R., Abdellatif, M., Moha, N., Mili, H., Boussaidi, G. E., Privat, J., & Guéhéneuc, Y. G. (2020). On the study of microservices antipatterns: A catalog proposal. In Proceedings of the European Conference on Pattern Languages of Programs 2020 (pp. 1-13). https://doi.org/10.1145/3424771.3424812.

[SS02] Chávez, K., Cedillo, P., Espinoza, M., & Saquicela, V. (2019). A systematic literature review on composition of microservices through the use of semantic annotations: solutions and techniques. In 2019 International Conference on Information Systems and Computer Science (INCISCOS) (pp. 311-318). https://doi.org/10.1109/INCISCOS49368.2019.00056.

[SS03] Valdivia, J. A., Limón, X., & Cortes-Verdin, K. (2019). Quality attributes in patterns related to microservice architecture: a Systematic Literature Review. In 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 181-190). https://doi.org/10.1109/10.1109/CONISOFT.2019.00034.

[SS04] Schmidt, R. A., & Thiry, M. (2020, June). Microservices identification strategies: A review focused on Model Driven Engineering and Domain Driven Design approaches. In 2020 15th Iberian Conference on Information Systems and Technologies (CISTI) (pp. 1-6). https://doi.org/10.23919/CISTI49556.2020.9141150.

[SS05] Prasetyo, Y. A. & Suhardi (2018). Microservice Platform for Smart City: Concepts, Services and Technology. In 2018 International Conference on Information Technology Systems and Innovation (ICITSI) (pp. 358-363). https://doi.org/10.1109/ICITSI.2018.8695927.

[SS06] Karabey Aksakalli, I., Çelik, T., Can, A. B., & Tekinerdogan, B. (2021). Deployment and communication patterns in microservice architectures: A systematic literature review. Journal of Systems and Software, 180, 111014. https://doi.org/10.1016/j.jss.2021.111014.

[SS07] Li, S., Zhang, H., Jia, Z., Zhong, C., Zhang, C., Shan, Z., ... & Babar, M. A. (2021). Understanding and addressing quality attributes of microservices architecture: A Systematic literature review. Information and software technology, 131, 106449. https://doi.org/10.1016/j.infsof.2020.106449.

[SS08] Garriga, M. (2017). Towards a taxonomy of microservices architectures. In International conference on software engineering and formal methods (pp. 203-218). https://doi.org/10.1007/978-3-319-74781-1_15.

[SS09] Fritzsch, J., Bogner, J., Zimmermann, A., & Wagner, S. (2019). From monolith to microservices: A classification of refactoring approaches. In International Workshop on Software Engineering Aspects

of Continuous Development and New Paradigms of Software Production and Deployment (pp. 128-141). https://doi.org/10.1007/978-3-030-06019-0_10.

[SS10] Razzaq, A. (2020). A systematic review on software architectures for IoT systems and future direction to the adoption of microservices architecture. SN Computer Science, 1(6), 1-30. https://doi.org/10.1007/s42979020-00359-w.

[SS11] Wang, L., Jiang, Y. X., Wang, Z., Huo, Q. E., Dai, J., Xie, S. L., ... & Jiang, Z. P. (2022). The operation and maintenance governance of microservices architecture systems: A systematic literature review. Journal of Software: Evolution and Process, e2433. https://doi.org/0.1002/smr.2433.

[SS12] Jawaddi, S. N. A., Johari, M. H., & Ismail, A. (2022). A review of microservices autoscaling with formal verification perspective. Software: Practice and Experience, 52(11), 2476-2495. https://doi.org/10.1002/spe.3135.

[SS13] Nguyen, H. X., Zhu, S., & Liu, M. (2022). A Survey on Graph Neural Networks for Microservice-Based Cloud Applications. Sensors, 22(23), 9492. https://doi.org/10.3390/s22239492.

[SS14] Söylemez, M., Tekinerdogan, B., & Kolukısa Tarhan, A. (2022). Challenges and Solution Directions of Microservice Architectures: A Systematic Literature Review. Applied Sciences, 12(11), 5507. https://doi.org/10.3390/app12115507.

[SS15] de Almeida, M. G., & Canedo, E. D. (2022). Authentication and Authorization in Microservices Architecture: A Systematic Literature Review. Applied Sciences, 12(6), 3023. https://doi.org/10.3390/app12063023.

[SS16] Berardi, D., Giallorenzo, S., Mauro, J., Melis, A., Montesi, F., & Prandini, M. (2022). Microservice security: a systematic literature review. PeerJ Computer Science, 8, e779. https://doi.org/10.7717/peerj-cs.779.

[SS17] Vera-Rivera, F. H., Gaona, C., & Astudillo, H. (2021). Defining and measuring microservice granularity – a literature overview. PeerJ Computer Science, 7, e695. https://doi.org/10.7717/peerj-cs.695.

[SS18] Leines-Vite, L., Pérez-Arriaga, J. C., & Limón, X. (2021). Confidentiality and Integrity Mechanisms for Microservices Communication. International Journal of Network Security & Its Applications, 13(6), 85-103. https://doi.org/10.5121/ijnsa.2021.13607.

[SS19] Osses, F., Márquez, G., & Astudillo, H. (2018). An exploratory study of academic architectural tactics and patterns in microservices: A systematic literature review. Avances en Ingenieria de Software a Nivel Iberoamericano, CIbSE, 2018, 71-84.

[SS20] Staegemann, D., Volk, M., Shakir, A., Lautenschläger, E., & Turowski, K. (2021). Examining the Interplay Between Big Data and Microservices–A Bibliometric Review. Complex Systems Informatics and Modeling Quarterly, (27), 87-118. https://doi.org/10.7250/csimq.2021-27.04.

[SS21] Nunes, J. P. K., Bianchi, T., Iwasaki, A. Y., & Nakagawa, E. Y. (2021). State of the art on microservices autoscaling: An overview. Anais do XLVIII Seminário Integrado de Software e Hardware, 30-38. https://doi.org/10.5753/semish.2021.15804.

[SS22] Lelovic, L., Mathews, M., Elsayed, A., Cerny, T., Frajtak, K., Tisnovsky, P., & Taibi, D. (2022). Architectural languages in the microservice era: a systematic mapping study. In Proceedings of the Conference on Research in Adaptive and Convergent Systems (pp. 39-46). https://doi.org/10.1145/3538641.3561486.

[SS23] Cerny, T., Donahoo, M. J., & Trnka, M. (2018). Contextual understanding of microservice architecture: current and future directions. ACM SIGAPP Applied Computing Review, 17(4), 29-45. https://doi.org/10.1145/3183628.3183631.

[SS24] Wolfart, D., Assunção, W. K., da Silva, I. F., Domingos, D. C., Schmeing, E., Villaca, G. L. D., & Paza, D. D. N. (2021). Modernizing legacy systems with microservices: A roadmap. In Evaluation and Assessment in Software Engineering (pp. 149-159). https://doi.org/10.1145/3463274.3463334.

[SS25] Waseem, M., Liang, P., Márquez, G., & Di Salle, A. (2020). Testing microservices architecture-based applications: A systematic mapping study. In 2020 27th Asia-Pacific Software Engineering Conference (APSEC) (pp. 119-128). https://doi.org/10.1109/APSEC51365.2020.00020.

[SS26] Velepucha, V., & Flores, P. (2021). Monoliths to microservices-Migration Problems and Challenges: A SMS. In 2021 Second International Conference on Information Systems and Software Technologies (ICI2ST) (pp. 135-142). https://doi.org/10.1109/ICI2ST51859.2021.00027.

[SS27] Di Francesco, P., Malavolta, I., & Lago, P. (2017). Research on architecting microservices: Trends, focus, and potential for industrial adoption. In 2017 IEEE International Conference on Software Architecture (ICSA) (pp. 21-30). https://doi.org/10.1109/ICSA.2017.24.

[SS28] Pereira-Vale, A., Márquez, G., Astudillo, H., & Fernandez, E. B. (2019). Security mechanisms used in microservices-based systems: A systematic mapping. In 2019 XLV Latin American Computing Conference (CLEI) (pp. 01-10). https://doi.org/10.1109/CLEI47609.2019.235060.

[SS29] Alshuqayran, N., Ali, N., & Evans, R. (2016). A systematic mapping study in microservice architecture. Proceedings-2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, SOCA 2016 pp. 44–51. https://doi.org/10.1109/SOCA.2016.15.

[SS30] Gortney, M. E., Harris, P. E., Cerny, T., Al Maruf, A., Bures, M., Taibi, D., & Tisnovsky, P. (2022). Visualizing Microservice Architecture in the Dynamic Perspective: A Systematic Mapping Study. IEEE Access, 10, 119999-120012. https://doi.org/10.1109/ACCESS.2022.3221130.

[SS31] Santos, R., Soares, P., Rodrigues, E., Maia, P. H. M., & Silveira, A. (2022). How Blockchain and Microservices are Being Used Together: a Systematic Mapping Study. In 2022 IEEE/ACM 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB) (pp. 39-46). https://doi.org/10.1145/3528226.3528371.

[SS32] Niño-Martínez, V. M., Ocharán-Hernández, J. O., Limón, X., & Pérez-Arriaga, J. C. (2021, October). Microservices Deployment: A Systematic Mapping Study. In 2021 9th International Conference in Software Engineering Research and Innovation (CONISOFT) (pp. 24-33). https://doi.org/10.1109/CONISOFT52520.2021.00016.

[SS33] Waseem, M., Liang, P., & Shahin, M. (2020). A systematic mapping study on microservices architecture in DevOps. Journal of Systems and Software, 170, 110798. https://doi.org/10.1016/j.jss.2020.110798.

[SS34] Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. Journal of Systems and Software, 150, 77-97. https://doi.org/10.1016/j.jss.2019.01.001.

[SS35] Hannousse, A., & Yahiouche, S. (2021). Securing microservices and microservice architectures: A systematic mapping study. Computer Science Review, 41, 100415. https://doi.org/10.1016/j.cosrev.2021.100415.

[SS36] Vural, H., Koyuncu, M., & Guney, S. (2017). A systematic literature review on microservices. In International Conference on Computational Science and Its Applications (pp. 203-217). https://doi.org/10.1007/978-3-319-62407-5_14.

[SS37] Taibi, D., Lenarduzzi, V., & Pahl, C. (2019). Continuous architecting with microservices and DevOps: A systematic mapping study. In International Conference on Cloud Computing and Services Science (pp. 126-151). https://doi.org/10.1007/978-3-030-29193-8_7.

[SS38] Razzaq, A., & Ghayyur, S. A. (2022). A systematic mapping study: The new age of software architecture from monolithic to microservice architecture—awareness and challenges. Computer Applications in Engineering Education. https://doi.org/10.1002/cae.22586.

[SS39] Bushong, V., Abdelfattah, A. S., Maruf, A. A., Das, D., Lehman, A., Jaroszewski, E., ... & Bures, M. (2021). On microservice analysis and architecture evolution: A systematic mapping study. Applied Sciences, 11(17), 7856. https://doi.org/10.3390/app11177856.

[SS40] Ghani, I., Wan-Kadir, W. M., Mustafa, A., & Babir, M. I. (2019). Microservice testing approaches: A systematic literature review. International Journal of Integrated Engineering, 11(8), 65-80. https://doi.org/10.30880/ijie.2019.11.08.008.

[SS41] Velepucha, V., Flores, P., & Torres, J. (2019). Migration of monolithic applications towards microservices under the vision of the information hiding principle: a systematic mapping study. In The International Conference on Advances in Emerging Trends and Technologies (pp. 90-100). https://doi.org/10.1007/978-3030-32022-5_9.

[SS42] Yu, D., Jin, Y., Zhang, Y., & Zheng, X. (2019). A survey on security issues in services communication of Microservices-enabled fog applications. Concurrency and Computation: Practice and Experience, 31(22), e4436. https://doi.org/10.1002/cpe.4436.

[SS43] Pahl, C., & Jamshidi, P. (2016). Microservices: A Systematic Mapping Study. In the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) CLOSER (1), 137-146.

[SS44] Parker, G., Kim, S., Al Maruf, A., Cerny, T., Frajtak, K., Tisnovsky, P., & Taibi, D. (2023). Visualizing Anti-Patterns in Microservices at Runtime: A Systematic Mapping Study. IEEE Access, 11, 4434-4442. https://doi.org/10.1109/ACCESS.2023.3236165.

Информация об авторах / Information about authors

Желько СТОЯНОВ имеет PhD степень в области компьютерных наук и информатики. Профессор Нови-Садского университета, технический факультет “Михайло Пупин”, Сербия. Научные интересы: разработка программного обеспечения, жизненный цикл программного обеспечения, бизнес-информатика, качественные исследования, управление знаниями, человеческие аспекты инженерии.

Zeljko STOJANOV – PhD degree in Computer Science and Informatics. Full professor at the University of Novi Sad, Technical Faculty “Mihajlo Pupin,” Serbia. Research interests: software engineering, software life cycle, business informatics, qualitative research, knowledge management, human aspects of engineering.

Илья ХРИСТОСКИЙ имеет степень PhD в области технических наук. Профессор университета “Св. Климент Охридский” в Битоле, кафедра бизнес-информатики на экономическом факультете в Прилепе, Северная Македония. Область научных интересов: моделирование и симуляция, стохастические сети Петри, системная динамика, системы баз данных, бизнес-информатика, управление информацией и знаниями, финансовая эконометрика.

Ilija HRISTOSKI – PhD degree in Technical Sciences. Full professor at the University “St. Kliment Ohridski” in Bitola, Department of Business Informatics at the Faculty of Economics in Prilep, North Macedonia. Research interests: modelling and simulation, stochastic Petri Nets, system dynamics, database systems, business informatics, information and knowledge management, financial econometrics.

Елена СТОЯНОВА имеет степень PhD в области математических наук. Доцент Нови-Садского университета, технический факультет “Михайло Пупин”, Сербия. Область научных интересов: прикладная математика, дифференциальная геометрия, математическое образование.

Jelena STOJANOV – PhD degree in Mathematics. Associate professor at the University of Novi Sad, Technical Faculty “Mihajlo Pupin,” Serbia. Research interests: applied mathematics, differential geometry, mathematics education.

Александра СТОЙКОВА – имеет степень магистра информационных технологий. Аспирант и ассистент преподавателя на техническом факультете “Михайло Пупин” Нови-Садского университета, Сербия. Область научных интересов: разработка программного обеспечения, информационные системы, базы данных, реинжиниринг программных систем.

Aleksandra STOJKOV – MsC degree in Information Technology. PhD student and teaching assistant at the University of Novi Sad, Technical Faculty “Mihajlo Pupin,” Serbia. Research interests: software engineering, information systems, databases, reengineering of software systems.