e-Informatica Software Engineering Journal, Volume 18, Issue 1, 2024, pages: 240104, DOI: 10.37190/e-Inf240104

Migrating a Legacy System to a Microservice Architecture

Kristian Tuusjärvi*, Jussi Kasurinen*, Sami Hyrynsalmi*

*School of Engineering Sciences, LUT University, Finland

kristian.tuusjarvi@student.lut.fi, jussi.kasurinen@lut.fi, sami.hyrynsalmi@lut.fi

Abstract

Background: In software engineering, each software product has a life cycle that at some point results in a decision being made with regard to extending its maintenance or upgrading the system to a new platform and architecture via a re-engineering or migration process. However, sometimes this decision is a non-starter; the technical dept accumulates, and platforms cease to exist, meaning that there will always be a time when extending the life support of a legacy system is no longer simply an option, and the service must be modernized.

Aim: In this paper, we focus on the migration processes, where a legacy system is updated to a microservice architecture, to understand the current state-of-the-art, applied industry practices and potential pitfalls or research gaps in the topic domain. The study aims to explore previous research to find related trends and expose gaps in the literature.

Method: We conducted a systematic mapping study on the research trends within the topic of redesign and re-engineering projects related to microservice architectures to understand what we know about microservices, what the current research trends in the area are, and if possible, what the common nominators for successful migration processes are.

Results: Our observations reveal that most microservice migration research is confined to journal articles and conference proceedings. However, a severe fragmentation in publication venues exists within the field. Furthermore, the focus of the research field is primarily on the transformation phase of the re-engineering process, with the majority of the contributions being managerial in nature, particularly of the process type. Additionally, over 50% of the research conducted is empirical in nature.

Conclusion: Based on the results, microservice migration research is maturing well; most of the research is empirical. The research field is scattered. There are notable technical, managerial, and organizational challenges and differing motivations. To better understand the motivations and challenges of the practitioners, we are going to conduct survey and interview studies within this field.

Keywords: legacy systems, microservice architecture, monolithic architecture, microservice migration

1. Introduction

In the software life cycle, each product reaches a point where a decision has to be made between extending the system maintenance, killing the software, or modernizing and migrating the legacy system into a new architecture and modern platform. In this work,

Submitted: 13 Jun. 2023; Revised: 14 Nov. 2023; Accepted: 15 Nov. 2023; Available online: 17 Nov. 2023

we conducted a systematic mapping study (SMS) [1] to study one such legacy software modernization trend – the trend of migrating monolithic legacy software systems toward microservice architecture (MSA). MSA has become popular due to recent technological advancements, such as cloud computing and containerization, which promise to allow easier service scaling, cost management, maintenance, and faster development cycles, amongst other benefits. Furthermore, the overall digitalization of the business world is forcing companies to search for more dynamic and adaptive software architectures.

1.1. Microservice technology

Microservices are small, self-sufficient processes that interact with each other using messaging protocols, such as Representational State Transfer (REST) [2–4]. MSA is a distributed cloud-native architecture that is based on service architecture (SOA) [5, 6], where developers can create, test, and deploy microservices using different development stacks and platforms [5]. Traditionally, the software has been developed as monolithic, meaning that a single executable handles all the features of a given software system [6].

Three core technologies are often utilized when using MSA. First is cloud computing; microservice based systems often run in a cloud environment where computing resources can be scaled up or down depending on the user traffic. The second is containerization; microservices are often containerized, which enables them to be deployed quickly and managed by container management software, such as Kubernetes [7]. Third, continuous integration and delivery automation processes [8] mean that the entire process, from development and quality assurance to staging and deployment, is automated to enable fast iterations and even roll-backs in case of faulty releases.

1.2. State of the industry

The research related to microservices has seen an increase during the past decade. In around 2010, the term microservice started to rise in popularity [9], with many large companies using MSA to build their software systems. For example, Sound Cloud [10], Netflix [11], and Uber [12] have adopted MSA as their service architecture. However, as the MSA model is mainly based on industry-driven needs and development, this might also correlate to the need for more literature and stricter definitions of MSA [13].

Large and complex monolithic software systems are prime candidates for MSA migration. Software systems built with MSA are less prone to accumulating complexity during their lifetimes. Software developed with distributed architecture is more self-contained than software with a monolithic architecture. It allows components to be developed and maintained separately from other parts, allowing the software system to stay robust and responsive [4]. Companies often choose to adopt MSA depending on their needs, as MSA has a reputation for having quality attributes such as availability, flexibility, maintainability, scalability, and loose coupling as built-in features [14, 15].

The motivation for this research arose from a previous study where we documented migrating from a legacy system to an MSA. During the research process, we found a lack of related research into migrating from legacy systems to MSA, which was unusual as the revision, replacement, and re-engineering work of legacy systems is not in any way an uncommon activity in the software industry. For this reason, we decided to conduct a more comprehensive literature review on migrating from legacy software to MSA.

This study aims to explore previous research to find related trends and expose gaps in the literature. In this study, we want to review the literature on migrating to MSA from legacy systems. More precisely, we want to study the research trends (publication venues and periods), the migration process phase in which migration to MSA research is focused (reverse engineering, transformation, or forward engineering), and the research contribution types to understand how MSA migration processes have been investigated in the prior works. The rest of this research paper is structured as follows: related work, methods, analysis, threats to validity, discussion, and conclusion.

2. Related work

This section discusses the research related to our topic: migrating to MSA. We will give short summaries of the related research papers, discuss their relevance to this study, and synthesize how they motivated it.

Carrasco et al. [16] conducted a literature study on microservice migration bad smells. They wanted to know what architectural and migration-related bad smells are common with MSA and how to avoid them. Their study identified nine common pitfalls as architectural smells from 58 sources, including academic and gray literature, between 2014 and 2018. The nine pitfalls are divided into five new architectural bad smells and four migration-related smells. The most common pitfalls were single-layered teams, including multiple services in one container, being greedy with containers, and simultaneously rewriting the entire system for microservices. They offer solutions for detecting and solving the pitfalls mentioned earlier. Carrasco et al.'s [16] research has an architectural focus, researching architectural pitfalls. Their study relates to ours by investigating the migration process toward MSA. However, they focus on specific architectural problems, whereas our work is more general and considers the literature and its visible trends. They also use grey literature, while we focus on academia [16].

In 2018, Knoche et al. [17] conducted a survey study on German professionals with 71 participants. They studied the primary drivers for MSA adoption, barriers to adoption, the goals of modernizing MSA, and how data consistency affects performance. They conclude that the prime drivers for modernization are scalability, maintainability, and time to market. The skills of developers and other staff were seen as the main barriers. As for goals, early adopters desired scalability from MSA, while traditional companies wanted maintainability. Performance was considered a minor issue. The authors call for similar work from other countries. They also researched migration to MSA. However, they conducted an empirical study in the form of a survey study. Furthermore, their research focuses on the motivation for migration and the barriers to adoption [17].

Velepucha et al. [18] conducted an SMS on migrating to microservices. The study included 32 primary research papers from 2012 to 2020. The research papers were only from academia. In their study, Velepucha et al. wanted to determine the types of migration proposals present in the literature and which are based on the information hiding principle. They found multiple proposals related to migrating to microservices, for example, those using DevOps, cloud computing, and performance in infrastructure. They identified that only two papers discussed migration principles, of which only one was the information hiding principle. None of the research papers proposed a software development principle to migrate from a monolithic system to MSA [18]. Our work shares some similarities with Velepucha et al. [18] as they also classified research papers based on the type of research.

However, unlike our approach, they did not use Wieringa et al.'s [19] classification. We limited our review to literature published after 2015 as we focused on the current state of the art rather than the early stages of MSA. Additionally, we analyzed various metrics related to research approaches and publication year, venue, and type.

Hassan et al. [20] conducted a large SMS based on academic and industrial literature. They analyzed 877 publications from various sources between January 2013 and April 2020. Their study had two objectives: first, to study the transition process to microservices and, second, to understand the fundamental problem of transitioning, the granularity problem of transitioning to microservices. Additionally, they classified the analyzed literature [20]. Our research shows some similarities with the SMS conducted by Hassan et al. [20] regarding MSA migrations. Like us, they also used the classification schema by Wieringa et al. [19] to classify their research. However, their research mainly focused on the issue of granularity when transitioning to microservices. Additionally, their study included grey literature, which is not the case in our research.

Auer et al. [21] conducted an interview study. They researched why companies migrate to MSA, the information metrics used, and the most helpful information metrics. Their interview study included 52 respondents from software development practitioners over five days in 2018. Based on the interviews, the authors generated an assessment framework to ease the decision-making when migrating to MSA. Their interviews with practitioners found that the most common reason for migrating to MSA was to improve maintainability. Other common reasons were independent teams, deployability, and cost, whereas modularity, complexity, fault tolerance, scalability, and reusability were less popular characteristics. The research by Auer et al. [21] relates to our study by discussing the MSA migration process. However, their study focuses on the motivation of the practitioners and the metrics they use to collect information. In contrast, our study focuses more on general information about the research field, the re-engineering phase, and the research contribution types. Moreover, their study is a survey study rather than an SMS [21].

Razzaq et al. [22] conducted an SMS study on MSA migrations. The study included 73 primary research papers from 2010 to 2021. Their goal was to evaluate the state and practice of MSA literature. They researched publication trends and venues, research focus, migration approaches and challenges, success factors post-migration, and the potential for industrial adoption. Related to the publication trends, they note that the volume of publications is progressively rising. They suggest future researchers focus on MSA in the context of the Internet of Things [22]. Razzaq et al. [22] and our study analyze publication trends, venues, and types. However, our approach differs as we delve into the publication contributions and re-engineering phase the research focuses on.

Our study on SMS migration to MSA did not have much directly comparable research available. However, we have identified the research papers that are most similar to our study in terms of the research period, method, contributions, and goals. The list of these research papers can be found in Table 1. While we found more secondary research related to MSA from different perspectives, there was no MSA-related research available in 2014, according to secondary sources cited in [23] and [24]. There has been a significant increase in MSA-related research since 2015 [23] and 2016 [24]. Waseem et al. [25] also reported a growth in MSA-related research between 2015 and 2018 [25]. Pahl et al. [23] suggest that follow-up research should be directed toward aspects such as microservices migration [23]. Furthermore, multiple authors highlight the novelty of the MSA research field [13, 20, 23, 24]. More recently (2022), Razzaq et al. [22] reported a progressively rising number of research from year to year.

Table 1. Main findings from the studies similar to ours

Author(s)	Sources	Main findings	Type	Year
Carrasco et al. [16]	58 (including grey literature)	Their study identified nine common pitfalls as architectural smells.	SMS	2018
Velepucha et al. [18]	32	They found that only two papers discussed migration principles, one of which was the information hiding principle.	SMS	2020
Hassan et al. [20]	877 (including grey literature)	They found and defined the granularity problem present in MSA migrations.	SMS	2020
Razzaq et al. [22]	73	They found that the number of publications is progressively rising.	SMS	2022

Many of the studies in our related research indicate the novelty of the research field and room for more research from different perspectives, including research within the subfield of migrating to MSA. We only found six directly related studies [16–18, 20–22], from which only four [16, 18, 20, 22] were SMSs. Our research attempts to fill this gap by summarizing the current state of research related to MSA migrations and by observing the specific re-engineering phase where the research is focused.

3. Methods

This section goes through the main phases of our research process; we followed the SMS guidelines described by Petersen et al. [1], also illustrated in Figure 1. An SMS is a research methodology that categorizes research papers and visualizes these to create a map of the researched subject [26] in the form of a conceptual map, categorization, or some other layout. An SMS is recommended as the research methodology in software engineering when the research area is still emerging and a substantial quantity of high-quality studies have yet to be completed. However, the data collection and analysis scheme is not as in-depth in similar systematic literature review models [26]. SMSs are commonly used in several domains [27], with medicine and software engineering being the most prominent areas of application [28]. The main phases of our research process are defining research questions, conducting the search, screening papers, keywording, and data extraction and mapping, as defined by Petersen et al., with all the phases having an output forwarded to the next stage; the final product is the systematic map that visualizes the results.

Based on our background work, we defined three research questions for this study. We wanted to research the period from 2015 to 2023 to capture the trends found in the microservice migration literature during that time and minimize the number of non-related

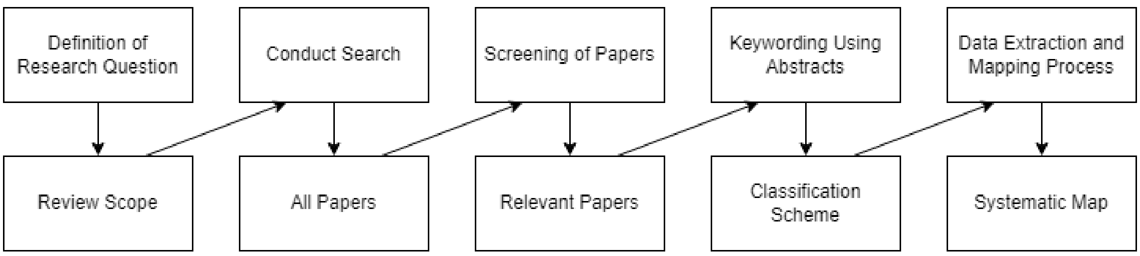

Figure 1. Systematic mapping study process as defined by Petersen et al. [1]

topics that might share similar terminology; we also wanted to analyze the research trends, the focus of research within the re-engineering process, and the contributions of the existing research work. These goals are reflected in the research questions as follows:

RQ1 – What are the research approaches implemented by the researchers?
This research question provides us with general information about the research field, which is important for analyzing the current trends in the literature.

RQ2 – What phase of the re-engineering process is addressed by the research papers?
The answer to this research question allows us to analyze which part of the re-engineering process is the most researched and where gaps in the research exist. We used the horseshoe model to define the different phases of re-engineering [29].

RQ3 – What are the contribution types of the research papers?
This question explores the concrete contributions to microservice migration research and the broader scientific community. We developed a contribution-type classification using an iterative process based on an example by Petersen et al. [1].

We used Google Scholar's research database to get research material for this SMS. We chose to use Google Scholar because it obtains research material from many different publishers and databases, such as ACM, IEEE, and Springer while having few to no limitations regarding the research domains or areas of expertise. Google Scholar can query articles with words using the following options: all, exact phrase, at least one, and without. These different options can be combined. It is possible to select where in the article the words appear: anywhere in the article or only in the title. Additionally, the author and publisher of the article can be specified, and publication dates can be indicated. We used the default search, which returned articles that included all the words from our search string, and looked for the latter anywhere in the article [30]. The only limitations we set for Google Scholar were not to include patents and the time frame. Google Scholar orders the results according to their relevance based on the full text, source, author, and number of citations [31].

The search was conducted using a search string developed by testing keywords against the database. The goal was to find a search string that yielded all the meaningful research papers that reflect the research area. The search results were evaluated manually to estimate whether they matched the research area. The evaluation was performed by the number of citations, the text's topic (related to the research topic), the author, and the source. As suggested by Petersen et al. [1], the search string reflected the research questions. The final search string was "microservices legacy software migration modernization". We conducted two searches on Google Scholar: the first in 8/2020 (from 2015 to 8/2020) and the second in 8/2023 (from 2020 to 8/2023) to update the primary research papers. The search yielded 487 initial results in 2020 and an additional 821 results in 2023, giving a total of 1308 search results. Because Google Scholar's results change over time, we have saved the original search results list to a cloud service for repeatability purposes (https://dx.doi.org/10.6084/m9.figshare.24426889).

Selection criteria were applied to the search results to filter out the unwanted results. The filtering used the inclusion and exclusion criteria shown in Tables 2 and 3, respectively. The number of citations was not considered as it would have given a less realistic view of the research area. Figure 2 shows the search process and application of the selection criteria. First, we applied our selection criteria to the initial research and removed any duplicates that could be identified. Later, during the data extraction phase, a few papers were removed as they did not fit the scope of this study.

Table 2. The inclusion criteria applied to our search results

ID	Inclusion criteria
I1	Research with more than four pages of text.
I2	Research from 2015 to 8/2020 in the first search and 2020 to 8/2023 in the second search
I3	Research in the following publication formats: books, research papers, conference papers, and journal articles.
I4	Research written in English.
I5	Research that is publicly accessible.
I6	Research that explicitly discusses the theme of this SMS (i.e., the migration of legacy software systems to MSA).

Table 3. The exclusion criteria applied to our search results

ID	Exclusion criteria
E1	Research duplicates matched with regard to the author, publication year, and title.
E2	Research that is not peer-reviewed.
E3	Research discussing microservices but not the process for migrating to MSA.

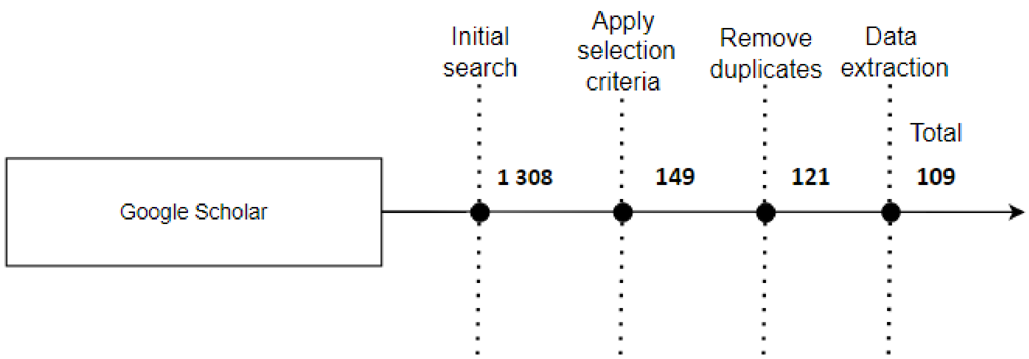

Figure 2. The process of screening the research papers

We used existing and generic classification methods to manage the data extraction process for the first two research questions. We generated a classification through the keywording process for the third research question. For the research approaches (RQ1), we used the following parameters: publication type, venue, and date; publisher; and research strategy. This research question queries general information about the research field. For the publication type, we use a simple categorization: journal, workshop paper, book, or conference proceeding and the publication venue, publisher, and date. Finally, we used the classification by Wieringa [19] to classify the research strategies, illustrated in Table 4. This categorization method is general and does not depend on any specific research field [1].

For the re-engineering phase (RQ2), we used the horseshoe model to divide the re-engineering process into reverse engineering, transformation, and forward engineering. Reverse engineering includes the acts of understanding, abstracting, and extracting a high-level model of the source system. For example, this could include software that helps understand existing systems or identifies microservice candidates. Transformation improves, restructures, and extends the previously mentioned high-level system model. For example, this could be in the form of processes or tools that help shape the new architecture. Finally, forward engineering generates a new, improved system [29]. For example, this could include guidelines and tools that help generate the new system in practice. We categorized the

Table 4. Classification of the primary papers identified, based on the principles presented in Wieringa et al. [19]

Category	Description
Validation Research	The techniques investigated are novel and have not yet been implemented. The techniques used are, for example, experiments (i.e., work done in the lab).
Evaluation Research	Techniques are implemented in practice and evaluated. This type of research shows how a technique is implemented in practice (solution implementation) and the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes identifying problems in the industry.
Solution Proposal	A solution for a problem is proposed; the solution can be either novel or a significant extension of an existing technique. A small example or a good line of argumentation shows the solution's potential benefits and applicability.
Philosophical Papers	These papers sketch a new way of looking at existing things by structuring the field in the form of a taxonomy or conceptual framework.
Opinion Papers	These papers express the opinion of somebody on whether a certain technique is good or bad or how things should be done. They do not rely on related work and research methodologies.
Experience Papers	Experience papers explain what and how something has been done in practice based on the author's personal experiences.

research papers by reading them and assigning them to one or more categories based on their topics and content, as a research paper can discuss multiple phases of the re-engineering process.

Finally, to assess the research contributions (RQ3), we classified them using an iterative keywording process by Petersen et al. [1]. We read through the studies and collected keywords and concepts representing their contributions. The context of the research was also identified. After collecting the keywords, we combined them to form a classification scheme. The classification scheme we ended up with consists of the following: process, analysis, tool, method, best practices, experience sharing, and metrics. A process is a structured approach to migrating to MSA. Analysis covers papers focusing on migration's issues and benefits and other literature. The tool assists in the migration process (i.e., software that can help with the migration process). A method provides systematic ways to achieve specific tasks within the broader migration process. Best practices are guidelines based on successful migrations. Experience sharing offers practical insights from real-world migration scenarios. The metrics can measure and evaluate different aspects of the migration process.

4. Analysis

This study aims to discover trends related to legacy software modernization, specifically migrating from legacy applications to microservices. The study was conducted as an SMS. A pool of 1308 research papers was the starting point. After the inclusion and exclusion, 109 were chosen for further analysis and categorization. In this section, we analyze the results of the categorization process.

4.1. Research areas and approaches (RQ1)

In this section, we review the results of the first research question: "(RQ1) What are the research approaches?" The first research question queried the research approaches and

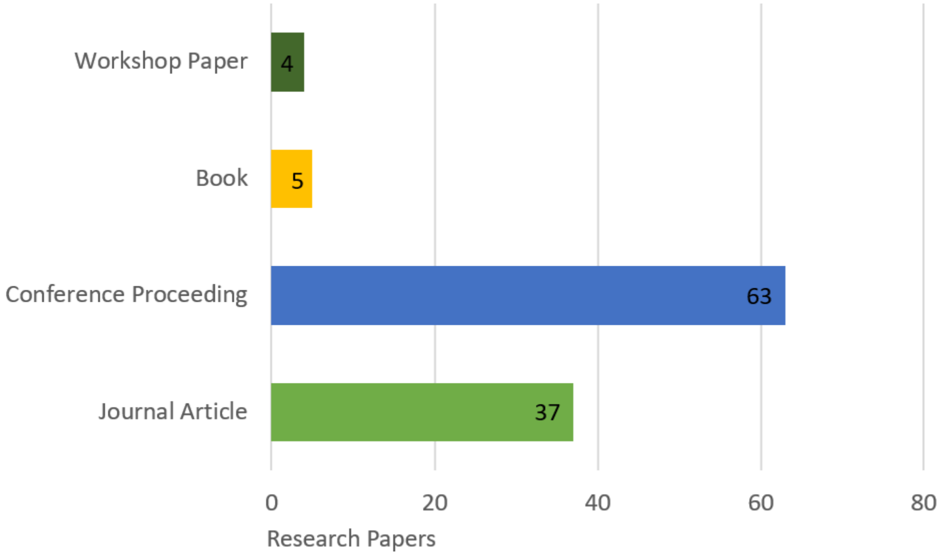

Figure 3. Publication type totals. References in Table 5

Table 5. References for the publication types

Publication type	ID
Workshop Paper	P25, P29, P36, P41
Book	P12, P15, P22, P79, P106
Conference Proceeding	P2, P3, P4, P5, P6, P7, P11, P13, P16, P17, P18, P19, P23, P26, P27, P28, P30, P31, P32, P33, P34, P35, P37, P38, P39, P40, P42, P43, P44, P46, P47, P48, P49, P50, P52, P54, P55, P57, P60, P61, P65, P66, P67, P68, P70, P72, P73, P76, P77, P80, P82, P83, P86, P88, P89, P94, P98, P99, P101, P102, P103, P104, P108
Journal Article	P1, P8, P9, P10, P14, P20, P21, P24, P45, P51, P53, P56, P58, P59, P62, P63, P64, P69, P71, P74, P75, P78, P81, P84, P85, P87, P90, P91, P92, P93, P95, P96, P97, P100, P105, P107, P109

publication information in relation to the primary research. These include the publication venue, publication time, publication type, and research type. The publication information helps us understand the current state of the research field and find any gaps that need to be filled.

We divided the publications into four groups: journal articles, conference papers, books, and workshop papers. Figure 3 shows the total numbers of the different publication types. Most research papers were either conference papers (58%) or journal articles (34%). In contrast, there is a relatively small amount of workshop papers.

Figure 4 shows the number of publications published annually from 2015 to 2023. It should be noted that 2023 has only partial data, as this study was conducted during the fall of that year. Also notable is that, in our primary research, there were no publications from 2015; the first MSA migration-related publications in our primary research material are from 2016. The lack of research is understandable since microservice technology only started to attract interest from 2010 onward [9]. We can observe a significant increase in publications from 2018, with a slight decrease in 2019 and a steady level thereafter. Regarding the publication types, we can see that conference papers and journal articles have more stable publication numbers than books and workshop papers.

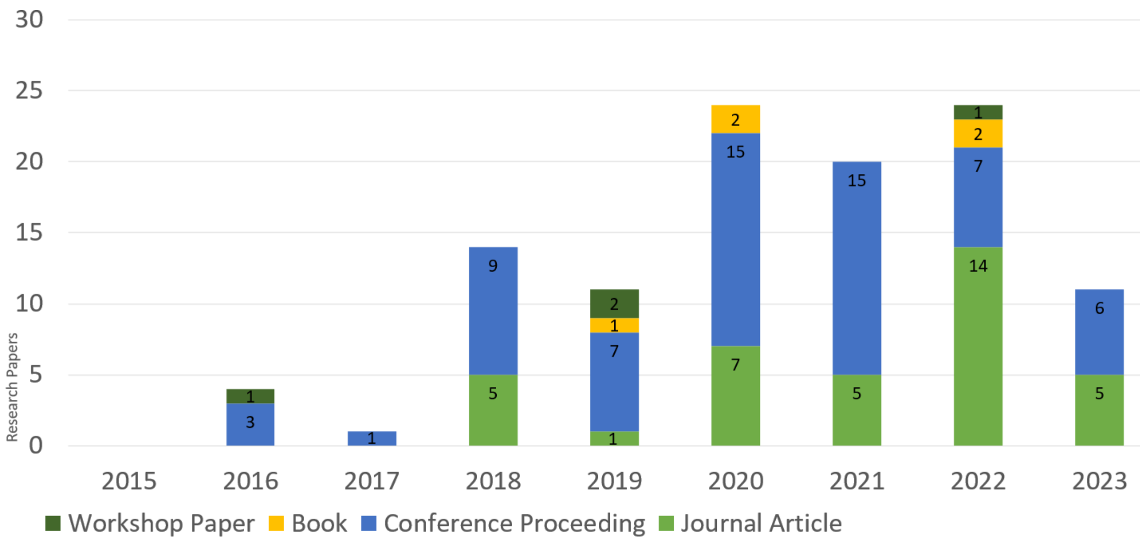

Figure 4. Publication trends from 2015 to 2023. References in Table 6

Table 6. References for the publication types per year

	Book	Conference proceeding	Journal article	Workshop paper
2016		P3, P5, P6		P36
2017		P18		
2018		P2, P13, P19, P23, P26, P30, P31, P32, P33	P1, P9, P10, P14, P21	
2019	P15	P7, P17, P27, P28, P35, P37, P48	P8	P25, P29
2020	P12, P22	P4, P11, P16, P34, P38, P39, P43, P65, P66, P67, P73, P77, P99, P103, P108	P20, P24, P62, P85, P92, P100, P107	
2021		P40, P42, P44, P46, P50, P52, P55, P60, P70, P72, P80, P83, P86, P89, P104	P58, P69, P93, P96, P105	
2022	P79, P106	P47, P57, P61, P68, P82, P94, P102	P45, P51, P53, P59, P74, P75, P78, P81, P84, P87, P91, P95, P97, P109	P41
2023		P49, P54, P76, P88, P98, P101	P56, P63, P64, P71, P90	

With regard to the various publication venues listed in Table 7, we can see severe fragmentation; only a few research papers are published through the same publication venues, meaning that 109 research papers are published through 91 different ones. The exceptions are the IEEE ICSA-C conference (4), IEEE Software Journal (3), IEEE International Conference on Software Architecture (ICSA) (3), Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (3), and International Journal of Advanced Computer Science and Applications (IJACSA) (3), all with three or more publications. The publication venues also have distinct focus areas, for example, software architecture (i.e., ICSA, ICSA-C), cloud computing (i.e., ESOCC), software maintenance (i.e., ICSME, VEM), development operations (i.e., DEVOPS), software refactoring (i.e., IWoR), data analysis (i.e., SADASC), and software engineering (i.e., APSEC, SBES, and SEAA). The rest of the publication venues are listed in Table B1 in Appendix B.

Table 7. Publication venues with more than one publication from our primary research papers. The rest of the publication venues are listed in Appendix B

Publication venue	#	ID
IEEE International Conference on Software Architecture Companion (ICSA-C)	4	P11, P34, P41, P54
IEEE Software	3	P1, P9, P10
IEEE International Conference on Software Architecture (ICSA)	3	P26, P61, P88
Euromicro Conference on Software Engineering and Advanced Applications (SEAA)	3	P38, P60, P66
International Journal of Advanced Computer Science and Applications (IJACSA)	3	P51, P63, P107
Software: Practice and Experience	2	P14, P64
On the Move to Meaningful Internet Systems: OTM Workshops	2	P16, P43
International Journal of Computer Applications (IJCA)	2	P24, P92
Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment (DEVOPS)	2	P25, P29
Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS)	2	P42, P89
Empirical Software Engineering (EMSE)	2	P56, P59
International Conference on Advanced Information Systems Engineering (CAiSE)	2	P104, P108

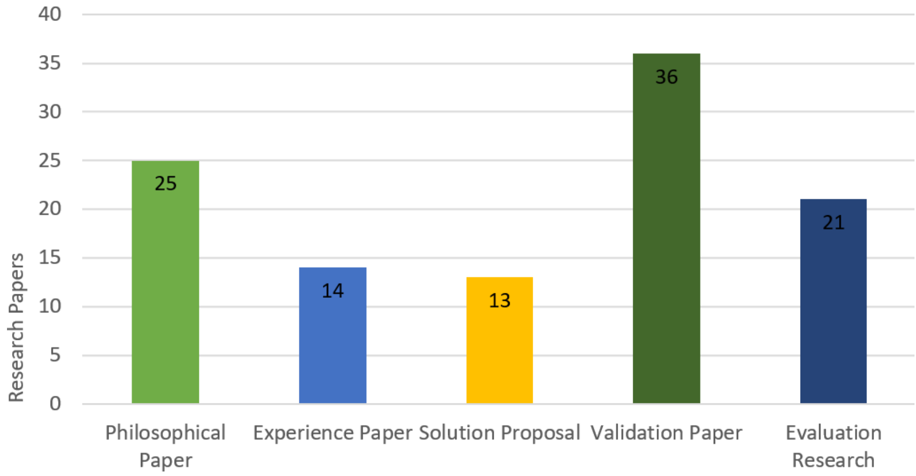

Figure 5. Research types of the papers discussing re-engineering or system migration. References in Table 8

The research types were analyzed using the classification schema by Wieringa et al. [19]. We chose this classification method because of its wide use in other systematic mapping studies (e.g., Di Francesco [24], Agilar et al. [32], Alshuqayran et al. [13]) and because it is often possible to classify a study without reading the whole paper, which saves time [1]. The classification schema consists of the following categories: validation research, evaluation research, solution proposals, philosophical papers, opinion papers, and experience papers, as listed in Table 2.

Figure 5 shows the distributions of the research types. The most used research type is validation research (33%), which investigates novel techniques in controlled environments. Philosophical papers (23%) are the second most popular research type; these papers review the research area and create taxonomies and conceptual frameworks. The third most popular research type is evaluation research (19%), meaning that many researchers are

Table 8. References for research types

Research type	ID
Philosophical paper	P7, P8, P9, P25, P26, P27, P28, P31, P35, P37, P41, P42, P44, P48, P56, P58, P64, P68, P71, P74, P79, P80, P84, P90, P100
Experience paper	P5, P10, P11, P12, P19, P30, P38, P47, P62, P66, P72, P82, P87, P103
Solution proposal	P15, P16, P17, P22, P40, P46, P53, P60, P63, P67, P81, P85, P106
Validation paper	P4, P6, P29, P32, P33, P34, P43, P49, P50, P51, P52, P55, P59, P61, P69, P70, P75, P76, P78, P83, P86, P88, P89, P91, P92, P93, P95, P96, P98, P99, P101, P102, P104, P105, P107, P108
Evaluation research	P1, P2, P3, P13, P14, P18, P20, P21, P23, P24, P36, P39, P45, P54, P57, P65, P73, P77, P94, P97, P109

testing their techniques in practice and showing the benefits and drawbacks of those techniques (evaluating their implementations). The fourth most popular research type is experience research (13%), indicating that many researchers in this field only reported their experiences. Finally, the least popular research types are solution proposals (12%) and opinion papers (0%).

The research types can be divided into empirical and non-empirical research. Non-empirical types are solution proposals, opinion papers, experience papers, and philosophical papers. Empirical study types are validation and evaluation research. Most of the studies (52%) are empirical and use verified data and observations to support research results, while only (48%) are non-empirical. Empirical studies are critical for validating theories, models, tools, and other migration-related artifacts.

4.2. Re-engineering phase (RQ2)

The second research question deals with the re-engineering phase: "(RQ2) What phase of the re-engineering process is addressed by the research papers?" To research this question, we utilized the horseshoe model, which divides the re-engineering process into three phases: reverse engineering, transformation, and forward engineering [29]. We classified the primary research papers according to the three re-engineering phases. The distribution of the re-engineering phases can be seen in Figure 6. It should be noted that a research paper can cover multiple re-engineering phases, meaning that the sum of the results is not the sum of the research papers. Almost half of the research papers (48%) focused on transformation,

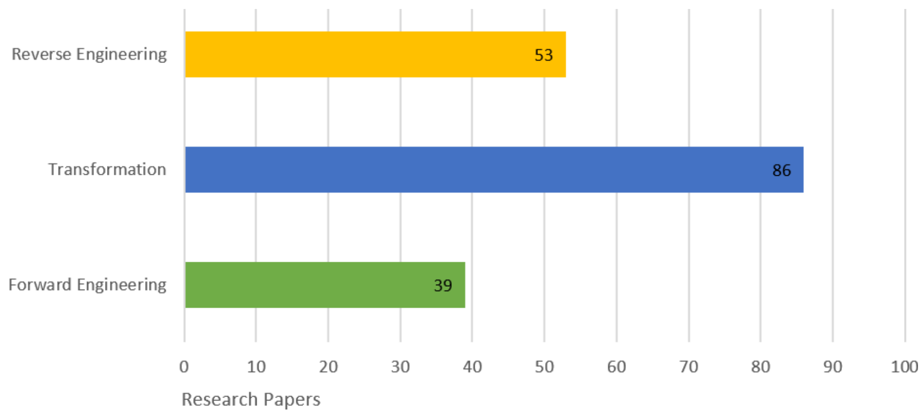

Figure 6. The re-engineering phases described as parts of the horseshoe model. References in Table 9

Table 9. References for the re-engineering phases

Re-engineering phase	ID
Reverse engineering	P1, P3, P10, P11, P13, P17, P19, P22, P23, P26, P33, P35, P36, P37, P38, P39, P40, P41, P42, P43, P44, P45, P48, P49, P52, P53, P56, P59, P60, P61, P66, P67, P68, P71, P73, P76, P78, P80, P85, P86, P89, P90, P92, P96, P98, P99, P100, P101, P102, P103, P104, P108, P109
Transformation	P2, P4, P5, P6, P7, P8, P9, P10, P11, P12, P14, P15, P16, P17, P19, P21, P22, P24, P25, P26, P27, P28, P29, P30, P31, P32, P33, P34, P35, P36, P37, P38, P39, P40, P41, P42, P43, P44, P45, P48, P49, P50, P51, P52, P54, P55, P56, P57, P58, P60, P62, P63, P65, P66, P67, P69, P70, P71, P72, P74, P75, P77, P79, P80, P81, P82, P83, P84, P85, P86, P87, P88, P90, P91, P92, P93, P94, P95, P97, P100, P103, P105, P106, P107, P108, P109
Forward engineering	P2, P5, P8, P10, P11, P12, P16, P18, P20, P22, P24, P26, P31, P37, P38, P40, P41, P42, P44, P45, P46, P47, P48, P49, P52, P56, P60, P64, P66, P67, P71, P80, P85, P86, P90, P92, P100, P103, P109

while around a third (30%) covered reverse engineering, and around a fifth (22%) focused on forward engineering, as shown in Figure 6.

4.3. Contributions to the domain (RQ3)

This section answers the third research question: "(RQ3) What are the contribution types of the research papers?" Seven different contribution categories were identified from the research papers: process, experience sharing, best practice, analysis, method, tool, and metric. Figure 7 shows the distribution of the research contributions. Nearly half (44%) of the research papers contributed to the research topic with a process, most often describing the actions that can or should be taken to accomplish the goal of migrating to MSA. After the process, the next most popular contribution type was analysis (24%), followed by tool (9%) and method (8%). Conversely, best practice (7%), experience sharing (6%), and metrics (2%) were the least common contribution types.

To better understand the contributions, they can be further categorized according to the contribution type into technical and managerial contributions. Agilar et al. [32] used this categorization method in their research for a similar purpose. A managerial contribution describes a process, method, or approach that manages the migration process. A technical

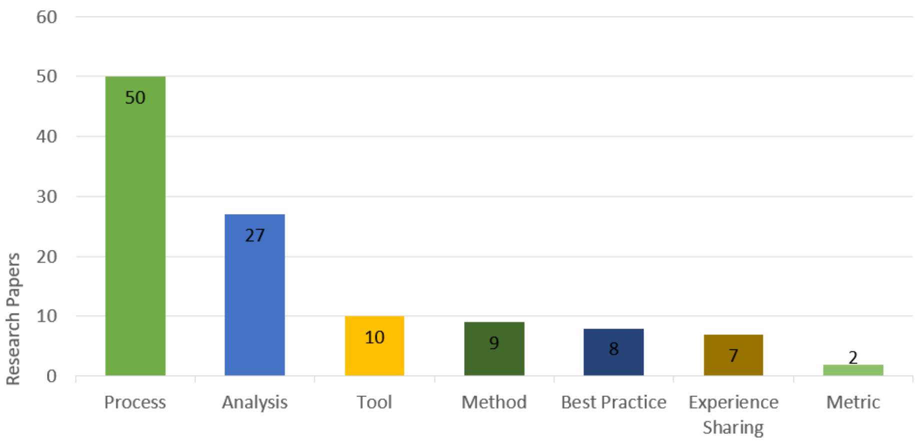

Figure 7. Distribution of the research paper contribution types toward the legacy system migration domain. References in Table 10

Table 10. References for the research types

Contribution classification	ID
Process	P1, P2, P3, P6, P10, P11, P12, P14, P15, P16, P17, P19, P21, P29, P32, P34, P36, P39, P40, P43, P45, P46, P47, P49, P50, P51, P53, P54, P59, P60, P61, P62, P65, P67, P70, P72, P73, P75, P77, P78, P81, P85, P91, P95, P96, P97, P98, P99, P106, P109
Analysis	P4, P7, P8, P18, P20, P26, P27, P28, P35, P37, P41, P42, P44, P48, P56, P57, P58, P64, P68, P71, P74, P79, P80, P82, P84, P90, P100
Tool	P13, P33, P76, P89, P93, P101, P104, P105, P107
Method	P13, P33, P76, P89, P93, P101, P104, P105, P107
Best practice	P9, P22, P25, P31, P69, P92, P100, P103
Experience sharing	P5, P30, P38, P52, P66, P87, P103
Metric	P24, P63

contribution might be a tool, a metric, or software to support migration efforts. Using this categorization method on our primary research documents yielded 90% managerial and 10% technical contributions.

4.4. Key findings

Our main findings are summarized in Table 11. While not all-inclusive, these gaps in the research and observations are worth highlighting. We can observe the publication trends, research approaches, and publication venues, as well as gain a better understanding of the MSA migration process.

Table 11. Observations regarding the research gaps related to migration processes

Source	Observation
RQ1	The year 2018 saw a significant increase in published research writings, mostly journal articles and conference papers. Overall, conference papers and journal articles dominate the publication types.
RQ1	Research into MSA migration has grown significantly between 2015 and 8/2023.
RQ1	Publication venues are scattered across different application domains and distinct topics.
RQ1	The primary research identified is split between empirical (52%) and non-empirical (48%) work.
RQ2	Primary research papers mostly focused on the transformation (48%) phase of the re-engineering process, rather than reverse engineering (30%) or forward engineering (22%)
RQ3	Managerial contributions account for 90% of all contributions, with processes being the most common type (44%).

5. Discussion

The first research question focused on the publication trends related to the primary research. Based on our results, the research on migrating legacy systems to MSA has increased from 2016 onward, as seen from Figure 4. Our primary research found no papers related to MSA migrations from 2015. However, the increasing number of publications in our primary research suggests growth, possibly supported by technological innovations (cloud platforms, containers, and DevOps), shifts in the software architectural landscape, and a move away from monolithic architecture and toward distributed architecture. Related to the growth of

MSA migration research, we observed common motivations for migration in the primary research papers: scalability (P5, P8, P12, P17, P19, P21, P25, P27, P28, P32, P42, P47, P58, P65, P71, P86, P87, P90, P103, P106, P107, P109), maintainability (P1, P2, P8, P18, P24, P25, P28, P32, P38, P42, P51, P65, P66, P77, P86, P103), time to market (P8, P30), adaptability to new technologies (P2, P25, P30), and flexibility (P42, P56, P57, P87, P90, P107). A possible reason for these motivations may be the need for more flexible software architectures in the increasingly digitalized world.

The distribution of research types in our primary research is dominated by validation research (33%), suggesting a strong emphasis on testing and validating new research artifacts in controlled settings. Research in the field of philosophy (23%) often involves the creation of taxonomies and conceptual frameworks, which provides clarity to the field and suggests a mature research area. Evaluation research, which accounts for 19% of the research conducted, serves as a bridge between theory and practice. The significant representation of evaluation research suggests that the field of research is increasingly focused on applying theoretical knowledge to practical situations and discovering the practical benefits and limitations of research techniques. Experience research accounts for only 13% of the data and is based on anecdotal evidence, lacking in testing or validation. Nevertheless, it still provides valuable real-world information. The presence of experience research indicates that practical insights and lessons learned from practitioners are still highly valued in the field. Solution proposals account for only 12% of papers in the field, focusing on validating rather than proposing new solutions.

Related to the research type, we found that a majority of it is empirical, indicating that the research field is developing and becoming more mature. This suggests that researchers are gaining a better understanding of the challenges, benefits, and nuances of migrating to MSA by validating their theories and strategies with real-world data. The abundance of validation and evaluation research also indicates that researchers are willing to test and improve the existing theories and strategies related to the migration process. Additionally, practitioners can benefit from the wealth of tried and tested tools, processes, and techniques available to them. Meanwhile, researchers have the opportunity to develop new methods by building on previous successful ones or experimenting with untested approaches.

Our primary research publications are mostly journal articles or conference proceedings. Similar results are reported in the related research by Di Francesco et al. [24] and Hassan et al. [20]. The tilt toward journals and conferences is understandable as they are more scientifically rewarding than workshop publications. Targeting more challenging publication venues is a good sign as that indicates that the research is of higher quality. We also observed fragmentation in the publication venues, as was also noted by Di Francesco et al. [24]; their study did not directly relate to ours as it focused on architecting with MSA rather than migrating to MSA; however, there is a clear parallel between their findings and ours regarding the publication venues. The fragmentation of publication venues suggests that researchers approach the MSA migration process from multiple disciplines with differing concerns. The fragmentation can also make it difficult for researchers and practitioners to navigate the literature on MSA migrations. This can make it challenging to identify influential research, gather knowledge, and ensure comprehensive coverage when conducting literature reviews. Furthermore, the fragmentation can cause redundancy as multiple researchers may work on similar research in isolation. For practitioners, this can cause obstacles when trying to access the latest best practices available, which can delay the introduction of new ones.

The second research question queried the phase of the re-engineering process that the research papers discussed. It was found that the most commonly discussed re-engineering phase was transformation, accounting for 48% of the papers, which was more popular than reverse engineering (30%) and forward engineering (22%). The difference between the transformation phase and the other re-engineering phases indicates that researchers are most interested in studying tools, workflows, and processes that transition the legacy architecture to a more modern form. This might imply that while reverse engineering and forward engineering are essential, the transformation phase is the most difficult of the three or that there are more well-established practices in the other two. From the primary research, we observed the following challenges related to the re-engineering process: the absence of suitable decomposition approaches (P28), the high level of coupling between software components (P23, P26, P27, P30, P35, P84), the lack of guidelines and best practices for migration (P7, P27, P30, P50, P67), and the identification of microservices from existing systems and boundary recognition (P3, P11, P13, P16, P17, P27, P28, P30, P32, P33, P34, P36, P39). However, many solutions are also proposed to identify microservices (P53, P55, P59, P61, P78).

The third research question aimed to identify the types of contributions made in each research paper. The results showed that the most popular contribution type was process (44%), which indicates that migration is a challenging task that requires precise and straightforward processes to guide practitioners. The analysis research type accounted for 24% of research, indicating interest in understanding migration's issues and benefits. Out of all the research articles, 9% describe tools. This indicates that the field is gradually moving toward creating software that can help with the migration process. However, further research into tooling related to the migration process toward MSA could benefit the field. Only 8% of the research articles were related to the method. This suggests less focus on refining and introducing new techniques to address specific challenges. Best practices (7%) are crucial for organizations migrating to MSA. However, their relatively low percentage suggests that the field is still consolidating these practices. As more organizations migrate, consolidating and documenting best practices will become increasingly important. Experience sharing is only responsible for 6% of contributions. Experience sharing provides valuable lessons for practitioners, given the unique challenges that each migration can present. The field could benefit from more experience-sharing contributions. Metrics account for only 2% of the contribution types. Their low percentage implies that standard metrics are yet to be developed. As the field grows, there may be an increasing demand for standardized metrics to assess the MSA migrations.

Further analysis shows that 90% of the research contributions are managerial instead of technical; Hassan et al. report similar results regarding MSA migration literature [20]. The significant difference between managerial and technical contributions might be that managerial contributions are more relevant in the migration process.

Other trends we noticed were the limitations of MSA, particularly that MSA is not a silver bullet solution for legacy migrations (P4, P5, P7, P25, P31). P4 and P7 analyzed different migration methods and concluded that there are many different methods for migration, and practitioners must choose the right one depending on their circumstances. The authors of P5 report on their experience working with an MSA migration project; they conclude that microservices are not a one-size-fits-all solution as they introduce new complexities into systems, and many factors, such as distribution complexities, should be considered before adopting this style. P31 argues that there is no single way to implement an MSA into an existing system but that practitioners should know the common pitfalls of such processes. Additionally, it is noted in P35 that there is a lack of evidence for the benefits of mixing different migration methods.

Challenges worth researching are organizational challenges, such as the mindsets of developers during the migration process (P27), the skill sets of developers (P8, P28), and convincing management of the importance of migration (P30). Decentralizing databases is another challenge worth investigating further (P35, P26, P27, P42, P49, P68, P87, P103). In addition, our research suggests that the transactions between microservices are a challenge for practitioners to deal with, so research defining proper guidelines on migrating without performance degradation is critical (P6, P27, P38, P32).

The future of migration toward MSA can be seen through various automated tools, frameworks, and methods for migration, identification, or refactoring. These tools are highlighted in the following research papers: P75, P76, P78, P86, P88, P89, P90, P91, P95, P96, P99, P100, P101, and P108. Advanced techniques and innovations, such as reinforcement learning, are introduced in papers P81, P94, P101, and P104 to support the migration process.

Our research has identified areas with significant challenges or a lack of research, and we recommend conducting further research in these fields. Evaluation research is a significant part of primary research but could still be expanded. This is because practitioners find it challenging to adopt scientific implementations without evidence that they work in practice. There may be a lack of opportunities for evaluation research, as modernizing software from old legacy systems to MSA is still relatively rare. Additionally, companies may hesitate to invite researchers to join or find it hard to enter these large projects that continue for many years. We suggest exploring experience research and solution proposals to provide more real-world insights and challenges and address unresolved issues with innovative solutions.

Regarding the fragmentation of publication venues, we suggest several potential solutions to address this issue. These include conducting more literature reviews that provide an overview of the research field, creating centralized repositories to gather literature and increase accessibility, organizing interdisciplinary workshops and conferences to bring together researchers from different disciplines working with MSA migrations, implementing unified standards for all publication venues, promoting open access publishing to increase availability, and educating new researchers about the research field.

Related to re-engineering, much of the research is focused on the transformation phase. We suggest conducting more research on reverse and forward engineering to understand the whole migration process better. Additionally, it would be useful to investigate the challenges of re-engineering, such as the absence of suitable decomposition strategies and the handling of high coupling in software components.

It is essential to encourage more research toward developing tools and methods that can support the process of migrating. The documentation and consolidation of the best practices for MSA migration should be promoted, and experts should share experiences to capture real-world insights. Developing standardized metrics for evaluating MSA migrations is also crucial.

Promoting research that provides guidelines for choosing the appropriate migration method based on specific circumstances is important. Additionally, organizational challenges, such as those related to the developer mindset and skill set, as well as the role of management in MSA migration, should be investigated. Research on decentralizing databases and managing transactions between microservices can also be promoted. The development of automated tools, frameworks, and methods should be encouraged to simplify the MSA migration process. Furthermore, promoting research on advanced techniques, such as reinforcement learning, can help to support the MSA migration process.

The challenges and motivations we observed from our primary research have inspired us to do more research in this field. In particular, we want to focus on the motivations and challenges related to MSA migrations. We will conduct empirical research on this topic using survey research and interviews.

6. Threats to validity

The threats to validity are classified according to the classification by Wohlin et al. [33]. They give four categories for threats to validity: conclusion, internal, external, and construct. The threats that we have identified are classified as internal and external. Internal threats are those that can affect the study results without the knowledge of the researcher. External threats limit the applicability of the results to the real world.

Regarding the internal validity threats, we implemented inclusion and exclusion criteria to enhance the exactness of the primary research further. As part of our criteria, we restricted the language to English only, which excluded 67 potential research papers from the study. Furthermore, we did not include any grey literature in our study. We do not believe that the lack of grey literature impacts the validity of our research, as peer-reviewed papers must go through strict quality gates, which improves the quality of the research papers included in this study. Another possible threat to validity is bias in selecting research papers. One researcher chose the research papers manually, which may have introduced bias in the selection process. The potential for bias was mitigated by strictly following the exclusion and inclusion criteria.

In terms of the external validity threats, the research in this paper is limited to the research discussing the migration of legacy systems to MSA. The most critical external threat to the validity of this study is that we did not do backward-forward snowballing to gather more potential primary research papers. The potentially missed primary research papers mean our study may not entirely represent the MSA migration research field. We utilized Google Scholar to search for research articles. The results are limited by publication policies, which may affect the accessibility and visibility of the results. Further limitations of Google Scholar are the inability to create complex search strings using Boolean operators and nesting. It is also only possible to search based on the title or full text; it cannot define proximity to the searched words; it cannot use complex dates, only date ranges. In addition, the subject area is broad because there are no predefined sections for each subject matter.

7. Conclusion

Interest in MSA has seen an increase during the past decade. The interest is fuelled by technological advancements, such as cloud computing, automation (DevOps), and containerization, as well as by the overall trend of digitalization and the way software is consumed through the internet via browsers and mobile devices. Therefore, in this study, we have studied the literature related to migrating from legacy systems to MSA.

We identified a pool of 1308 research publications and narrowed it down to 109 primary sources discussing migrating from a legacy system to an MSA-based system. Our first research question focused on the primary research and, more specifically, looked into the research strategy, publication year, and publication venue. The second research question covered the specific area of research related to migrating from legacy systems to MSA. Lastly, the third research question investigated the contributions of the research papers.

From the primary document analysis, we identified the following observations. Legacy system to MSA migration research has increased from 2016 and reached a stable level from 2018 onward. The amount of evaluation research suggests that the research field is maturing. Another indication is that researchers in this field mostly target challenging publication venues (conferences and journals). Related to the publication venues, we also observed that there is major fragmentation in the publication forums, which suggests that researchers approach the MSA migration process from multiple disciplines with differing goals. This can make the research harder to find as it is scattered across many different publication forums. Regarding the focus of the research (estimated with the horseshoe model), we found that more studies focused on the transformation phase compared to reverse engineering or forward engineering. Finally, the most common research contribution type was a process.

The other trends identified from the literature include migration motivations: scalability, maintainability, time to market, and adaptability to new technologies. We also note the challenges observed in the research: that MSA is not a silver bullet solution for legacy migrations, the decomposition of existing systems and identifying microservice candidates from existing legacy systems, organizational challenges, decentralizing databases, the migration of databases, and performance degradation during migration.

The MSA migration research is mature based on the publication venues and research types utilized. The research field is scattered across many publication venues. There are notable technical, managerial, and organizational challenges and differing motivations. We have included our recommendations for future research in the discussion section. We will continue conducting research in this field with survey and interview studies to study industry practitioner's challenges and motivations to understand the current problems and pitfalls that affect the migration processes and the reverse engineering and re-engineering tasks required.

Information about funding/support sources

We received no funding for this research.

References

[1] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering," in *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2008, pp. 1–10.
[2] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables DevOps: Migration to a cloud-native architecture," *IEEE Software*, Vol. 33, No. 3, 2016, pp. 42–52.
[3] M. Garriga, "Towards a taxonomy of microservices architectures," in *Software Engineering and Formal Methods. SEFM 2017*, Lecture Notes in Computer Science, A. Cerone and M. Roveri, Eds., Vol. 10729. Springer, 2018, pp. 203–218.
[4] M. Richards, *Microservices vs. service-oriented architecture*. O'Reilly Media, 2016.
[5] X. Larrucea, I. Santamaria, R. Colomo-Palacios, and C. Ebert, "Microservices," *IEEE Software*, Vol. 35, No. 3, 2018, pp. 96–100.
[6] N. Dragoni, S. Dustdar, S.T. Larsen, and M. Mazzara, "Microservices: Migration of a mission critical system," *arXiv preprint arXiv:1704.04173*, 2017.

[7] P. Jamshidi, C. Pahl, N.C. Mendonça, J. Lewis, and S. Tilkov, "Microservices: The journey so far and challenges ahead," *IEEE Software*, Vol. 35, No. 3, 2018, pp. 24–35.
[8] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Migrating to cloud-native architectures using microservices: An experience report," in *European Conference on Service-Oriented and Cloud Computing*. Springer, 2015, pp. 201–215.
[9] M. Fowler, "Microservices guide," martinfowler.com, Tech. Rep., 2014.
[10] P. Calcado, "Building products at SoundCloud – Part II: Breaking the monolith," SoundCloud, Tech. Rep., 2014. [Online]. https://developers.soundcloud.com/blog/building-products-at-soundcloud-part-2-breaking-the-monolith
[11] C. Bampis, C. Chen, A. Moorthy, and Z. Li, "Netflix video quality at scale with cosmos microservices." Medium, Tech. Rep., 2021. [Online]. https://netflixtechblog.com/netflix-video-quality-at-scale-with-cosmos-microservices-552be631c113
[12] E. Haddad, "Service-oriented architecture: Scaling the Uber engineering codebase as we grow." Uber, Tech. Rep., 2015. [Online]. https://eng.uber.com/service-oriented-architecture/
[13] N. Alshuqayran, N. Ali, and R. Evans, "A systematic mapping study in microservice architecture." in *9th International Conference on Service-Oriented Computing and Applications (SOCA)*, Vol. 9. IEEE, 2016, pp. 44–51.
[14] D.S. Linthicum, "Practical use of microservices in moving workloads to the cloud," *IEEE Cloud Computing*, Vol. 3, No. 5, 2016, pp. 6–9.
[15] W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," in *International Conference on Software Architecture Workshops (ICSAW)*. IEEE, 2017.
[16] A. Carrasco, B.V. Bladel, and S. Demeyer, "Migrating towards microservices: Migration and architecture smells," in *Proceedings of the 2nd International Workshop on Refactoring*. ACM, 2018, pp. 1–6.
[17] H. Knoche and W. Hasselbring, "Drivers and barriers for microservice adoption – A survey among professionals in Germany. Enterprise modelling and information systems architectures," *IEEE Software*, Vol. 14, 2019, pp. 1–35.
[18] V. Velepucha, P. Flores, J. Torres, M. Botto-Tobar, J. León-Acurio et al., "Migration of monolithic applications towards microservices under the vision of the information hiding principle: A systematic mapping study," in *Advances in Emerging Trends and Technologies*. Springer, 2020, pp. 90–100.
[19] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, "Requirements engineering paper classification and evaluation criteria: A proposal and a discussion," *Requirements engineering*, Vol. 11, No. 1, 2006, pp. 102–107.
[20] S. Hassan, R. Bahsoon, and R. Kazman, "Microservice transition and its granularity problem: A systematic mapping study," *Software: Practice and Experience*, Vol. 50, 2020, pp. 1651–1681.
[21] H. Knoche and W. Hasselbring, "From monolithic systems to microservices: An assessment framework," *Information and Software Technology*, Vol. 137, 2021.
[22] A. Razzaq and S.A.K. Ghayyur, "A systematic mapping study: The new age of software architecture from monolithic to microservice architecture – awareness and challenges," *Computer Applications in Engineering Education*, Vol. 31, No. 2, 2023, pp. 421–451.
[23] C. Pahl and P. Jamshidi, "Microservices: A systematic mapping study," in *6th International Conference on Cloud Computing and Services Science*, Vol. 1. ACM, 2016, pp. 137–146.
[24] P. Di Francesco, P. Lago, and I. Malavolta, "Architecting with microservices: A systematic mapping study," *Journal of Systems and Software*, Vol. 150, 2019, pp. 77–97.
[25] M. Waseem, P. Liang, and M. Shahin, "A systematic mapping study on microservices architecture in DevOps," *Journal of Systems and Software*, Vol. 170, 2020.
[26] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE, Tech. Rep., 2007.
[27] E. Vanhala, J. Kasurinen, A. Knutas, and A. Herala, "The application domains of systematic mapping studies: A mapping study of the first decade of practice with the method," *IEEE Access*, Vol. 10, 2022, pp. 37 924–37 937.

[28] J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton et al., "Evidence relating to object-oriented software design: A survey," in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. IEEE, 2007, pp. 482–484.

[29] R. Kazman, S. Woods, and C. S.J, "Requirements for integrating software architecture and reengineering models: CORUM II," in *Proceedings Fifth Working Conference on Reverse Engineering*. IEEE, 1998.

[30] M. Bangert, *Research Guides: Google Scholar: Advanced Searching*. [Online]. https://semo.libguides.com/google-scholar/advanced-searching

[31] S.H.S. University, *What is Google Scholar and how do I use it?*, 2022. [Online]. https://www.shsu.edu/research/guides/tutorials/googlescholar/index.html

[32] E. de Vargas Agilar, R.B. de Almeida, and E.D. Canedo, "A systematic mapping study on legacy system modernization," *SEKE*, 2016, pp. 345–350.

[33] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell et al., *Experimentation in Software Engineering*. Springer. [Online]. http://link.springer.com/10.1007/978-3-642-29044-2

Appendix A. Primary studies

Table A1. Primary research papers

ID	Title	Author	Year
P1	Using Microservices for Legacy Software Modernization	Holger Knoche, Wilhelm Hasselbring	2018
P2	On the Modernization of ExplorViz towards a Microservice Architecture	Christian Zirkelbach, Alexander Krause, Wilhelm Hasselbring	2018
P3	Towards the Understanding and Evolution of Monolithic Applications as Microservices	Daniel Escobar, Diana Cárdenas, Rolando Amarillo, Eddie Castro, Kelly Garcés, Carlos Parra, Rubby Casallas	2016
P4	Analysis of Legacy Monolithic Software Decomposition into Microservices	Justas Kazanavičius, Dalius Mazeika	2020
P5	Migrating to cloud-native architectures using microservices: An experience report	Armin Balalaie, Abbas Heydarnoori, Pooyan Jamshidi, Antonio Celesti, Philipp Leitner	2016
P6	Sustaining Runtime Performance while Incrementally Modernizing Transactional Monolithic Software towards Microservices	Holger Knoche	2016
P7	Migrating Legacy Software to Microservices Architecture	Justas Kazanavičius, Dalius Mažeika	2019
P8	Drivers and Barriers for Microservice Adoption – A Survey among Professionals in Germany	Holger Knoche, Wilhelm Hasselbring	2019
P9	Migrating enterprise legacy source code to microservices: On multitenancy, statefulness, and data consistency	Andrei Furda, Colin Fidge, Olaf Zimmermann, Wayne Kelly, Alistair Barros	2018
P10	Microservices	Xabier Larrucea, Izaskun Santamaria, Ricardo Colomo-Palacios, Christof Ebert	2018
P11	Microservice Decomposition via Static and Dynamic Analysis of the Monolith	Alexander Krause, Christian Zirkelbach, Wilhelm Hasselbring, Stephan Lenga, Dan Kröger	2020
P12	Principles of the Newdimensions Software Creation for a Control Centre of the Future: Cloud Computing and Software Architecture	Rúben Araújo, Joaquim Nunes, Afonso Fernandes, Rolando Martins	2020
P13	Extracting Candidates of Microservices from Monolithic Application Code	Manabu Kamimura, Keisuke Yano, Tomomi Hatano, Akihiko Matsuo	2018
P14	Microservices migration patterns	Armin Balalaie, Abbas Heydarnoori, Pooyan Jamshidi, Damian A. Tamburri, Theo Lynn	2018
P15	Migrating to Microservices	Alexis Henry, Youssef Ridene, Antonio Bucchiarone, Nicola Dragoni, Schahram Dustdar, Patricia Lago, Manuel Mazzara, Victor Rivera, Andrey Sadovykh	2019
P16	Translating a Legacy Stack to Microservices Using a Modernization Facade with Performance Optimization for Container Deployments	Prabal Mahanta, Suchin Chouta, Christophe Debruyne, Hervé Panetto, Wided Guédria, Peter Bollen, Ioana Ciuciu, George Karabatis, Robert Meersman	2020
P17	From Monolith to Cloud Architecture Using Semi-automated Microservices Modernization	Salvatore Augusto Maisto, Beniamino Di Martino, Stefania Nacchia, Leonard Barolli, Peter Hellinckx, Juggapong Natwichai	2019

Table A1 continued

ID	Title	Author	Year
P18	Using Microservices and Software Product Line Engineering to Support Reuse of Evolving Multi-tenant SaaS	Leonardo P. Tizzei, Marcelo Nery, Vinícius C.V.B. Segura, Renato F.G. Cerqueira	2017
P19	Cracking the Monolith: Challenges in Data managementing to Cloud Native Architectures	Mishra Mayank, Kunde Shruti, Nambiar Manoj	2018
P20	Does Migrate a Monolithic System to Microservices Decreases the Technical Debt?	Valentina Lenarduzzi, Francesco Lomio, Nyyti Saarimäki, Davide Taibi	2020
P21	Microservices: Migration of a Mission Critical System	Manuel Mazzara, Nicola Dragoni, Antonio Bucchiarone, Alberto Giaretta, Stephan T. Larsen, Schahram Dustdar	2018
P22	Migrating from Monoliths to Cloud-Based Microservices: A Banking Industry Example	Alan Megargel, Venky Shankararaman, David K. Walker, Muthu Ramachandran, Zaigham Mahmood	2020
P23	Function-Splitting Heuristics for Discovery of Microservices in Enterprise Systems	Adambarage Anuruddha Chathuranga De Alwis, Alistair Barros, Artem Polyvyanyy, Colin Fidge, Claus Pahl, Maja Vukovic, Jianwei Yin, Qi Yu	2018
P24	A Decoupled Health Software Architecture using Microservices and OpenEHR Archetypes	Marcio Silva, André Araújo, Paulo Caetano da Silva	2020
P25	From Monolith to Microservices: A Classification of Refactoring Approaches	Jonas Fritzsch, Justus Bogner, Alfred Zimmermann, Stefan Wagner	2019
P26	Migrating Towards Microservice Architectures: An Industrial Survey	Paolo Di Francesco, Patricia Lago, Ivano Malavolta	2018
P27	Strategies Reported in the Literature to Migrate to Microservices Based Architecture	Heleno Cardoso da Silva Filho, Glauco de Figueiredo Carneiro, Shahram Latifi	2019
P28	Microservices Migration in Industry: Intentions, Strategies, and Challenges	Jonas Fritzsch, Justus Bogner, Stefan Wagner, Alfred Zimmermann	2019
P29	A Model-Driven Approach Towards Automatic Migration to Microservices	Antonio Bucchiarone, Kemal Soysal, Claudio Guidi, Jean-Michel Bruel, Manuel Mazzara, Bertrand Meyer	2019
P30	An Experience Report on the Adoption of Microservices in Three Brazilian Government Institutions	Welder Luz, Everton Agilar, Marcos César de Oliveira, Carlos Eduardo R. de Melo, Gustavo Pinto, Rodrigo Bonifácio	2018
P31	Migrating towards microservices: Migration and architecture smells	Andrés Carrasco, Brent van Bladel, Serge Demeyer	2018
P32	Discovering Microservices in Enterprise Systems Using a Business Object Containment Heuristic	Adambarage Anuruddha Chathuranga De Alwis, Alistair Barros, Colin Fidge, Artem Polyvyanyy, Hervé Panetto, Christophe Debruyne, Henderik A. Proper, Claudio Agostino Ardagna, Dumitru Roman, Robert Meersman	2018
P33	Migrating Web Applications from Monolithic Structure to Microservices Architecture	Zhongshan Ren, Wei Wang, Guoquan Wu, Chushu Gao, Wei Chen, Jun Wei, Tao Huang	2018
P34	Towards Identifying Microservice Candidates from Business Rules Implemented in Stored Procedures	Marx Haron Gomes Barbosa, Paulo Henrique M. Maia	2020
P35	Migrating from monolithic architecture to microservices: A Rapid Review	Francisco Ponce, Gastón Márquez, Hernán Astudillo	2019
P36	Towards a Technique for Extracting Microservices from Monolithic Enterprise Systems	Alessandra Levcovitz, Ricardo Terra, Marco Tulio Valente	2016

Table A1 continued

ID	Title	Author	Year
P37	Migration to Microservices: Barriers and Solutions	Javad Ghofrani, Arezoo Bozorgmehr	2019
P38	From a Monolithic Big Data System to a Microservices Event-Driven Architecture	Rodrigo Laigner, Marcos Kalinowski, Pedro Diniz, Leonardo Barros, Carlos Cassino, Melissa Lemos, Darlan Arruda, Sergio Lifschitz, Yongluan Zhou	2020
P39	Automatic Microservices Identification from a Set of Business Processes	Mohamed Daoud, Asmae El Mezouari, Noura Faci, Djamal Benslimane, Zakaria Maamar, Aziz El Fazziki	2020
P40	Modernizing legacy systems with microservices: A roadmap	Daniele Wolfart, Wesley K.G. Assunção, Ivonei F. da Silva, Diogo C.P. Domingos, Ederson Schmeing, Guilherme L. Donin Villaca, Diogo do N. Paza	2021
P41	A Systematic Literature Review on Migration to Microservices: A Quality Attributes perspective	Roberta Capuano, Henry Muccini	2022
P42	Are we speaking the industry language? The practice and literature of modernizing legacy systems with microservices	Thelma Colanzi, Aline Amaral, Wesley Assunção, Arthur Zavadski, Douglas Tanno, Alessandro Garcia, Carlos Lucena	2021
P43	Translating a legacy stack to microservices using a modernization facade with performance optimization for container deployments	Prabal Mahanta, Suchin Chouta	2020
P44	Monoliths to microservices-migration problems and challenges: A SMS	Victor Velepucha, Pamela Flores	2021
P45	SPReaD: service-oriented process for reengineering and DevOps: Developing microservices for a Brazilian state department of taxation	Carlos Eduardo da Silva, Yan de Lima Justino, Eiji Adachi	2022
P46	Migration of monoliths through the synthesis of microservices using combinatorial optimization	Gianluca Filippone, Marco Autili, Fabrizio Rossi, Massimo Tivoli	2021
P47	The Adoption of Microservices Architecture as a Natural Consequence of Legacy System Migration at Police Intelligence Department	Murilo Góes de Almeida, Edna Dias Canedo	2022
P48	Migration of monolithic applications towards microservices under the vision of the information hiding principle: A systematic mapping study	Victor Velepucha, Pamela Flores, Jenny Torres	2019
P49	An Approach to Migrate from Legacy Monolithic Application into Microservice Architecture	Justas Kazanavičius, Dalius Mažeika	2023
P50	A multi-criteria strategy for redesigning legacy features as microservices: An industrial case study	Wesley K.G. Assunção, Thelma Elita Colanzi, Luiz Carvalho, Juliana Alves Pereira, Alessandro Garcia, Maria Julia de Lima, Carlos Lucena	2021
P51	From Monolith to Microservices: A Semi-Automated Approach for Legacy to Modern Architecture Transition using Static Analysis	Mohd Hafeez Osman, Cheikh Saadbouh, Khaironi Yatim Sharif, Novia Admodisastro	2022
P52	Design Patterns and Microservices for Reengineering of Legacy Web Applications	V. Dattatreya, K.V. Chalapati Rao, M. Raghava	2021
P53	Improving microservices extraction using evolutionary search	Khaled Sellami, Ali Ouni, Mohamed Aymen Saied, Salah Bouktif, Mohamed Wiem Mkaouer	2022

Table A1 continued

ID	Title	Author	Year
P54	The Quality-Driven Refactoring Approach in BIM Italia	Roberta Capuano, Fabio Vaccaro	2023
P55	Applying Microservice Refactoring to Object-oriented Legacy System	Junfeng Zhao, Ke Zhao	2021
P56	An empirical study of the systemic and technical migration towards microservices	Hamdy Michael Ayas, Philipp Leitner, Regina Hebig	2023
P57	Towards a Multi-Tenant Microservice Architecture: An Industrial Experience	Cesar Batista, Bruno Proença, Everton Cavalcante, Thais Batista, Felipe Morais, Henrique Medeiros	2022
P58	Review of methods for migrating software systems to microservices architecture	Aleksandra Stojkov, Zeljko Stojanov	2021
P59	Analysis of a many-objective optimization approach for identifying microservices from legacy systems	Wesley K.G. Assunção, Thelma Elita Colanzi, Luiz Carvalho, Alessandro Garcia, Juliana Alves Pereira, Maria Julia de Lima, Carlos Lucena	2022
P60	Migrating monoliths to microservices-based customizable multi-tenant cloud-native apps	Sindre Grønstøl Haugeland, Phu H. Nguyen, Hui Song, Franck Chauvel	2021
P61	Leveraging the layered architecture for microservice recovery	Pascal Zaragoza, Abdelhak-Djamel Seriai, Abderrahmane Seriai, Anas Shatnawi, Mustapha Derras	2022
P62	The collaborative modularization and reengineering approach CORAL for open source research software	Christian Zirkelbach, Alexander Krause, Wilhelm Hasselbring	2020
P63	From Monolith to Microservice: Measuring Architecture Maintainability	Muhammad Hafiz Hasan, Mohd. Hafeez Osman, Novia Indriaty Admodisastro, Muhamad Sufri Muhammad	2023
P64	Adopting microservices and DevOps in the cyber-physical systems domain: A rapid review and case study	Jonas Fritzsch, Justus Bogner, Markus Haug, Ana Cristina Franco da Silva, Carolin Rubner, Matthias Saft, Horst Sauer, Stefan Wagner	2023
P65	Microservice migration using strangler fig pattern: A case study on the green button system	Chia-Yu Li, Shang-Pin Ma, Tsung-Wen Lu	2020
P66	From a monolithic big data system to a microservices event-driven architecture	Rodrigo Laigner, Marcos Kalinowski, Pedro Diniz, Leonardo Barros, Carlos Cassino, Melissa Lemos, Darlan Arruda, Sérgio Lifschitz, Yongluan Zhou	2020
P67	Towards a process for migrating legacy systems into microservice architectural style	Daniele Wolfart, Ederson Schmeing, Gustavo Geraldino, Guilherme Villaca, Diogo Paza, Diogo Paganini, Wesley K.G. Assunção, Ivonei F. da Silva, Victor F.A. Santander	2020
P68	Using Database Schemas of Legacy Applications for Microservices Identification: A Mapping Study	Antonios Mparmpoutis, George Kakarontzas	2022
P69	Patterns for Migration of SOA Based Applications to Microservices Architecture.	Vinay Raj, Ravichandra Sadam	2021
P70	A Hot Decomposition Procedure: Operational Monolith System to Microservices	Nikolay Ivanov, Antoniya Tasheva	2021
P71	A systematic mapping study: The new age of software architecture from monolithic to microservice architecture–awareness and challenges	Abdul Razzaq, Shahbaz A.K. Ghayyur	2023

Table A1 continued

ID	Title	Author	Year
P72	From Monolithic to Microservice Architecture: The Case of Extensible and Domain-specific IDEs	Romain Belafia, Pierre Jeanjean, Olivier Barais, Gurvan Le Guernic, Benoit Combemale	2021
P73	On the performance and adoption of search-based microservice identification with tomicroservices	Luiz Carvalho, Alessandro Garcia, Thelma Elita Colanzi, Wesley K.G. Assunção, Juliana Alves Pereira, Baldoino Fonseca, Márcio Ribeiro, Maria Julia de Lima, Carlos Lucena	2020
P74	Transformation of Monolithic Applications towards Microservices	Zaigham Mushtaq, Najia Saher, Faisal Shazad, Sana Iqbal, Anam Qasim, Imran Imran	2022
P75	An Approach to Migrate a Monolith Database into Multi-Model Polyglot Persistence Based on Microservice Architecture: A Case Study for Mainframe ...	Justas Kazanavičius, Dalius Mažeika, Diana Kalibatienė	2022
P76	Code vectorization and sequence of accesses strategies for monolith microservices identification	Vasco Faria, António Rito Silva, Irene Garrigós, Juan Manuel Murillo Rodríguez, Manuel Wimmer	2023
P77	Microservice Decompositon: A Case Study of a Large Industrial Software Migration in the Automotive Industry.	Heimo Stranner, Stefan Strobl, Mario Bernhart, Thomas Grechenig	2020
P78	Expert system for automatic microservices identification using API similarity graph	Xiaoxiao Sun, Salamat Boranbaev, Shicong Han, Huanqiang Wang, Dongjin Yu	2022
P79	Re-engineering Legacy Systems as Microservices: An Industrial Survey of Criteria to Deal with Modularity and Variability of Features	Luiz Carvalho, Alessandro Garcia, Wesley K.G. Assunção, Thelma Elita Colanzi, Rodrigo Bonifácio, Leonardo P. Tizzei, Rafael de Mello, Renato Cerqueira, Márcio Ribeiro, and Carlos Lucena	2022
P80	The migration journey towards microservices	Hamdy Michael Ayas, Philipp Leitner, Regina Hebig	2021
P81	Facilitating the migration to the microservice architecture via model-driven reverse engineering and reinforcement learning	MohammadHadi Dehghani, Shekoufeh Kolahdouz-Rahimi, Massimo Tisi, Dalila Tamzalit	2022
P82	Migration from monolithic to microservice architecture: Research of impacts on agility	Josef Doležal, Alena Buchalcevová	2022
P83	Mono2micro: A practical and effective tool for decomposing monolithic java applications to microservices	Anup K. Kalia, Jin Xiao, Rahul Krishna, Saurabh Sinha, Maja Vukovic, Debasish Banerjee	2021
P84	Accumulation and prioritization of architectural debt in three companies migrating to microservices	Saulo Soares De Toledo, Antonio Martini, Phu H. Nguyen, Dag I.K. Sjøberg	2022
P85	How to transition incrementally to microservice architecture	Karoly Bozan, Kalle Lyytinen, Gregory M. Rose	2020
P86	Materializing Microservice-oriented Architecture from Monolithic Object-oriented Source Code	Pascal Zaragoza, Abdelhak-Djamel Seriai, Abderrahmane Seriai, Anas Shatnawi, Hinde-Lilia Bouziane, Mustapha Derras	2021
P87	Developing a Microservices Integration Layer for Next-Generation Rail Operations Centers	Andrei Furda, Lionel van den Berg, Graeme Reid, Giancarlo Camera, Matteo Pinasco	2022

Table A1 continued

ID	Title	Author	Year
P88	From monolithic to microservice architecture: An automated approach based on graph clustering and combinatorial optimization	Gianluca Filippone, Nadeem Qaisar Mehmood, Marco Autili, Fabrizio Rossi, Massimo Tivoli	2023
P89	Microservice decomposition and evaluation using dependency graph and silhouette coefficient	Ana Santos, Hugo Paula	2021
P90	Decomposition of Monolith Applications Into Microservices Architectures: A Systematic Review	Yalemisew Abgaz, Andrew McCarren, Peter Elger, David Solan, Neil Lapuz, Marin Bivol, Glenn Jackson, Murat Yilmaz, Jim Buckley, Paul Clarke	2023
P91	Migrating Monoliths to Microservices Integrating Robotic Process Automation into the Migration Approach	Burkhard Bamberger, Bastian Körber	2022
P92	Mind Overflow: A Process Proposal for Decomposing Monolithic Applications in Microservices	Tcharles Pereira, Kleinner Farias	2020
P93	Benchmarks and performance metrics for assessing the migration to microservice-based architectures.	Nichlas Bjørndal, Antonio Bucchiarone, Manuel Mazzara, Nicola Dragoni, Schahram Dustdar	2021
P94	CARGO: AI-guided dependency analysis for migrating monolithic applications to microservices architecture	Vikram Nitin, Shubhi Asthana, Baishakhi Ray, Rahul Krishna	2022
P95	From Legacy to Microservices: A Type-based Approach for Microservices Identification using ML and Semantic Analysis	Imen Trabelsi, Manel Abdellatif, Abdalgader Abubaker, Naouel Moha, Sébastien Mosser, Samira Ebrahimi-Kahou, Yann-Gaël Guéhéneuc	2022
P96	A multi-model based microservices identification approach	Mohamed Daoud, Asmae El Mezouari, Noura Faci, Djamal Benslimane, Zakaria Maamar, Aziz El Fazziki	2021
P97	Building a Performance Efficient Core Banking System Based on the Microservices Architecture	Fikri Aydemir, Fatih Başçiftçi	2022
P98	A DDD Approach Towards Automatic Migration To Microservices	Malak Saidi, Anis Tissaoui, Sami Faiz	2023
P99	Automated Planning for Software Architectural Migration	Nacha Chondamrongkul, Jing Sun, Ian Warren	2020
P100	Design principles, architectural smells and refactorings for microservices: A multivocal review	Davide Neri, Jacopo Soldani, Olaf Zimmermann, Antonio Brogi	2020
P101	Improving Industry 4.0 Readiness: Monolith Application Refactoring using Graph Attention Networks	Tanisha Rathod, Christina Terese Joseph, John Paul Martin	2023
P102	Incremental analysis of legacy applications using knowledge graphs for application modernization	Saravanan Krishnan, Alex Mathai, Amith Singhee, Atul Kumar, Shivali Agarwal, Keerthi Narayan Raghunath, David Wenk	2022
P103	Keep it in Sync! Consistency Approaches for Microservices – An Insurance Case Study	Arne Koschel, Andreas Hausotter, Moritz Lange, Sina Gottwald	2020
P104	Microservice remodularisation of monolithic enterprise systems for embedding in industrial IoT networks	Adambarage Anuruddha Chathuranga De Alwis, Alistair Barros, Colin Fidge, Artem Polyvyanyy	2021
P105	An Approach to Break Down a Monolithic App into Microservices	Taimoor Syed, Jun Long, Vijay Khatri, Mansoor Khuhro	2021

Table A1 continued

ID	Title	Author	Year
P106	A Novel Methodology to Restructure Legacy Application onto Micro-Service-Based Architecture System	T.R. Vinay, Ajeet A. Chikkamannur	2022
P107	Impacts of Decomposition Techniques on Performance and Latency of Microservices	Chaitanya K. Rudrabhatla	2020
P108	Remodularization analysis for microservice discovery using syntactic and semantic clustering	Adambarage Anuruddha Chathuranga De Alwis, Alistair Barros, Colin Fidge, Artem Polyvyanyy	2020
P109	Analysis and Development of Microservices Architecture in Loan Application System of Cooperative Enterprise in Indonesia	Reynaldi Lie, Ahmad Nurul Fajar	2022

Appendix B. Publication venues

Table B1. Publication venues of the primary research papers

Publication venue	#	ID
IEEE International Conference on Software Architecture Companion (ICSA-C)	4	P11, P34, P41, P54
IEEE Software	3	P1, P9, P10
IEEE International Conference on Software Architecture (ICSA)	3	P26, P61, P88
Euromicro Conference on Software Engineering and Advanced Applications (SEAA)	3	P38, P60, P66
International Journal of Advanced Computer Science and Applications (IJACSA)	3	P51, P63, P107
Software: Practice and Experience	2	P14, P64
On the Move to Meaningful Internet Systems: OTM Workshops	2	P16, P43
International Journal of Computer Applications (IJCA)	2	P24, P92
Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment (DEVOPS)	2	P25, P29
Brazilian Symposium on Software Components, Architectures, and Reuse (SBCARS)	2	P42, P89
Empirical Software Engineering (EMSE)	2	P56, P59
International Conference on Advanced Information Systems Engineering (CAiSE)	2	P104, P108
Collaborative Workshop on Evolution and Maintenance of Long-Living Software Systems	1	P2
Conferencia Latinoamericana En Informatica (CLEI)	1	P3
Baltic DB&IS Conference Forum and Doctoral Consortium	1	P4
Advances in Service-Oriented and Cloud Computing (ESOCC)	1	P5
Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering (ICPE)	1	P6
Open Conference of Electrical, Electronic and Information Sciences (eStream)	1	P7
Enterprise Modelling and Information Systems Architectures (EMISAJ)	1	P8
Computers in Railways XVII	1	P12
Asia-Pacific Software Engineering Conference (APSEC)	1	P13
Microservices	1	P15
Advances on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)	1	P17
International Systems and Software Product Line Conference (SPLC)	1	P18
European Conference on Software Architecture (ECSA)	1	P19

Table B1 continued

Publication venue	#	ID
Journal of Systems and Software (JSS)	1	P20
IEEE Transactions on Services Computing	1	P21
Software Engineering in the Era of Cloud Computing	1	P22
Service-Oriented Computing (ICSOC)	1	P23
International Conference on Information Technology-New Generations (ITNG)	1	P27
IEEE International Conference on Software Maintenance and Evolution (ICSME)	1	P28
Brazilian Symposium on Software Engineering (SBES)	1	P30
International Workshop on Refactoring (IWoR)	1	P31
On the Move to Meaningful Internet Systems (OTM)	1	P32
Asia-Pacific Symposium on Internetware	1	P33
International Conference of the Chilean Computer Science Society (SCCC)	1	P35
Workshop on Software Visualization, Evolution, and Maintenance (VEM)	1	P36
Applied Informatics (ICAI)	1	P37
Smart Applications and Data Analysis (SADASC)	1	P39
International Conference on Evaluation and Assessment in Software Engineering (EASE)	1	P40
Second International Conference on Information Systems and Software Technologies (ICI2ST)	1	P44
Service Oriented Computing and Applications (SOCA)	1	P45
IEEE International Conference on Software Reliability Engineering Workshops (ISSRE Wksp)	1	P46
International Conference on Computational Science and Its Applications (ICCSA)	1	P47
The International Conference on Advances in Emerging Trends and Technologies (ICAETT)	1	P48
IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)	1	P49
IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)	1	P50
International Conference on Smart Computing and Informatics	1	P52
Information and Software Technology	1	P53
International Conference on Dependable Systems and Their Applications (DSA)	1	P55
IEEE Annual Computers, Software, and Applications Conference (COMPSAC)	1	P57
Journal of Engineering Management and Competitiveness (JEMC)	1	P58
International Journal on Advances in Software (IARIA)	1	P62
International Computer Symposium (ICS)	1	P65
Anais Da Escola Regional De Engenharia De Software (ERES)	1	P67
International Conference on Algorithms, Computing and Systems (ICACS)	1	P68
Journal of Web Engineering (JWE)	1	P69
International Conference Automatics and Informatics (ICAI)	1	P70
Computer Applications in Engineering Education	1	P71
Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)	1	P72
International Conference on Software Maintenance (ICSM)	1	P73
International Journal of Innovations in Science and Technology (IJIST)	1	P74
Applied Sciences	1	P75
International Conference on Web Engineering (ICWE)	1	P76
International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)	1	P77
Expert Systems	1	P78
Handbook of Re-Engineering Software Intensive Systems into Software Product Lines	1	P79
Product-Focused Software Process Improvement (PROFES)	1	P80
Software and Systems Modeling (SoSyM)	1	P81
Digitalization of society, business and management in a pandemic: Interdisciplinary Information Management Talks (IDIMT)	1	P82

Table B1 continued

Publication venue	#	ID
ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)	1	P83
IEEE Access	1	P84
Communications of the ACM (CACM)	1	P85
International Conference on Software Technologies (ICSOFT)	1	P86
IEEE Software	1	P87
IEEE Transactions on Software Engineering (TSE)	1	P90
Journal of Automation, Mobile Robotics and Intelligent Systems (JAMRIS)	1	P91
Journal of Object Technology (JOT)	1	P93
IEEE/ACM International Conference on Automated Software Engineering (ASE)	1	P94
Journal of Software: Evolution and Process	1	P95
Journal of Systems Architecture (JSA)	1	P96
Journal of Grid Computing	1	P97
International Conference on Advanced Systems and Electric Technologies (IC_ASET)	1	P98
IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)	1	P99
Software-Intensive Cyber-Physical Systems (SICS)	1	P100
IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)	1	P101
Joint International Conference on Data Science and Management of Data (CODS-COMAD)	1	P102
International Conference on Advanced Service Computing	1	P103
Sylwan Journal	1	P105
Emerging Research in Computing, Information, Communication and Applications (ERCICA)	1	P106
Journal of Theoretical and Applied Information Technology (JATIT)	1	P109