Received: 14 March 2022 | Revised: 20 September 2022 | Accepted: 4 November 2022

DOI: 10.1002/cae.22586

REVIEW

WILEY

A systematic mapping study: The new age of software architecture from monolithic to microservice architecture—awareness and challenges

Abdul Razzaq[1] | **Shahbaz A. K. Ghayyur**[2]

[1]Ocean Technology and Engineering, Ocean College, Zhejiang University, Zhoushan, Zhejiang, People's Republic of China

[2]Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan

Correspondence
Abdul Razzaq, Ocean Technology and Engineering, Ocean College, Zhejiang University, Zhoushan, Zhejiang 316021, People's Republic of China.
Email: 11934071@zju.edu.cn and abdull.razzaq786@outlook.com

Abstract

Microservice architecture (MSA) is an emerging architectural style as an innovative approach that is growing in quality over time. Microservices are endorsed by various researchers to shatter the issues and limitations encountered by the use of aging approaches for monolithic legacy architecture styles. Nonetheless, there exists no state of the research exhaustive study on migration to MSA from monolithic architecture. MSA has made the need to design the development methods of software and styles of architectures that satisfy these demands. The design of software architecture is the ongoing rise of MSA to address the independent deployment, scalability, and maintenance requirements. In this article, a systematic mapping study (SMS) was performed, including the literature on the migration approaches published between 2010 and 2021 to evaluate the contemporary state of the art and practice of software architecting. This study outlines the awareness and importance of MSA. Seventy-three studies were finally selected in this SMS. The following perspectives of this study are publication trends, main venues, the focus of research, migration approaches, migration challenges, successful factors after migration, and the potential for industrial adoption. We synthesize the data and give a summary of the state of the art. This study also provides future research directions and applications for adopting microservices.

KEYWORDS

architecting approach, microservice architecture, monolithic to microservice, MSA, MSA awareness, software architecture

1 | INTRODUCTION

1.1 | Microservices

Microservices have disadvantages, such as asserting self-contained services' moral responsibility. It satisfies a single business's holistic needs, may be transmitted independently, roughly linked, and it is autonomously aware since it is self-contained [6, 20]. An enterprise system application dedicated to a certain organization is made up of a number of microservices that are able to communicate with one another using a lightweight convention and an application programming interface (API) contract [13]. Because of the manner, in which it may be communicated constantly, microservice architecture (MSA) configuration is preferred over traditional

Monolithic Architecture. Its adaptability is unrivaled. The standard Monolithic Architecture, on the other hand, lacks all of these essential features. Because of the MSA outline's right approach, the more significant part of the ventures tends to lean toward this plan [7]. In the beginning, the engineers presented the idea of administration empowering the assistance of service-oriented architecture (SOA). Afterward, the transformative procedure occurred, and the introduction of the benefits ended up being fit for supporting the quick and straightforward operability of the applications outlined according to necessity [33]. With more studies being conducted in this sector, experts have begun to create free, numerous, and independent administrations to overcome the market's challenges. As a result of these points, there is no disputing element that software architecture has a paramount significance in the programming lifecycle, assisting in the quality of products and vital characteristics [34]. This method urges engineers to guarantee that quality extends to entire fulfillment and that the product frameworks are free of flaws. If the data framework's support and development have to be enhanced, component-based development is a viable and cost-effective approach [34]. In light of the market's requirements for a Component-aided Distributed Architecture, the necessity for an SOA architecture, as well as its procedures, was felt. Besides, an answer was needed to make the business deft and address the difficulties that emerge when specific business requirements are to be taken into consideration. Additionally, a functional and adaptable arrangement was required, this may prove adequate for keeping up with the rapid rate of change that occurs with each passing day [35]. In service-oriented organizations, the small-scale administrations have moved toward becoming a design style that is motivated by benefit arranged registering. [7, 14]. Microservices design builds up the perplexing application alongside the dissemination of the use in lumps or units by creating it [16]. At present, in any framework, the versatility, benefit disclosure, and correspondence among administrations that are being upheld by the microservices engineering being developed stage are two essential areas. At the same time, microservices engineering additionally handles a substantial simultaneousness amid the input stack [7]. Microservices API can be composed in any dialect. At that point, the microservices design would naturally make every one of the dialects perfect for showing the coveted yield. [24]. There's no disputing that the cyclic nature of the always-changing transformational process of styles of correspondence and joining has emerged. SOA is the more established of the two approaches for obvious reasons [35].

This study has been structured as follows. Section 2 discusses the design of the research, Section 3 includes the research approach, and Section 4 concludes the research. We also give the Appendix section which presents all relevant tables of selected studies and the rest of the research steps.

2 | BACKGROUND AND RELATED WORK

2.1 | Background

The migration of system applications is turning into a developing issue with various difficulties. Change that relocation is the strategy of mobility from the use of one operating condition to another working condition with similar characteristics.

The key focus of migration is that demonstrates some unpredictable issues for transformation from monolithic to microservices. Decentralization is improved by migrating from monolithic to MSA [18], supplanting capacity, and self-governance of programming models. Although the researcher is undecided on the precise meaning of MSA, it is showing techniques, and its characteristics [22]. It is mindful of framework relocation to MSA. The parts which are utilized as a part of these fundamental programming applications are comprised of essential squares which can be joined together relying on necessity [1]. Microservice engineering is fevered inferable from the reasons that it can address every one of the worries beginning from the prerequisite of the undertaking to the tasks to be performed by the product of a specific business for which it is planned [32]. Additionally, it can likewise guarantee the obligation regarding singular groups. In this approach to deal with discovering the arrangement, the design, open-source improvement, hierarchical structure, and duty have vertically deteriorated [23]. Each service is free, and correspondence is acknowledged through lightweight systems, for example, hypertext transfer protocol API. Microservices are the different styles of monolithic, which might be written in different languages. Inverse to the microservice worldview, there are points of interest yet additional burdens when utilizing the common monolithic legacy approach.

A growing range of tools is used in the creation, service-oriented applications deployment, and packaging. Along with the programming language itself, middleware, container formats, building tools, and deployment tools are also used. All tools should be configured separately, and the entire chain can only function if all switches are set in accordance with requirements. Additionally, if many services are employed, the system

can only function if levels of freedom were considered during development. The remaining service dependencies must then be connected properly. This issue emerges specifically in the context of microservice-based architectures, where it is intended for numerous services to interact [8].

2.2 | Related work

MSA is an architectural style for software development as a suite of self-deployable services. Microservices are small modules and lightweight mechanism which runs as a unique process and automated deployment [21]. Microservices can be written in different programming languages, and they can use different database technologies. Monolithic architecture is a simple and general way for software applications that can be developed and deployed as a single unit with all required functionalities requirements. In monolithic systems, the large product distribution time is also a concern due to the size of the products. Despite these limitations, the monolithic paradigm has shifted to designing modern cloud applications [25, 28].

From the backdates of the 1970s, the research on the modernization or migration of monolithic systems particularly, Lehman's software evolution laws demonstrated the need to periodically restructure monolithic systems to extend their operating life [27]. However, there is less work of published research in the context of monolithic migration to MSA and awareness of MSA; there is no effort to investigate systematically the general effect of the existing research. A mapping of monolithic migration to MSA and systematic categorization can help us to identify the research path and limitations alongside available and future research.

Microservice has indicated an extraordinary light in the software industry for the last few years. Because of its more effective outcomes, it circumvents the existing software infrastructure, which is a global microservices renovation. It will, at some point, be the best-suited architectural style to its adaptive setting. A wide range of software organizations has newly shifted to MSA. Presently, microservices are getting exceptionally well known with the cloud stage, which is a rising style with regard to application improvement because of its independence, flexibility, adaptability, execution, and responsiveness [2, 15]. In the last few years, software organizations have discovered the microservices approach useful, as it helps community and software associations to grow productivity [7].

Monolithic systems restrict continuous delivery by the slowing and painful deployment of software and its development [15]. Continuous software engineering (CSE), a new method of developing software, allows for the delivery of operationally sound software on a frequent basis—up to multiple times per day. The delivery of commercial-grade software is sped up using this method's complex collection of separate technologies [30]. The key features that establish and encourage the demand for such a technology are scalability, protection, reliability, rapid progress, and network speed [3, 19]. The researchers are working to incorporate new types of software architectural design and software development methods to satisfy the needs of the market [7]. System migration to microservice optimizes software architecture decentralization, replaceability, traceability, and autonomy. Although researchers are not persuaded by any clear concept of microservice, they are modeling techniques and their characteristics [21]. The main drawback of the existing monolithic systems is their lack of scalability when performing a certain service task [6].

By implementing the study project known as WINNER, the author [8] explores the motivations of microservices. Through various endpoints and along the tools used for service publication, the selected MSA separates various implementations and establishes implicit correlations. This research [11] compares the monolithic and microservice designs in terms of scalability and performance on a web application. The application was developed using four separate implementations that included two different implementation technologies (Java vs. C#. NET) with two different architectural styles, including monolith and microservices. This article [26] analyses whether the data selection approach is appropriate for knowledge transfer between buildings. The author developed per-building models and examined how effectively they could be applied to unobserved buildings before proposing a microservice-oriented architecture that would allow for faster system development and better system scalability and evolvability.

A migration process and pattern a list of patterns for migration and rearchitecting have been published [10]. Rearchitecting non-cloud-native systems to convert them to cloud-native microservices-based architectures is made easier by these migration patterns. The different observations in several medium to large-scale industrial initiatives from various fields were used to derive the migratory patterns. A method based on SME that chooses and composes migration patterns to create a migration plan. A software architect might identify potential remodeling of monolithic functionalities with the aid of a heuristic technique [12] to lessen the effort involved in moving to a microservices architecture using the SAGA pattern. This endeavor is necessary due to the

functionality migration and the future inclusion of consistency in functionality behavior, which increases the implementation's complexity.

NSGA-III in opposition to RS. The performance of NSGA-III was objectively studied and compared to identify microservice problems and produce findings more quickly [9]. A solution called Microscaler [36] is designed to help application providers find the services and scale them to meet the service level agreement requirement for microservice-based systems in the service-mesh-enabled settings. A fresh criterion, service power, is proposed to confirm the scaling of required services. MicroMiner is a type-based method for locating microservices in monolithic software systems, which was introduced by the author [4]. MicroMiner is guided by a taxonomy of service types that are anticipated using machine learning classification models. It makes use of the system's source code, which it uses to extract the static relationships between its constituent parts. It also conducts a semantic analysis of the source code to identify the semantic similarities between components to guarantee a single constrained context per microservice.

A method that uses the service graph (SG) [31], the SG representation of SOA-based applications, and the task graph in each node of the SG is provided. We provided algorithms for extracting microservices from a given SOA application, building the SG of an upcoming microservices application, and building the SG of an existing SOA application. After choosing a vehicle management system application, they used our suggested method to extract potential microservices and produce the SG for microservices. The created SG serves as a template for the creation of the new application and facilitates quick and simple migration to microservices. It is possible to make a progressive, straightforward transition to microservices from a code and data perspective by using the AOP- [17] based strategy, as well as with very little effort by modifying a few lines of code. The proposed approach showed encouraging results, demonstrating that it does not add substantial performance overhead, maintains similar prices, and presents a lower coding effort. It was demonstrated that with the proposed approach, it is not necessary to manually update a single line of code or piece of data, the application does not have to be stopped if the developer decides they should have stayed with the monolithic version of the software instead of making the migration.

Microservices are a key component of contemporary software architectures because they enable the addition of new functionalities while maintaining a high level of quality, as well as scalability and maintenance of services. The next generation of network functions is implemented in a virtual machine (5 G service) utilizing conventional architecture [29]. A good strategy for achieving the 5 G SBA is to use microservices within the orchestration systems for containers that are currently available. However, as there is a greater need for communication across services, the intentional decoupling and separation of functionality in microservices might cause deployment problems.

State-of-the-art this investigated work on migration reported in the literature by using different techniques in the context of the MA and MSA. The relationships among migration approaches are classified into two different categories: practical migration and awareness for understanding migration. Some recent studies that identified different approaches for migration from monolithic to MSA are presented [5] and we notice that these studies are not enough to provide the migration execution properly also the need for awareness of migration is required.

3 | DESIGN OF RESEARCH

A systematic mapping investigation has the reason to give by investigated published works, a significant level diagram over research. Following the entrenched rules to structure efficient mapping work the resulting segments exhibit the strategy of our research.

3.1 | Investigation queries

The interrogations of this study are a way to find solutions to points of interest in the addressed space. Converging on research for MSA with regard to migration from monolithic to MSA, we built up the accompanying questions:

RQ1: How many articles in this field of research can be found each year?

RQ2: What are the primary venues for the production and printing of the research?

RQ3: What are the principal publication types in the research area?

Achieved in the detailed systematic study:

The key contributions as an outcome of the study that has been chosen

- Limitations and challenges for migration to MSA.
- The study gives the inspiration to adopt the MSA.
- Success scenarios after migrating from monolithic to MSA.
- The study gives the awareness of MSA.

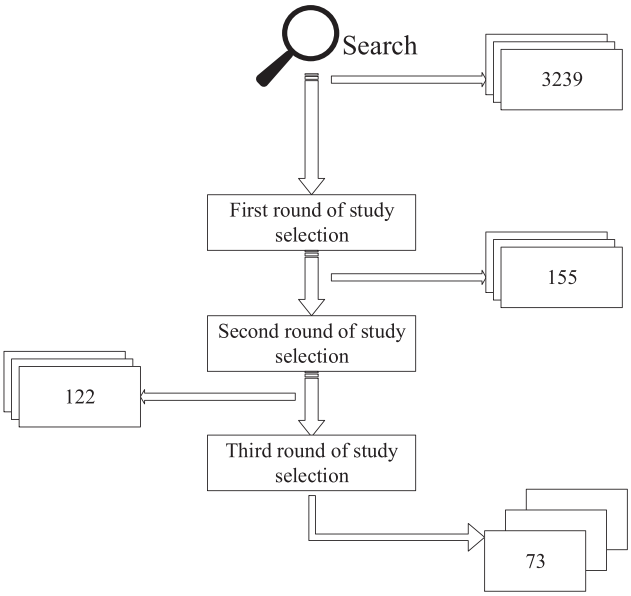

FIGURE 1 Study selection process

3.2 | The quest sequence

In the next step, we composed and described the search string for collecting the existing publications. The following search is:

To combine the keywords in the search string, we used the Boolean operators "OR" and "AND." Furthermore, we use a wildcard to avoid missing relevant publications. Like, when using the mixture *architect* and migrat* (migration/migrating/migrate)*, it may result in these keywords, which are acronyms for architecture, architecting, or architectural.

3.3 | Sources

After making a search string, we identified the five most significant causes of publications.

To conduct the mapping research, we followed the suggestions, which demonstrate how to conduct literature reviews and systematically map studies using evidence-based software engineering (EBSE) methodologies. We want to decrease biases and dangers to research integrity during the key phases of project preparation, data collecting, and reporting findings by using an EBSE technique to undertake the systematic mapping task (see Figure 2).

After adjusting the presence and segregation standards, the traditional was finished by an aggregate of 73 research studies, way of established in Figure 1 and presented the detail of study selection in Figure 3. Figure 3 describes the complete process of study selections and shows the proper report of studies. The process of study selection is started from the "Start" and moves to each step one by one to collect the relevant studies.

3.4 | Data removal

In the final phase, the information from the chosen studies is diagrammed to the three-cataloging compared with the investigation questions. The ruling class consists of data about the quantity of the research study dispersed throughout the years. The subsequent classification extricates the scene data where the selected studies were published. While the third part represents the fundamental themes tended through the journals from the selected conventional. These portions are depicted briefly as cutting-edge in the accompanying areas (Table 1).

After the selection of electronic databases, we applied a search string to search publications (Table 2). We described the process for publication selection using the following steps:

Phase 1. Initial search: Using the search string, the total identified results are 3239 studies for IEEE Explorer, SCOPUS, ACM Digital Library, SpringerLink, and Direct Science, as demonstrated (see Figure 1). The considered reviews are a total of 155. For purposes of suppleness, we implemented the key phrase in all digital archives to name, abstract, and phrases.

Phase 2. Consolidating and expulsion duplication: During this stage, merge the came about productions from the databases, into a single data set. What's more, the copied or comparable inductions are isolated and sorted out by abstract. In this step, came about a sum of 122 productions and the final selected studies are 73.

Phase 3. Application of assortment standards: After evacuating the copied sections (Stage 2), we further sort out the consolidated data set as indicated by a lot of exclusion and inclusion criteria. Our characterized exclusion (E) and inclusion (I) criteria are introduced as follows:

I 1. Publications concentrated on migration from monolithic to MSA.

I 2. Publications are in English language format.

E1. Publications based on abstracts and extended abstracts, books, and editorials.

E2. Publications cannot be accessible as full content.

4 | RESEARCH FOCUS AND IMPACT

4.1 | Publication dissemination (RQ1)

The appropriation of studies during the time is presented (find in Figure 4). In 2016, the number of studies

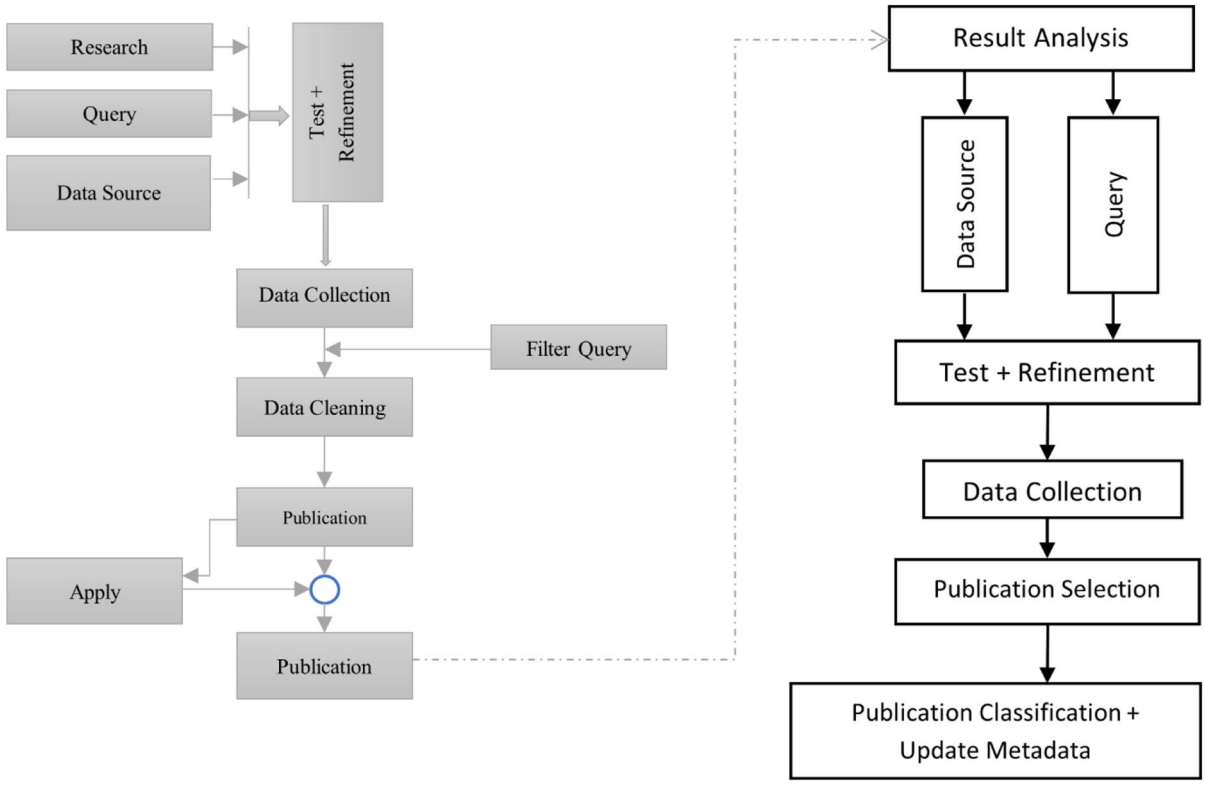

FIGURE 2 Research mapping overview

increased altogether (such as nine publications), although happening in 2018 remains much higher, coming toward an absolute amount of 22 reviews. Now 2019, in hand, 19 studies are available. In 2014, we did not find any relevant articles.

The research topics that we selected were divided into three groups (awareness, migration, and mixed) see Figures 5 and 6 present the yearly studies. We did not find a relevant study in 2014. Several awareness studies have been conducted to provide insight into monolithic, microservices, and migration. Only migration studies were included in the migration section. Studies on awareness and migration are included in the mixed section. We kept track of all the research throughout time to see how frequently they were published.

The distribution of selected studies defines the scientific inclination toward migration approaches from monolithic to MSAs. We have confidence that the total number of publications for the years 2018 and 2019 will be meaningfully more than the published articles in previous years.

4.2 | Publication locales (RQ2)

The crucial five research studies spots on behalf of the measured investigation part can be visualized (see Table 3). We perceive a considerable variation of publication spots for the selected 73 published studies. Many venues indicate that microservices, monolithic architecture, and migration approaches are great research topics, including several concerns. Also, figure out the challenges and mapped accordingly their categories (see Figure 7). Figure 7 is classified into five categories including (migration motivation, MSA motivation, monolithic challenges in MSA studies, success migration factors, and MSA challenges). All classified categories are discussed in tables one by one.

4.3 | Publication types (RQ3)

A total of 73 studies are selected for this study work. In conferences, 55 studies were published while the rest 18 studies were disseminated in Journals. The broad measure of conference papers, related to riposte from RQ1, means that the examination point is quickly developing as an exploration subject, even with a moderately newly created theme.

4.4 | A brief discussion about tables in appendix

Table A1 presents the venues of publication for selected studies, Table A2 presents the distribution

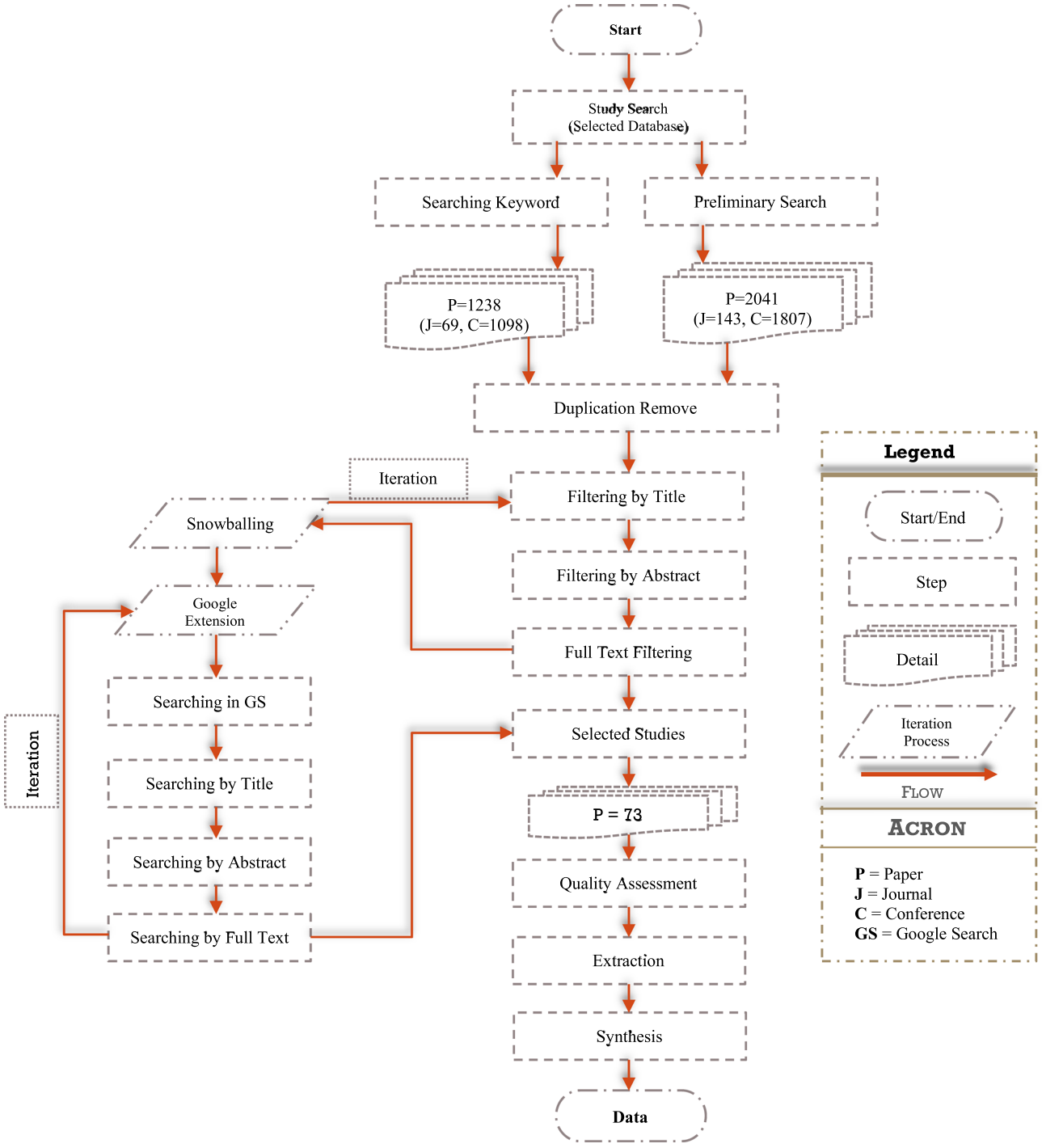

FIGURE 3 The process of the systematic mapping study (SMS) execution in phases and tasks

TABLE 1 Search keywords and search string of this SMS

Searching by query string	Searching by keywords
(((microservices) OR (MSA)) AND ((monolith*) OR (traditional) OR (legacy) OR (migrat*) OR (transform*)))	1. Microservices 2. Microservices architecture challenges 3. Migration monolithic to microservices 4. Migration challenges to microservices

Abbreviation: SMS, systematic mapping study.

TABLE 2 Electronic digital library

No.	Electronic database	Search terms	Web address
EDb-1	IEEE Explore	Paper title, keywords, abstract	https://ieeexplore.ieee.org
EDb-2	Scopus	Paper title, keywords	https://www.scopus.com
EDb-3	ACM Digital Library	Paper title, abstract	https://dl.acm.org
EDb-4	Springer Link	Paper title, keywords	https://link.springer.com
EDb-5	Science Direct	Paper title, keywords	https://www.sciencedirect.com

Note: e-databases are encompassed in the charting study.

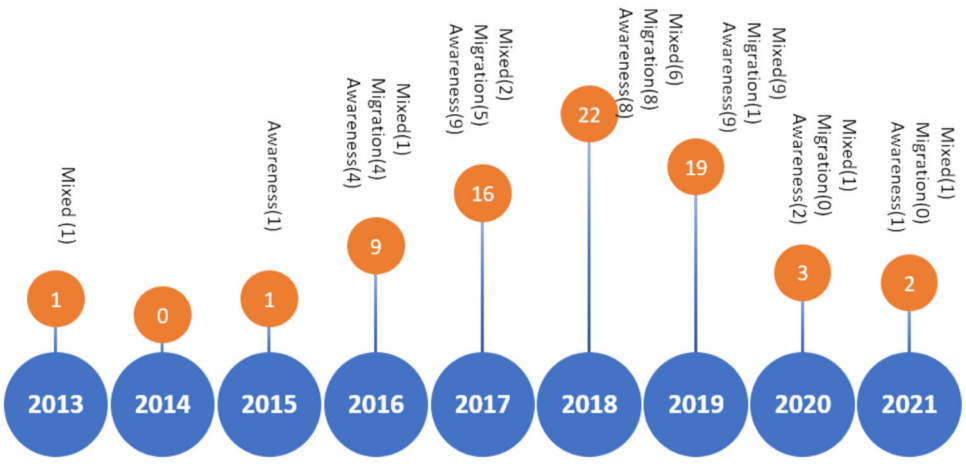

FIGURE 4 The publications' distribution over the years

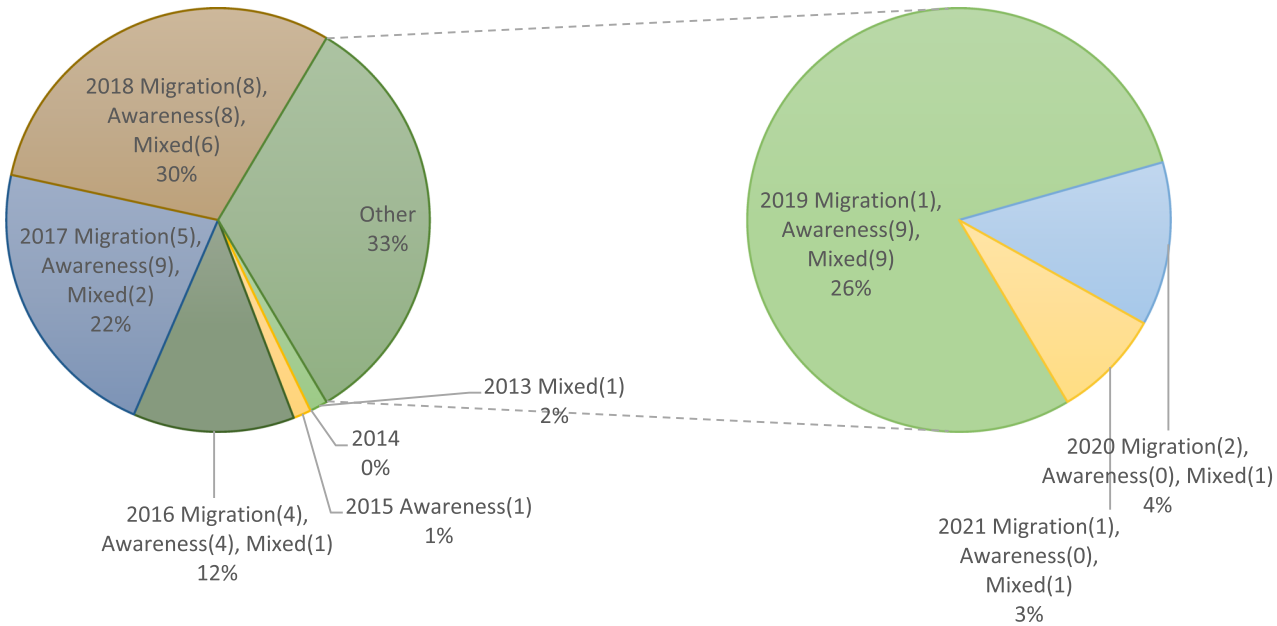

FIGURE 5 A selection of yearly research on migration and awareness

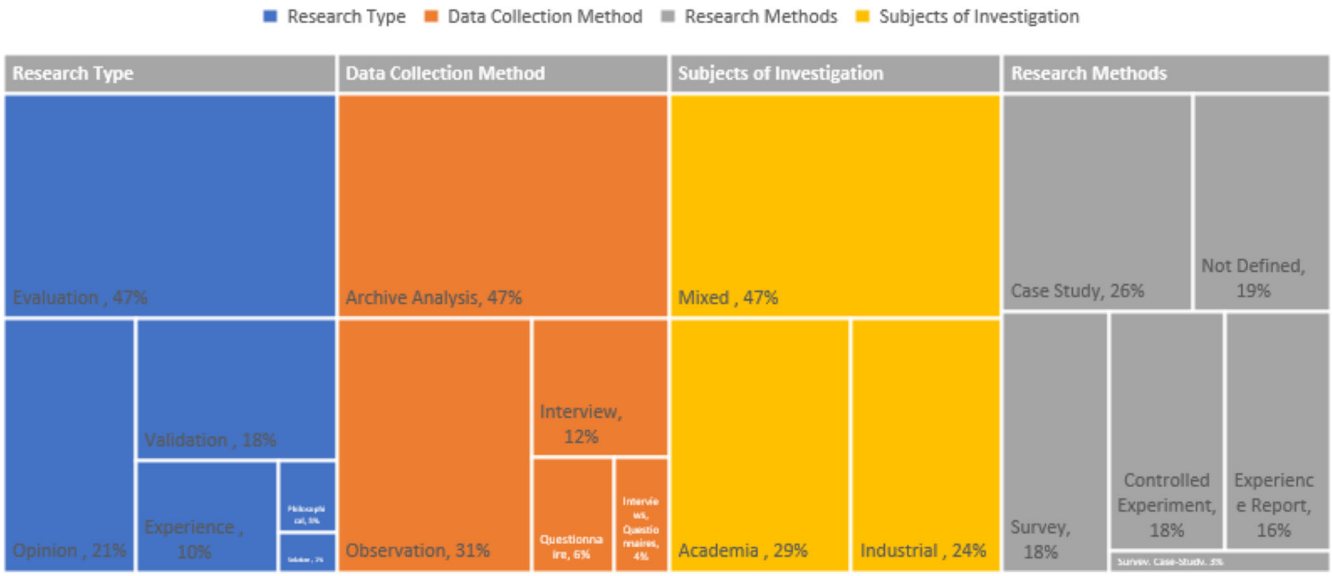

FIGURE 6 Data collection methods

TABLE 3 Maximum used spots in the research part

No.	Publication type	Ellipsis of the publication locations	Number of publications
1	Journal	IEEE Software	8
2	Journal	ICSME	2
3	Conference	ICSA	3
4	Conference	APSEC	2
5	Conference	ICSA-C	2

of these studies over the industries and academia, Table A3 presents the distribution of these studies over migration and awareness, and Table A4 gives the detail of distribution for selected studies over the different types of research categories, Table A5 presents the distribution over the data collection methods, Table A6 shows the list of selected studies for different research strategies, Table A7 presents the distribution of migration approaches over the research methods, and Table A8 discusses each selected paper and gives the brief.

The concept of microservices architectures arose from the industry and was found to be of considerable functional significance. A significant number of companies have opted to implement the MSA within a short period. Companies need to better and analyze their results faster, develop, and launch new products and services than their competitors. To satisfy their customer's needs, they need to be versatile. Migration to the architecture of microservices helps them to do so more effectively.

This study is useful and designed for developers and researchers. The collection and extraction of data are based on academic and industry work. This study helps to understand and gives awareness about MSA and MA. The software architect's role is more important, and architectural decisions must be made based on well understanding and in-depth knowledge of MSA. From this work, the technology industries can decide to migrate from a MA to an MSA or adoption of MSA.

We found a number of studies that appeared to discuss the success factors after migration from monolithic to MSA. We listed the factors in tabular form (see Table 4). we also brief each factor. These identified factors help the software companies to understand its importance and benefits after migrating to MSA. These factors will also motivate those software companies which are thinking of adopting the MSA.

4.5 | Awareness of MSA, SOA, & monolithic architecture

We identified the list of key characteristics of MSA and showed the results difference (see Table 5) to give awareness and to understand the importance of MSA among other architecture. These characteristics (e.g., Maintainability, Evolvability, independent deployment, low costs as a single service, Enhancements, Faster Time to Market) are identified from the selected literature state of the art.

Table 6 reported a list of success factors in selected studies. The successful factors (e.g., Application Size,

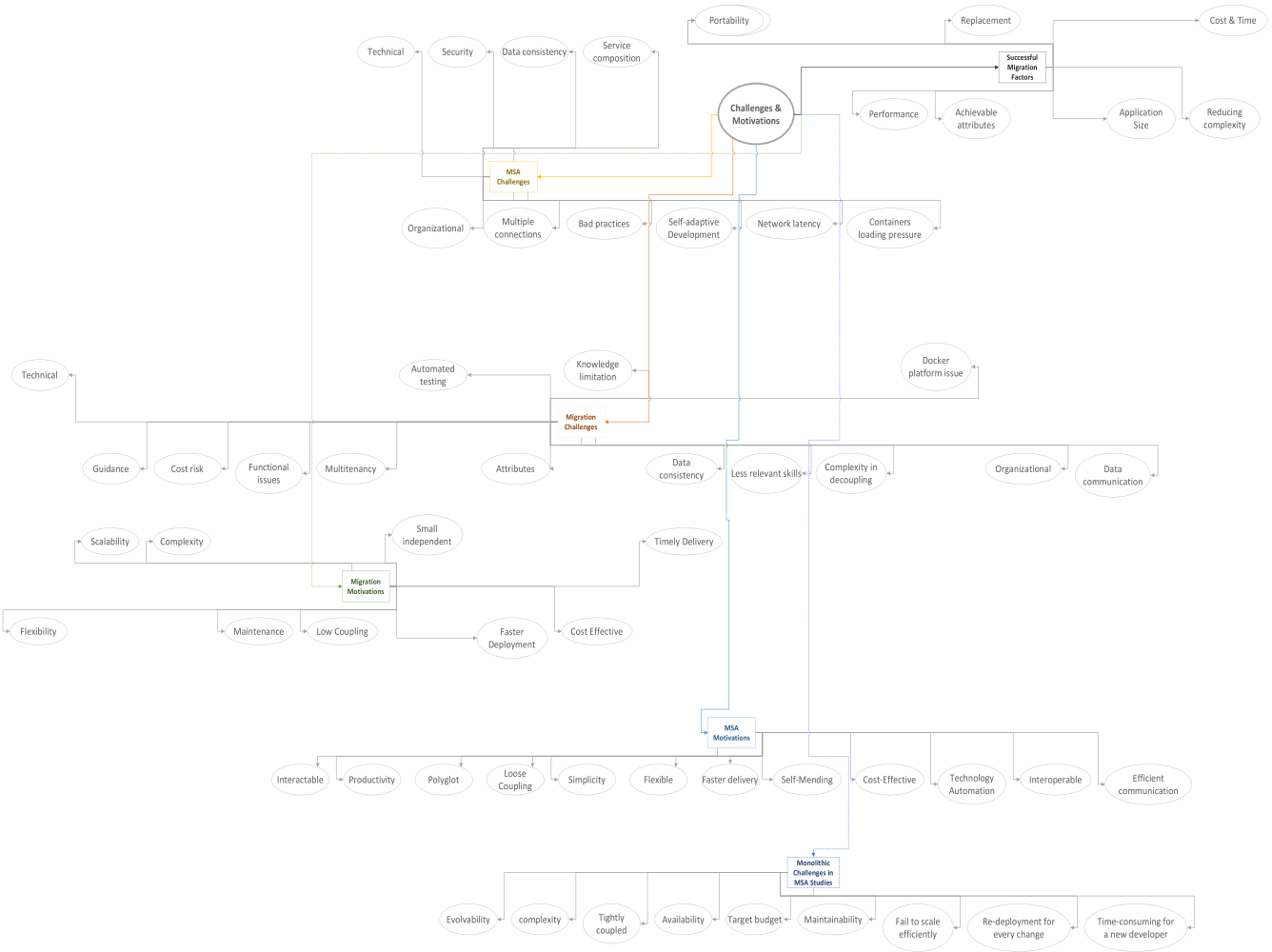

FIGURE 7 Challenges and motivations are classified

Performance, Replacement, Cost, Time, Flexible system, Portability, reducing complexity) can impact the system after migration from a monolithic architecture to MSA. Those factors, however, are not of the same importance or priority for development consideration. These studies [S-11, S-25] tell how they achieved a sized reduction of monolithic legacy applications, were more comfortable maintaining the code, creating smaller ones, and made a flexible system. Based on these factors, monolithic systems can be reused and adopt new technologies. The identified success factors list is also providing motivation to migrate the monolithic systems to MSA.

Migration Successful Factors:

1. *Multiple factors*: The author just discusses the list of factors (Logs, Codebase, Config, Dependencies, Backing Services, Build-Release, and Run, Port Binding, Process, Disposability, Concurrency, Administration Process, Development, and Production Parity). These are achievable factors successfully after migration from monolithic to microservices.
2. *Application size:* Reducing the size of the monolithic is one of the key challenges which be achieved after migration from monolithic to microservices.
3. *Performance & replacement:* Easy replacement can cut costs, and performance can be increased. Production based on microservices can be quickly validated, but also very few bugs are created when replacing another version.
4. *Cost & time:* After migration, cost and time are the most saving factors.
5. *Flexible system:* After successful migration from monolithic to microservices, these factors can be achieved (Maintenance, Deployment, Scalability, Resource Management, Production deployment, Future Development, and Platform hosting).
6. *Achievable attributes:* Successfully achieved the most important factors after migration to microservices, which are (Maintainability improvement, Scalability improvement, Architectural complexity reduction, Simplifies distributed work, Performance improvement,

TABLE 4 Motivations of microservices

ID	Motivation type	Description	Studies
1	Efficient communication	Efficient communication and keeping teams small, (Smaller code, test, deploy, scale), (Polyglot architecture), (Evolutionary design < eliminate, change services, improve>).	S-1
2	Faster delivery	Frequent and more rapid delivery timely. Independently replaceable and upgradeable.	S-3
3	Flexible	Reusability, maintainability, availability, scalability, and computerized sending. Principle points of interest of MSA (depend on innovation heterogeneity, which implies each help in one framework can utilize unexpected innovation in comparison to different administrations to accomplish the ideal objectives and performance). Second, another preferred microservices position is that if one component falls for a system, at that point, it doesn't influence the entire framework. Independently Deployment. Minimize the number of individuals that are taking a shot at a codebase.	S-4
4	Characteristics	MSA style's common characteristics are: • Business capability • Deployment automation • Intellect of endpoints • Language and data with decentralization control	S-5
5	MSA Scope	The scope of microservice is in a business bounded context. Microservices are deployed, scaled, and managed independently with their release lifecycle.	S-7
6	Technology Automation	Microservice has the capability to a small, independent, scalable, and independent in hardware and technology also versioning and automation.	S-9
7	Simplicity	Up-gradation, Fault tolerance, Scalability, Simplified testing, Simplified development, MSA overheads compared to the legacy system (Communication overhead, Orchestration overhead, Decomposition overhead, Integration testing overhead)	S-10
8	Loose Coupling	Microservices are deployed independently and developed in business capabilities.	S-11
9	Cost-Effective	Microservices are flexible and low-cost deployment. Resilience, scalability, and maintainability	S-34
10	Self-Mending	The capacity to change the quantity consequently of service cases and self-mending, i.e., the capacity to, therefore, resume the unsuccessful service, is a major aspect of its local self-adjustment abilities.	S-37
11	Polyglot	Independently scaled using several policies at the business level, which saves the costs of IT infrastructure.	S-43
12	Interoperable	A microservice strategy is considered to enable interoperability from a technical perspective for Business-to-Business (B2B) platforms.	S-51
13	Interactable	Integration and interoperability	S-53
14	Productivity	Scalability, reusability, Technology heterogeneity, resilience, easy & faster deployment, and replaceability.	S-62
		Small services, Enabled the messaging, Developed Autonomously, Context bounded, independently deployable, Developed, decentralized, and released with automated processes.	S-66

Testability, Separation of concerns, Single clear responsibility, and System understandability).

7. *Portability:* One of the major benefits is Portability which can be achieved after migration and can deploy anywhere without any change to source code or container images.
8. *Reducing complexity:* After adopting microservices from monolithic also helps to reduce complexity.

We collected 14 different challenges and limitations from the literature (see Table 7) that presents a detailed description of each challenge and barrier for migrating from MA to MSA. In [S-2], the most highlighted limitation during migration is a "limitation of knowledge" and using harmful practices. The challenges give awareness to practitioners to understand the challenges of microservices.

TABLE 5 Key characteristics of MSA

ID	Characteristics	Monolithic	SOA	MSA	Studies
1	Maintainability and evolvability.			*	S-1, S-14, S-46, S-48, S-55, S-71
2	Related consequences and independent deployment parts.			*	S-1, S-2, S-4, S-5, S-11, S-12, S-17, S-19, S-22, S-36, S-59, S-62
3	Efficient communications and keeping teams small.			*	S-1, S-2
4	Lead to high development and maintenance with low costs as a single service.			*	S-1
5	business-to-business intercommunication		*	*	S-1
6	Simple to develop and smaller code, deploy, and test			*	S-1, S-9, S-12, S-18
7	Allows fast to start and it's easier for new developers			*	S-1, S-72
8	Polyglot technology (each service can use different technology)			*	S-1, S-2, S-4, S-12, S-36, S-43, S-55, S-62
9	Evolutionary in design (maintain, scale, replaceability services).			*	S-1, S-2, S-4, S-5, S-9, S-10, S-12, S-13, S-36, S-43, S-59, S-62
10	Due to some reason, if any service of a component of a system fails, it won't affect the whole system.			*	S-4, S-9, S-13
11	Performance (less load and not more than 100 users)	*			S-4
12	Performance (Heavy Load)			*	S-4
13	Low Cost (Small Projects)	*			S-4
14	Low Cost (Big Projects)			*	S-4
15	New business services instead of new modules			*	S-2, S-11
16	Changes & Enhancements			*	S-12, S-62
17	Faster Time to Market in beginning	*			S-36
18	Faster Time to Market Later			*	S-36
19	Refactoring			*	S-27, S-36
20	ESB communication restriction	*	*		S-3, S-5
21	Automated deployment			*	S-5, S-55
22	Loose coupling inside the service		*	*	S-5
23	Loose coupling outside service			*	S-5
24	Protocols: REST, JSON			*	S-7
25	Decentralized governance			*	S-7
26	Legacy environments	*	*		S-7, S-9, S-18, S-21
27	Portability with different OS			*	S-9, S-55
28	Fault tolerance			*	S-5, S-10
29	Orchestration overhead			*	S-10, S-19, S-32
30	Decomposition			*	S-13, S-14, S-17, S-22
31	Security			*	S-13, S-37, S-41, S-55, S-56
32	Design excellency			*	S-9, S-14
33	Load Balancing		*	*	S-5, S-14
34	Reduce Complexity			*	S-9, S-18, S-36, S-62

TABLE 5 (Continued)

ID	Characteristics	Monolithic	SOA	MSA	Studies
35	Multitenancy			*	S-21
36	Integration			*	S-10, S-25, S-46, S-53, S-56, S-62

Abbreviations: MSA, microservice architecture; SOA,service-oriented architecture.

TABLE 6 Successful factors after migration to microservices

ID	Factor type	Description	Studies
1	List of factors	Logs, Codebase, Config, Dependencies, Backing Services, Build-Release and Run, Port Binding, Process, Disposability, Concurrency, Administration Process, Development and Production Parity	S-3
2	Application size	Reducing the size of monolithic legacy applications and more comfortable maintaining the code and creating smaller ones.	S-11
3	Performance & replacement	Replacement facilitated Performance increases, much lower levels of support	S-19, S-57
4	Cost & time	Low Cost, Timely Development	S-24
5	Flexible system	Maintenance, Deployment, Scalability, Resource Management, Production deployment, Future development, Platform hosting	S-25
	Achievable attributes	Maintainability improvement, Scalability improvement, Architectural complexity reduction, Simplifies distributed work, Performance improvement, Testability, Separation of concerns, Single clear responsibility, System understandability	S-42
7	Portability	Portability	S-61
8	Reducing complexity	This research has reported: simpler integration, reduced complexity, higher cohesion, lower coupling, performance increase, and better reusability after migrating to a microservices	S-62

TABLE 7 Challenges of migrating the monolithic systems to MSA

ID	Challenge type	Description	Studies
1	Knowledge limitation	Patterns and bad practices: lack of knowledge	S-2
2	Cost	Decomposition, Deployment, Technology Heterogeneity, Scalability, Client-Server Interaction, Inter-Process Communication between Microservices, Monitoring	S-12
3	Automated testing	During the migration process, the most occurring issues are automated testing, system decomposition, high coupling, boundaries of service identification dilemmas	S-14
4	Multitenancy, statefulness, and data consistency	Multitenancy, statefulness, and data consistency are three issues of microservice migration.	S-21
5	Less relevant skills	Less relevant skills, the dependence on Software organizational culture and structure, on-premises applications, administration, troubles related to the decomposition of a monolithic, master information management, coordination and movement, testing, and execution.	S-32
6	Complexity in decoupling	The principal issues detailed are the unpredictability to decompose from the monolithic legacy systems and the splitting of information in traditional database communication among services.	S-42

(Continues)

TABLE 7 (Continued)

ID	Challenge type	Description	Studies
7	Data communication	The key issue is figuring out how to manage data transfer from the monolith to the new microservices.	S-45
8	Guidance	There isn't a good guide to decomposing services.	S-54
9	Docker platform issue	Docker for the Windows operating system is still not as advanced as the Linux partner, and network support for the Windows system is deficient as compared with the Linux system.	S-57
10	Functional issues	There are four critical issues in the transformation procedure. The first is identified with the distinguishing proof of functionalities. This isn't paltry, particularly in instances of enormous modules through which functionalities are dispersed and tangled among themselves. The subsequent test is the meaning of ideal limits among up-and-comer highlights for microservices. There follows the third test to conclude which will be changed over to microservices. The fourth test was identified by cautiously breaking down these up-and-comer microservices concerning their particular granularity and individual attachment.	S-58
11	Cost risk	Transformation to microservices is an inclined error procedure with profound results bringing about significant expenses for disasters—complexities and difficulties at conceivably a high cost.	S-62
12	Technical	Technical Challenges: The most concerning issue of cases is to isolate the services. It might require a lot of effort & time to refactor the services from the legacy architecture.	S-36
13	Organizational	Challenges of Organizational: most investigated organization structure is one of them. To grow an excessive system, the association must adjust its structure to the system boundary of architecture.	S-36
14	Migration to MSA	Decomposition of monolithic, Splitting of Data and transformation of databases, Service communication, Effort assessment, Library conversion effort, and return on investment are expected long-term.	S-42

5 | CONCLUSION

One of the most used research methodologies for extensive literature is the systematic mapping study. The MSA, a generally recently created paradigm, gets incredible consideration from the software industries. Besides, the advantages of using MSA, are independent development, deployment, highly scalable, and lightweight mechanisms after migrating from monolithic. Since the development of the research work concerning the adoption of MSA by migrating from monolithic, we can not reach reliable determinations. However, because of the available work of the exercised research, the findings of this extent might be determined. The goal of this mapping study is to find answers to our defined research questions. The awareness and significance of MSA are described in this article. In this SMS, 73 studies were ultimately chosen. We identified the migration approaches from monolithic to MSA and the most investigated quality attributes that are compromised during migration from monolithic to MSA. The following aspects of this study are examined: publication patterns, primary research settings, research focus, migration strategies, migration difficulties, successful migration factors, and possibilities for industry adoption. We compile the information and provide an overview of the state of the art. In the end, our plotting investigation exhibits an inclination of importance for this study. The quantity of production in cutting-edge 2018 exists higher than in earlier years. Our findings reveal that the volume of published studies that were conducted in 2017 is likewise at a higher rate, however, it is progressively rising. Future studies on this topic must concentrate on microservices for IoT and underwater IoT to investigate and provide the most effective solutions.

AUTHOR CONTRIBUTIONS

Conceptualization: Abdul Razzaq. *Methodology*: Abdul Razzaq. *Resources*: Abdul Razzaq. *Validation:* Abdul Razzaq; *Writing—original draft preparation:* Abdul

Razzaq; *Writing—review and editing:* Abdul Razzaq, Shahbaz Ahmed Khan Ghayyur. All authors have read and agreed to the published version of the manuscript.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

Not applicable.

ORCID

Abdul Razzaq http://orcid.org/0000-0002-4465-6365

REFERENCES

1. D. Liu, C.-H. Lung, and S. A. Ajila. Adaptive Clustering Techniques for Software Components and Architecture. 2015 IEEE 39th Annual Computer Software and Applications Conference. Vol. 3. IEEE; 2015.
2. G. Granchelli, M. Cardarelli, P. Di Francesco, I. Malavolta, L. Iovino, and A. Di Salle, *Towards recovering the software architecture of microservice-based systems*. In: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), IEEE; 2017, pp. 46–53.
3. Jaramillo, D., Nguyen, D. V., Smart, R, *Leveraging microservices architecture by using Docker technology*. In: SouthEastCon 2016, IEEE; 2016, pp. 1–5.
4. I. Trabelsi, M. Abdellatif, A. Abubaker, N. Moha, S. Mosser, S. Ebrahimi-Kahou, and Y. G. Guéhéneuc, *From legacy to microservices: A type-based approach for microservices identification using machine learning and semantic analysis*, J Softw. Evol. Process, e2503.
5. F. Ponce, G. Márquez, and H. Astudillo, *Migrating from monolithic architecture to microservices: A rapid review*. In: 2019 38th International Conference of the Chilean Computer Science Society (SCCC), IEEE; 2019, pp. 1–7.
6. A. Mohsen, and A. Ibrahim, *Requirements reconciliation for scalable and secure microservice (de)composition*. In: 2016 IEEE 24th International Requirements Engineering Conference Workshops. IEEE, 2016.
7. A. Nuha, N. Ali, and R. Evans, *A systematic mapping study in microservice architecture*. In: 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications, IEEE, 2016, pp. 44–51.
8. S. Apel, F. Hertrampf, and S. Späthe, *Toward a knowledge model focusing on microservices and cloud computing*, Concurr. Comput. Pract. Exp. **32** (2020), e5414.
9. W. K. G. Assunção, T. E. Colanzi, L. Carvalho, A. Garcia, J. A. Pereira, M. J. de Lima, and C. Lucena, *Analysis of a many-objective optimization approach for identifying microservices from legacy systems*, Empir. Softw. Eng. **27** (2022), no. 2, 51.
10. A. Balalaie, A. Heydarnoori, P. Jamshidi, D. A. Tamburri, and T. Lynn, *Microservices migration patterns*, Softw. Pract. Exp. **48** (2018), no. 11, 2019–2042.
11. G. Blinowski, A. Ojdowska, and A. Przybylek, *Monolithic vs. microservice architecture: A performance and scalability evaluation*, IEEE Access **10** (2022), 20357–20374.
12. J. Correia and A. Rito Silva, *Identification of monolith functionality refactorings for microservices migration*, Softw. Pract. Exp. **52** (2022), 2664–2683.
13. N. H. Do, T. Van Do, X. Thi Tran, L. Farkas, and C. Rotter, *A scalable routing mechanism for stateful microservices*, 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), IEEE, 2017, pp. 72–78, https://doi.org/10.1109/ICIN.2017.7899252
14. Christian Esposito, Aniello Castiglione, and Kim-Kwang Raymond Choo, *Challenges in delivering software in the cloud as microservices. IEEE cloud computing published by the IEEE computer society*, IEEE; **3** (2016), 10–14.
15. C. Esposito, A. Castiglione, and K. K. R. Choo, *Challenges in delivering software in the cloud as microservices*, IEEE Cloud Comput. **3** (2016), no. 5, 10–14.
16. L. Florio, E. Di Nitto. *An approach to introduce decentralized autonomic behavior in microservices architectures*. In: 2016 IEEE International Conference on Autonomic Computing (ICAC), IEEE, 2016, pp. 357–362.
17. A. F. A. Freire, A. F. Sampaio, L. H. L. Carvalho, O. Medeiros, and N. C. Mendonça, *Migrating production monolithic systems to microservices using aspect oriented programming*, Softw. Pract. Exp. **51** (2021), no. 6, 1280–1307.
18. S. A. K. Ghayyur et al., *"Matrix clustering based migration of system application to microservices architecture."*, Int. J. Adv. Comput. Sci. Appl. **9.1** (2018), 284–296.
19. S. A. K. Ghayyur, A. Razzaq, S. Ullah, and S. Ahmed, *Matrix clustering basedmigration of system application tomicroservices architecture*, Int. J. Adv. Comput. Sci. Appl. **9** (2018), no. 1, 284–296.
20. S. Haselböck and R. Weinreich, *Decision guidance models for microservice monitoring*. In: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), IEEE, 2017, pp. 54–61.
21. S. Hassan, R. Bahsoon, and R. Buyya, *Systematic scalability analysis for microservices granularity adaptation design decisions*, Softw. Pract. Exp. **52** (2022), no. 6, pp. 1378–1401. https://doi.org/10.1002/spe.3069
22. S. Hassan, and R. Bahsoon, *Microservices and their design trade-offs: A self-adaptive roadmap*, In: 2016 IEEE International Conference on Services Computing (SCC), IEEE, 2016, pp. 813–818.
23. W. Hasselbring, and G. Steinacker, *Microservice architectures for scalability, agility and reliability in e-commerce*. In: 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), IEEE, 2017, pp. 243–246.
24. H. Khazaei, C. Barna, N. Beigi-Mohammadi, and M. Litoiu, *Efficiency Analysis of Provisioning Microservices*. In: 2016 IEEE International conference on cloud computing technology and science (CloudCom), IEEE, 2016, pp. 261–268.
25. H. Knoche. *Sustaining runtime performance while incrementally modernizing transactional monolithic software towards microservices*. In: Proceedings of the 7th ACM/SPEC on International Conference on Performance Engineering, 2016, pp. 121–124.
26. M. Labiadh, C. Obrecht, C. Ferreira da Silva, and P. Ghodous, *A microservice-based framework for exploring data selection in*

cross-building knowledge transfer, Serv. Oriented Comput. Appl. **15** (2021), no. 2, 97–107.
27. M. M. Lehman, *Laws of software evolution revisited*. In European Workshop on Software Process Technology, Springer, Berlin, Heidelberg, 1996, pp. 108–124.
28. J. Lin, L. C. Lin, and S. Huang, *Migrating web applications to clouds with microservice architectures*. In: 2016 International conference on applied system innovation (ICASI), IEEE, 2016, pp. 1–4.
29. J. B. Moreira, H. Mamede, V. Pereira, and B. Sousa, *Next generation of microservices for the 5G service-based architecture*, Int. J. Netw. Manage. **30** (2020), no. 6, e2132.
30. R. V. O'Connor, P. Elger, and P. M. Clarke, *Continuous software engineering—A microservices architecture perspective*, J. Softw. Evol. Process **29** (2017), no. 11, e1866.
31. V. Raj and S. Ravichandra, *A service graph based extraction of microservices from monolith services of service-oriented architecture*, Softw. Pract. Exp. **52** (2022), 1661–1678.
32. A. Razzaq, *A systematic review on software architectures for iot systems and future direction to the adoption of microservices architecture*, SN Comput. Sci. **1** (2020), 350.
33. C. Rotter, J. Illés, G. Nyíri, L. Farkas, G. Csatári, and G. Huszty, "Telecom strategies for service discovery in microservice environments," 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, 2017, pp. 214-218.
34. G. Shahmohammadi, S. Jalili, and S. M. H. Hasheminejad, *Identification of system software components using clustering approach*, J. Object Technol. **9** (2010), 77–98.
35. Z. Xiao, I. Wijegunaratne, and X. Qiang, *Reflections on SOA and Microservices*. In: 2016 4th International Conference on Enterprise Systems (ES), IEEE, 2016, pp. 60–67.
36. G. Yu, P. Chen, and Z. Zheng, *Microscaler: Cost-effective scaling for microservice applications in the cloud with an online learning approach*, IEEE Trans. Cloud Comput. **10** (2020), 1100–1116.

AUTHOR BIOGRAPHIES

Abdul Razzaq is currently working as Software Solution Architect and PhD at Zhejiang University of China. He did his MS in Software Engineering, in 2018 at International Islamic University of Islamabad (IIUI), Pakistan. His research of interest areas are Software Engineering, Blockchain, Internet of Things, Ocean Internet of Technologies, AI, and Mobile Computing.

Shahbaz A. K. Ghayyur is an assistant professor at IIUI. He did his MS in Software Engineering, in 2007 at International Islamic University of Islamabad (IIUI), Pakistan and PhD degree in computer sciences from Preston University Islamabad, Pakistan, in 2018. His research of interest areas are Software Engineering, Blockchain, Internet of Things, Software Processes, Requirement Engineering.

How to cite this article: 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*, Comput. Appl. Eng. Educ. (2022), 1–31. https://doi.org/10.1002/cae.22586

APPENDIX A

See Table A9.

TABLE A1 Maximum used spots in the research part

ID	Selected studies	Publication type	Year	Total citation
S-1	Mazzara, Manuel et al. "Microservices Science and Engineering." Advances in Intelligent Systems and Computing, 2018, pp. 11-20. Springer International	Conference	2016	8
S-2	D. Taibi and V. Lenarduzzi, "On the Definition of Microservice Bad Smells," in IEEE Software, vol. 35, no. 3, pp. 56-62, 2018.	Journal	2018	54
S-3	K. Bakshi, "Microservices-based software architecture and approaches," 2017 IEEE Aerospace Conference, Big Sky, MT, 2017, pp. 1-8.	Conference	2017	30
S-4	O. Al-Debagy and P. Martinek, "A Comparative Review of Microservices and Monolithic Architectures," 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 2018, pp. 000149-000154.	Conference	2018	1
S-5	P. D. Francesco, I. Malavolta, and P. Lago, "Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption," 2017 IEEE International Conference on Software Architecture (ICSA), Gothenburg, 2017, pp. 21-30.	Conference	2017	126
S-6	T. Ueda, T. Nakaike, and M. Ohara, "Workload characterization for microservices," 2016 IEEE International Symposium on Workload Characterization (IISWC), Providence, RI, 2016, pp. 1-10.	Conference	2016	62
S-7	Z. Xiao, I. Wijegunaratne and X. Qiang, "Reflections on SOA and Microservices," 2016 4th International Conference on Enterprise Systems (ES), Melbourne, VIC, 2016, pp. 60-67.	Conference	2016	46
S-8	P. D. Francesco, "Architecting Microservices," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 224-229.	Workshop	2017	27
S-9	D. Richter, M. Konrad, K. Utecht, and A. Polze, "Highly-Available Applications on Unreliable Infrastructure: Microservice Architectures in Practice," 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), Prague, 2017, pp. 130-137.	Conference	2017	16
S-10	S. Sharma, N. Uniyal, B. Tola, and Y. Jiang, "On Monolithic and Microservice Deployment of Network Functions," 2019 IEEE Conference on Network Softwarization (NetSoft), Paris, France, 2019, pp. 387-395.	Conference	2019	1
S-11	Levcovitz, A., Terra, R., & Valente, M.T. (2016). Towards a Technique for Extracting Microservices from Monolithic Enterprise Systems. ArXiv, abs/1605.03175.	Workshop	2016	58
S-12	A. Koschel, I. Astrova, and J. Dötterl, "Making the move to microservice architecture," 2017 International Conference on Information Society (i-Society), Dublin, 2017, pp. 74-79.	Conference	2017	3
S-13	M. Ahmadvand and A. Ibrahim, "Requirements Reconciliation for Scalable and Secure Microservice (De)composition," 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), Beijing, 2016, pp. 68-73.	Workshop	2016	30
S-14	M. Cojocaru, A. Uta, and A. Oprescu, "Attributes Assessing the Quality of Microservices Automatically Decomposed from Monolithic Applications," 2019 18th International Symposium on Parallel and Distributed Computing (ISPDC), Amsterdam, Netherlands, 2019, pp. 84-93.	Conference	2019	1

(Continues)

TABLE A1 (Continued)

ID	Selected studies	Publication type	Year	Total citation
S-15	Zaschke, C. (2019). Elaboration of a Domain Model for Migrating the Monolithic Software Architecture of a Data Management Server into a Microservice Architecture. KMIS.	Conference	2019	0
S-16	M. Kamimura, K. Yano, T. Hatano, and A. Matsuo, "Extracting Candidates of Microservices from Monolithic Application Code," 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, Japan, 2018, pp. 571-580.	Conference	2018	1
S-17	G. Mazlami, J. Cito and P. Leitner, "Extraction of Microservices from Monolithic Software Architectures," 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, 2017, pp. 524-531.	Conference	2017	61
S-18	R. Chen, S. Li, and Z. Li, "From Monolith to Microservices: A Dataflow-Driven Approach," 2017 24th Asia-Pacific Software Engineering Conference (APSEC), Nanjing, 2017, pp. 466-475.	Conference	2017	21
S-19	J. Gouigoux and D. Tamzalit, "From Monolith to Microservices: Lessons Learned on an Industrial Migration to a Web Oriented Architecture," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 62-65.	Workshop	2017	35
S-20	Balalaie, Armin et al. "Microservices Migration Patterns." Software: Practice And Experience, 2018. Wiley	Journal	2018	57
S-21	A. Furda, C. Fidge, O. Zimmermann, W. Kelly, and A. Barros, "Migrating Enterprise Legacy Source Code to Microservices: On Multitenancy, Statefulness, and Data Consistency," in IEEE Software, vol. 35, no. 3, pp. 63-72, May/June 2018.	Journal	2018	30
S-22	J. Kazanavičius and D. Mažeika, "Migrating Legacy Software to Microservices Architecture," 2019 Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2019, pp. 1-5.	Conference	2019	1
S-23	C. Fan and S. Ma, "Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report," 2017 IEEE International Conference on AI & Mobile Services (AIMS), Honolulu, HI, 2017, pp. 109-112.	Conference	2017	18
S-24	J. Lin, L. C. Lin, and S. Huang, "Migrating web applications to clouds with microservice architectures," 2016 International Conference on Applied System Innovation (ICASI), Okinawa, 2016, pp. 1-4.	Conference	2016	13
S-25	E. Djogic, S. Ribic, and D. Donko, "Monolithic to microservices redesign of event driven integration platform," 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2018, pp. 1411-1414.	Conference	2018	4
S-26	A. Shimoda and T. Sunada, "Priority Order Determination Method for Extracting Services Stepwise from Monolithic System," 2018 7th International Congress on Advanced Applied Informatics (IIAI-AAI), Yonago, Japan, 2018, pp. 805-810.	Conference	2018	0
S-27	M. Tusjunt and W. Vatanawood, "Refactoring Orchestrated Web Services into Microservices Using Decomposition Pattern," 2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018, pp. 609-613.	Conference	2018	0
S-28	G. Kecskemeti, A. C. Marosi and A. Kertesz, "The ENTICE approach to decompose monolithic services into microservices," 2016 International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, 2016, pp. 591-596.	Conference	2016	31
S-29	Silva, H.H., Carneiro, G.D., & Monteiro, M.P. (2019). Towards a Roadmap for the Migration of Legacy Software Systems to a Microservice based Architecture. CLOSER.	Conference	2019	0

TABLE A1 (Continued)

ID	Selected studies	Publication type	Year	Total citation
S-30	Auer, Florian et al. "Towards Defining A Microservice Migration Framework." Proceedings Of The 19Th International Conference On Agile Software Development Companion - XP '18, 2018. ACM Press	Conference	2018	3
S-31	Sarita and S. Sebastian, "Transform Monolith into Microservices using Docker," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp. 1-5.	Conference	2017	0
S-32	Baskarada, S., Nguyen, V., & Koronios, A. (2018). Architecting Microservices: Practical Opportunities and Challenges.	Journal	2018	10
S-33	J. Bogner, J. Fritzsch, S. Wagner, and A. Zimmermann, "Assuring the Evolvability of Microservices: Insights into Industry Practices and Challenges," 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), Cleveland, OH, USA, 2019, pp. 546-556.	Conference	2019	5
S-34	C. Esposito, A. Castiglione, and K. R. Choo, "Challenges in Delivering Software in the Cloud as Microservices," in IEEE Cloud Computing, vol. 3, no. 5, pp. 10-14, Sept.-Oct. 2016.	Journal	2016	57
S-35	F. Rademacher, J. Sorgalla, and S. Sachweh, "Challenges of Domain-Driven Microservice Design: A Model-Driven Perspective," in IEEE Software, vol. 35, no. 3, pp. 36-43, May/June 2018.	Journal	2018	25
S-36	Kalske, M., Mäkitalo, N., & Mikkonen, T. (2017). Challenges When Moving from Monolith to Microservice Architecture. ICWE Workshops.	Workshop	2017	22
S-37	N. C. Mendonça, P. Jamshidi, D. Garlan, and C. Pahl, "Developing Self-Adaptive Microservice Systems: Challenges and Directions," in IEEE Software.	Journal	2019	9
S-38	R. M. Munaf, J. Ahmed, F. Khakwani, and T. Rana, "Microservices Architecture: Challenges and Proposed Conceptual Design," 2019 International Conference on Communication Technologies (ComTech), Rawalpindi, Pakistan, 2019, pp. 82-87.	Conference	2019	0
S-39	J. Fritzsch, J. Bogner, S. Wagner, and A. Zimmermann, "Microservices Migration in Industry: Intentions, Strategies, and Challenges," 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME), Cleveland, OH, USA, 2019, pp. 481-490.	Conference	2019	1
S-40	S. K. Garg, J. Lakshmi, and J. Johny, "Migrating VM Workloads to Containers: Issues and Challenges," 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, 2018, pp. 778-785.	Conference	2018	1
S-41	T. Yarygina and A. H. Bagge, "Overcoming Security Challenges in Microservice Architectures," 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), Bamberg, 2018, pp. 11-20.	Conference	2018	20
S-42	D. Taibi, V. Lenarduzzi and C. Pahl, "Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation," in IEEE Cloud Computing, vol. 4, no. 5, pp. 22-32, September/October 2017.	Journal	2017	89
S-43	N. C. Mendonça, P. Jamshidi, D. Garlan, and C. Pahl, "Developing Self-Adaptive Microservice Systems: Challenges and Directions," in IEEE Software.	Conference	2015	9
S-44	Soares de Toledo, Saulo et al. "Architectural Technical Debt In Microservices: A Case Study In A Large Company." 2019 IEEE/ACM International Conference On Technical Debt (Techdebt), 2019. IEEE	Conference	2019	4
S-45	Smid, Antonin, et al. "Case Study On Data Communication In Microservice Architecture." Proceedings Of The Conference On Research In Adaptive And Convergent Systems - RACS '19, 2019. ACM Press	Conference	2019	0

(Continues)

TABLE A1 (Continued)

ID	Selected studies	Publication type	Year	Total citation
S-46	V. Debroy and S. Miller, "Overcoming Challenges with Continuous Integration and Deployment Pipelines When Moving From Monolithic Apps to Microservices: An experience report from a small company," in IEEE Software.	Journal	2019	1
S-47	E. Fernandes Mioto de Oliveira dos Santos and C. M. Lima Werner, "A Survey on Microservices Criticality Attributes on Established Architectures," 2019 International Conference on Information Systems and Software Technologies (ICI2ST), Quito, Ecuador, 2019, pp. 149-155.	Conference	2019	0
S-48	Bogner, Justus et al. "Limiting Technical Debt With Maintainability Assurance". Proceedings Of The 2018 International Conference On Technical Debt - Techdebt '18, 2018. ACM Press	Conference	2018	7
S-49	P. Di Francesco, P. Lago, and I. Malavolta, "Migrating Towards Microservice Architectures: An Industrial Survey," 2018 IEEE International Conference on Software Architecture (ICSA), Seattle, WA, 2018, pp. 29-2909.	Conference	2018	38
S-50	S. Haselböck, R. Weinreich and G. Buchgeher, "An Expert Interview Study on Areas of Microservice Design," 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), Paris, 2018, pp. 137-144.	Conference	2018	2
S-51	S. Cisneros-Cabrera, P. Sampaio, and N. Mehandjiev, "A B2B Team Formation Microservice for Collaborative Manufacturing in Industry 4.0," 2018 IEEE World Congress on Services (SERVICES), San Francisco, CA, 2018, pp. 37-38.	Conference	2018	2
S-52	Carvalho, Luiz et al. "Analysis Of The Criteria Adopted In Industry To Extract Microservices." 2019 IEEE/ACM Joint 7Th International Workshop On Conducting Empirical Studies In Industry (CESI) And 6Th International Workshop On Software Engineering Research And Industrial Practice (SER& IP), 2019. IEEE,	Workshop	2019	3
S-53	J. A. Bigheti, M. M. Fernandes and E. P. Godoy, "Control as a Service: A Microservice Approach to Industry 4.0," 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0& IoT), Naples, Italy, 2019, pp. 438-443.	Workshop	2019	0
S-54	H. Zhang, S. Li, Z. Jia, C. Zhong, and C. Zhang, "Microservice Architecture in Reality: An Industrial Inquiry," 2019 IEEE International Conference on Software Architecture (ICSA), Hamburg, Germany, 2019, pp. 51-60.	Conference	2019	0
S-55	J. Bogner, J. Fritzsch, S. Wagner, and A. Zimmermann, "Microservices in Industry: Insights into Technologies, Characteristics, and Software Quality," 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), Hamburg, Germany, 2019, pp. 187-195.	Conference	2019	8
S-56	J. GOUIGOUX and D. TAMZALIT, "Functional-First" Recommendations for Beneficial Microservices Migration and Integration Lessons Learned from an Industrial Experience," 2019 IEEE International Conference on Software Architecture Companion (ICSA-C), Hamburg, Germany, 2019, pp. 182-186.	Conference	2019	0
S-57	S. Sarkar, G. Vashi, and P. P. Abdulla, "Towards Transforming an Industrial Automation System from Monolithic to Microservices," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, 2018, pp. 1256-1259.	Conference	2018	1
S-58	da Silva, Hugo Henrique S. et al. "An Experience Report From The Migration Of Legacy Software Systems To Microservice Based Architecture." 16Th International Conference On Information Technology-New Generations (ITNG 2019), 2019, pp. 183-189. Springer International Publishing	Conference	2019	1
S-59	Villamizar, Mario et al. "Cost Comparison Of Running Web Applications In The Cloud Using Monolithic, Microservice, And AWS Lambda Architectures."	Journal	2017	43

TABLE A1 (Continued)

ID	Selected studies	Publication type	Year	Total citation
	Service Oriented Computing And Applications, vol 11, no. 2, 2017, pp. 233-247. Springer Science And Business Media LLC			
S-60	M. Mazzara, N. Dragoni, A. Bucchiarone, A. Giaretta, S. T. Larsen and S. Dustdar, "Microservices: Migration of a Mission Critical System," in IEEE Transactions on Services Computing.	Journal	2018	29
S-61	Balalaie, Armin et al. "Migrating To Cloud-Native Architectures Using Microservices: An Experience Report." Communications In Computer And Information Science, 2016, pp. 201-215. Springer International Publishing	Conference	2016	136
S-62	Carrasco, Andrés et al. "Migrating Towards Microservices: Migration And Architecture Smells." Proceedings Of The 2Nd International Workshop On Refactoring - Iwor 2018, 2018. ACM Press	Workshops	2018	19
S-63	Ren, Zhongshan et al. "Migrating Web Applications From Monolithic Structure To Microservices Architecture." Proceedings Of The Tenth Asia-Pacific Symposium On Internetware - Internetware '18, 2018. ACM Press	Conference	2018	5
S-64	C. Fehling, F. Leymann, S. T. Ruehl, M. Rudek and S. Verclas, "Service Migration Patterns -- Decision Support and Best Practices for the Migration of Existing Service-Based Applications to Cloud Environments," 2013 IEEE 6th International Conference on Service-Oriented Computing and Applications, Koloa, HI, 2013, pp. 9-16.	Conference	2013	36
S-65	H. Knoche and W. Hasselbring, "Using Microservices for Legacy Software Modernization," in IEEE Software, vol. 35, no. 3, pp. 44-49, May/June 2018.	Journal	2018	33
S-66	C. Pautasso, O. Zimmermann, M. Amundsen, J. Lewis, and N. Josuttis, "Microservices in Practice, Part 1: Reality Check and Service Design," in IEEE Software, vol. 34, no. 1, pp. 91-98, Jan.-Feb. 2017.	Journal	2017	89
S-67	C. Pautasso, O. Zimmermann, M. Amundsen, J. Lewis, and N. Josuttis, "Microservices in Practice, Part 2: Service Integration and Sustainability," in IEEE Software, vol. 34, no. 2, pp. 97-104, Mar.-Apr. 2017.	Journal	2017	21
S-68	C. Berger, B. Nguyen, and O. Benderius, "Containerized Development and Microservices for Self-Driving Vehicles: Experiences & Best Practices," 2017 IEEE International Conference on Software Architecture Workshops (ICSAW), Gothenburg, 2017, pp. 7-12.	Workshop	2017	14
S-69	O. Al-Debagy and P. Martinek, "Extracting Microservices' Candidates from Monolithic Applications: Interface Analysis and Evaluation Metrics Approach," 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), 2020, pp. 289-294.	Conference	2020	1
S-70	D. Kuryazov, D. Jabborov and B. Khujamuratov, "Towards Decomposing Monolithic Applications into Microservices," 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), 2020.	Conference	2020	5
S-71	Kalia, Anup K., et al. "Mono2Micro: an AI-based toolchain for evolving monolithic enterprise applications to a microservice architecture." Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2020.	Symposium	2020	1
S72	Brito, Miguel, Jácome Cunha, and João Saraiva. "Identification of microservices from monolithic applications through topic modelling." Proceedings of the 36th Annual ACM Symposium on Applied Computing. 2021.	Symposium	2021	18
S-73	Auer, Florian, et al. "From monolithic systems to microservices: an assessment framework." *Information and Software Technology* 137 (2021): 106600.	Journal	2021	2

TABLE A2 Number of selected studies and proportion over the venues of publication

Publisher	Publication venue	Acronym	Type	No.	%
IEEE	IEEE Software		Journal	8	13%
	Cloud Computing		Journal	2	3%
	Transactions on Services Computing		Journal	1	1.5%
	International Conference on Software Architecture	ICSA	Conference	3	4.5%
	Asia-Pacific Software Engineering Conference (APSEC)	APSEC	Conference	2	3%
	International Conference on Software Maintenance and Evolution (ICSME)		Conference	2	3%
	International Conference on Software Architecture Companion (ICSA-C)	ICSA-C	Conference	2	3%
	IEEE Aerospace Conference		Conference	1	1.5%
	International Conference on Enterprise Systems		Conference	1	1.5%
	International Conference on Software Quality, Reliability and Security		Conference	1	1.5%
	Conference on Network Softwarization		Conference	1	1.5%
	International Conference on Information Society		Conference	1	1.5%
	International Conference on Web Services		Conference	1	1.5%
	Open Conference of Electrical, Electronic and Information Sciences (eStream)		Conference	1	1.5%
	International Conference on AI & Mobile Services		Conference	1	1.5%
	International Conference on Applied System Innovation (ICASI)	ICASI	Conference	1	1.5%
	International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)	MIPRO	Conference	1	1.5%
	International Congress on Advanced Applied Informatics		Conference	1	1.5%
	International Conference on Computer and Communications		Conference	1	1.5%
	International Conference on High Performance Computing & Simulation (HPCS)	HPCS	Conference	1	1.5%
	International Conference on Computing, Communication, Control, and Automation (ICCUBEA)		Conference	1	1.5%
	International Conference on Communication Technologies (ComTech 2019)	ComTech	Conference	1	1.5%
	International Conference on Cloud Computing		Conference	1	1.5%
	Symposium on Service-Oriented System Engineering		Conference	1	1.5%
	Computing Colombian Conference (10CCC)	10CCC	Conference	1	1.5%
	International Conference on Information Systems and Software Technologies (ICI2ST)	ICI2ST	Conference	1	1.5%
	International Conference on Service-Oriented Computing and Applications		Conference	2	3%
	World Congress on Services (SERVICES)	SERVICES	Conference	1	1.5%
	International Conference on Emerging Technologies and Factory Automation (ETFA)	ETFA	Conference	1	1.5%
	International Symposium on Computational Intelligence and Informatics		Symposium	1	1.5%

TABLE A2 (Continued)

Publisher	Publication venue	Acronym	Type	No.	%
	International Symposium on Workload Characterization (IISWC)	IISWC	Symposium	1	1.5%
	International Symposium on Parallel and Distributed Computing (ISPDC)	ISPDC	Symposium	1	1.5%
	International Conference on Software Architecture Workshops		Workshop	3	4.5%
	International Requirements Engineering Conference Workshops		Workshop	1	1.5%
	Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0& IoT)	MetroInd4.0 & IoT	Workshop	1	1.5%
ACM	International Conference on Agile Software Development		Conference	1	1.5%
	ACM International Conference on Technical Debt (TechDebt)	TechDebt	Conference	1	1.5%
	Conference on Research in Adaptive and Convergent Systems		Conference	1	1.5%
	International Conference on Technical Debt		Conference	1	1.5%
	Asia-Pacific Symposium on Internetware		Symposium	1	1.5%
	International Workshop on Conducting Empirical Studies in Industry and International Workshop on Software Engineering Research and Industrial Practice		Workshop	1	1.5%
	International Workshop on Refactoring	IWoR	Workshop	1	1.5%
SpringerLink	Service-Oriented Computing and Applications (Journal)		Journal	1	1.5%
	Communications in Computer and Information Science		Journal	1	1.5%
	International Conference on Information Technology-New Generations (ITNG 2019)	ITNG	Conference	1	1.5%
	International Conference in Software Engineering for Defence Applications, SEDA 2015	SEDA	Conference	1	1.5%
	Current Trends in Web Engineering ICWE Workshops		Workshop	1	1.5%
SCITEPRESS	International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management		Conference	1	1.5%
	International Conference on Cloud Computing and Services Science		Conference	1	1.5%
Taylor & Francis	Journal of Computer Information Systems		Journal	1	1.5%
WILEY	Software Practice & Experience		Journal	1	1.5%
Semantic Scholar	Brazilian Workshop on Software Visualization, Evolution and Maintenance (VEM)		Workshop	1	1.5%

TABLE A3 Studies distribution over industries and academia

Industrial (16, 24%)	Academia (20, 29%)	Mixed (32, 47%)
S-3, S-9, S-13, S-17, S-20, S-24, S-30, S-32, S-39, S-47, S-50, S-52, S-53, S-57, S-60, S-68	S-1, S-5, S-7, S-12, S-15, S-18, S-21, S-22, S-23, S-31, S-34, S-35, S-37, S-38, S-40, S-41, S-43, S-58, S-66, S-67	S-2, S-4, S-6, S-8, S-10, S-11, S-14, S-16, S-19, S-25, S-26, S-27, S-28, S-29, S-33, S-36, S-42, S-44, S-45, S-46, S-48, S-49, S-51, S-54, S-55, S-56, S-59, S-61, S-62, S-63, S-64, S-65

TABLE A4 Studies distribution over migration and awareness

Migration (18, 26%)	Awareness (32, 48%)	Mixed (18, 26%)
S-11, S-13, S-15, S-16, S-17, S-18, S-19, S-23, S-24, S-25, S-26, S-27, S-28, S-30, S-31, S-57, S-63, S-65	S-1, S-2, S-3, S-4, S-5, S-6, S-7, S-8, S-9, S-10, S-12, S-22, S-29, S-32, S-33, S-34, S-35, S-38, S-41, S-43, S-44, S-47, S-48, S-50, S-51, S-53, S-55, S-58, S-59, S-66, S-67, S-68	S-14, S-20, S-21, S-36, S-37, S-39, S-40, S-42, S-45, S-46, S-49, S-52, S-54, S-56, S-60, S-61, S-62, S-64

TABLE A5 Studies distribution over research types

Evaluation (32, 47%)	Experience (7, 10%)	Validation (12, 18%)	Opinion (14, 21%)	Solution (1, 1.4%)	Philosophical (2, 2.6%)
S-2, S-4, S-5, S-10, S-15, S-16, S-19, S-20, S-22, S-23, S-26, S-31, S-33, S-36, S-38, S-39, S-41, S-42, S-43, S-44, S-45, S-46, S-47, S-48, S-49, S-50, S-52, S-53, S-54, S-55, S-59, S-64	S-3, S-9, S-17, S-56, S-61, S-62, S-68	S-11, S-13, S-14, S-18, S-24, S-25, S-27, S-28, S-40, S-57, S-63, S-65	S-1, S-6, S-7, S-12, S-21, S-29, S-30, S-32, S-34, S-35, S-37, S-51, S-58, S-60	S-8	S-66, S-67

TABLE A6 Studies distribution over data collection methods

Data collection methods	Studies	Total, %
Archive Analysis	S-3, S-4, S-9, S-10, S-11, S-13, S-15, S-16, S-17, S18, S-22, S-23, S-24, S-25, S-27, S-28, S-36, S38, S40, S-46, S-53, S-56, S-57, S-58, S-59, S-60, S-61, S-62, S-63, S-64, S-65, S-68	32, 47%
Observation	S-1, S-5, S-6, S-7, S-8, S-12, S-19, S-20, S-21, S-26, S-29, S-31, S-34, S-35, S-37, S-41, S-43, S-45, S-51, S-66, S-67	21, 31%
Interview	S-2, S-14, S-32, S-33, S-39, S-44, S-50, S-55	8, 12%
Questionnaire	S-30, S-47, S-48, S-52	4, 6%
Interviews, Questionnaires	S-42, S-49, S-54	3, 4%

TABLE A7 Studies distribution over research methods

Data collection methods	Studies	Total, %
Case Study	S-1, S-13, S-15, S-16, S-17, S-18, S-20, S-27, S-28, S-29, S-33, S-40, S-43, S-44, S-45, S-53, S-59, S-60	18, 26%
Survey	S-2, S-30, S-32, S-39, S-42, S-47, S-48, S-49, S-50, S-52, S-54, S-55	17.5%
Survey, case-study	S-14, S-36	2, 3%
Experience report	S-3, S-19, S-23, S-31, S-46, S-56, S-58, S-61, S-62, S-66, S-67	11, 16%
Controlled experiment	S-4, S-9, S-10, S-11, S-24, S-25, S-26, S-57, S-63, S-64, S-65, S-68	12, 17.5%
Not defined	S-5, S-6, S-7, S-8, S-12, S-21, S-22, S-34, S-35, S-37, S-38, S-41, S-51	13, 19%

TABLE A8 distribution of migration approaches over research methods

Investigation	Industry	S-13, S-17, S-20, S-24, S-30, S-57
	Academic	S-15, S-18, S-23, S-31
	Mixed	S-11, S-16, S-19, S-25, S-26, S-27, S-28, S-45, S-65
Research type	Evaluation	S-15, S-16, S-19, S-20, S-23, S-26, S-31, S-45
	Experience	S-17
	Validation	S-11, S-13, S-18, S-24, S-25, S-27, S-28, S-57, S-63, S-65
	Opinion	S-30
Research method	Case study	S-13, S-15, S-16, S-17, S-18, S-20, S-27, S-28, S-45
	Survey	S-30
	Experience report	S-19, S-23, S-31
	Controlled experiment	S-11, S-24, S-25, S-26, S-57, S-63, S-65

TABLE A9 Detail description of selected papers

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
1	Conference	Review paper	MSA Awareness to understand an emerging field.	Key principles of Service Science and Engineering	Collecting bibliographic references and links	misunderstandings related to microservices and Service-Oriented Architectures	S-1
2	Journal	Interviews by an experienced developer	To identify microservice-specific bad smells	The proposed catalog of smells can be used by practitioners	Eleven bad smells of microservices are classified as bad practices and practitioners consider harmful	bad practices	S-2
3	Conference	Discussion	awareness about the technical implementation of microservices by reviewing containerization	Factors list	Comparison	Dependency in legacy systems	S-3
4	Conference	Qualitative & quantitative synthesis methods	Awareness & implementation of MSA	Highlighted gaps in MSA research & future suggestions	Architectural challenges, view, and quality attributes related to MSA	Security issue	S-4
5	Conference	Mapping study (synthesize data)	Identification, classification, & evaluation of MSA	Categorization of research by classifying the framework for architecting microservices	Benefits for industrial adoption, the focus of research, and publication trends	MSA existing research has no clear view	S-5
6	Conference	Quantitative analysis	Workload behavior of MSA	Language runtime optimization technique	Performance analysis for both software and hardware perspectives	Workload issue of many services in MSA	S-6
7	Conference	Discussion & analysis	Compare SOA, API & MSA	Recommendation & best practices	Critical investigation	Service-oriented challenges	S-7
8	Workshop	Qualitative analysis	Addressing the challenges of MSA	Future work	Key properties, architecture designing & factors impact migration	MSA challenges	S-8
9	Conference	Empirical analysis	Understand the Impact of dependency	High availability of the system	Migration & MSA system's requirements	System failure	S-9
10	Conference	Empirical analysis	Migration to MSA	Scaling & decomposition	Migration challenges & performance comparison	System flexibility	S-10
11	Conference	Empirical analysis, Describe a technique	To define & extract microservices from monolithic systems	Reduced the size & took benefits	To find MSA candidates	Single large applications are risky & expensive to evolve	S-11

(Continues)

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
12	Conference	Qualitative analysis	To make flexible on-demand	List of Technical challenges	Characteristics of Monolithic & microservice architecture	Scalability cost	S-12
13	Workshop	Conceptual methodology	To adopt MSA	break monolithic systems into microservice candidates	Requirements engineers can reconcile security and a scalability requirement	Traditional systems have long software release cycles because of the system's complexity	S-13
14	Conference	Interviews with an industry expert	semi-automatic migration tools or techniques for MSA	Understanding of decomposition and migration techniques	MSA quality attributes from semi-automatic decomposition tools or techniques	Monolithic, & SOA to MSA migration understanding issues	S-14
15	Conference	Doman model	Migrate monolithic architecture to MSA	Best practices approach	Describe the procedures to create the model	Model creates challenges	S-15
16	Conference	Empirical analysis & method	Rapid modification & make a more adaptable system	Evaluated results by developers	A method for identification of MSA candidates	MSA candidate from a monolithic system	S-16
17	Conference	Quantitative metrics	Industrial legacy system migration to MSA	Recommendations of quality	Tackle the challenges of the Microservice extraction model	Migration challenges	S-17
18	Conference	Top-down analysis	Reduce the complexity of decomposition	Comparison & Evaluation	Dataflow driver algorithm	reduce the complexity	S-18
19	Workshop	Qualitative analysis	Understanding of migration to MSA	Technical lesson	Addressing three crucial questions	SA strategic and technical change from legacy to MSA	S-19
20	Journal	Qualitative empirical analysis	To migrate traditional architecture to MSA	Effective patterns for migration of system	Identification migration patterns	Legacy migration issues	S-20
21	Journal	Qualitative analysis	To understand the migration challenges	Best practices to develop a microservice	Elaborate on the migration challenges	Migration issues	S-21
22	Conference	Qualitative analysis	Awareness of migration	Benefits & drawbacks of migration methods	Elaborate & review migration techniques	Legacy migration to MSA	S-22
23	Conference	Empirical analysis	A migration process facilitates the monolithic transformation to MSA	provide valuable references to transform the existing systems into microservices	Report the challenges that are encountered in the process	SDLC methods migration issues	S-23
24	Conference	Empirical analysis	migration method for MSA to support the transformation of web applications to clouds	it helps the customers to receive customized services at a low cost & risk away	Presented a method directing web application transformation to selected cloud	migration to cloud	S-24

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
25	Conference	Qualitative method	Redesign of SOA to MSA	scalability, better resource management, maintenance, and deployment	Described the technical model of SOA integration	SOA migration issue	S-25
26	Conference	Empirically quantitative analysis	A desirable method for converting the monolithic system's functions into microservice	e-commerce system, feasibility was confirmed	Portrayed by numerically assessing the benefits and negative marks of moving over monolithic into microservices and requests of need.	Enterprise systems of information are required to have the adaptability to change, high quality	S-26
27	Conference	Decomposition pattern	flexibility, the productivity of constant changes, and continuous modifications	Business capabilities list for subsequent processes and refactor to microservices	Extracting services from existing SOA system & refactor to microservices	inflexibility for modification in the SOA system	S-27
28	Conference	Decomposing methodology	decomposing monolithic legacy services to microservices	optimized images in a distributed repository	microservices created through the proposed methodology	Less portability	S-28
29	Conference	Empirically qualitative analysis	migration of legacy software systems to microservices	Processes and issues found while the migration reported	characterizing the relevant steps of such guidelines	Migration to MSA	S-29
30	Conference	Decision-based framework	mitigate the process of deployment or decrease the effort of maintenance	Practitioners support understanding the benefits and issues of migration	work plan to identify a decision framework	insufficient knowledge to adopt MSA	S-30
31	Conference	Qualitative approach	Understanding the transformation from monolithic to MSA	can deploy easily more containers and can scale application	transform method, deployment strategy, a pattern of service discovery, strategy to build & ship microservices	Migration understanding	S-31
32	Journal	Interviews	Discuss the benefits & challenges	Factors list	Identification of novel factors	Microservices challenges	S-32
33	Conference	Qualitative interview	The usage of patterns, tools, and metrics to explore and apply evolvability assurance	Found the outer customers who relied on fundamental governance for pledge	Guidelines and principles of architecture and its rules for communicating services	Microservices challenges	S-33
34	Journal	Review & discussion	Awareness of using microservices	Flexible & low-cost development	improve their maintainability, scalability, and testability.	Legacy System Integration	S-34
35	Journal	Review & discussion	Overview of supporting tools to address the challenges	Tools understanding	Explore several challenges & overview of tools	Microservices challenges	S-35

(Continues)

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
36	Workshop	Survey method	Identify the reasons for the migration of monolithic to MSA	Tools list & overview	Findings expose the reasons for adopting MSA are scalability, complexity, and code ownership	Introduces new challenges by making a distributed system	S-36
37	Journal	Qualitative analysis	development key challenges of microservice-based application	Potential of a new direction to address most of the identified issues	Identification of challenges for self-adaptive microservices, new migration strategies	Self-adaptive MSA challenges	S-37
38	Conference	Empirical Conceptual design	Overview of MSA for researcher & practitioners	conceptual design and prospects.	Practical implementation of the proposed modules is present in the industry	Implementation challenges	S-38
39	Conference	Interviews	Planning for the migration process	14 migration scenarios	migration drivers, maintainability, and scalability	Lack knowledge	S-39
40	Conference	Systematic analysis	Highlighted issues & challenges with VM migration	Container using understanding	Container performance comparison	VM workload issue	S-40
41	Conference	Qualitative analysis	To address the lack of security guidelines & security framework for microservices	providing a taxonomy of microservices security	Examining the microservice architectural style considering its security issues	security concerns of a microservice system	S-41
42	Journal	Interviews	Awareness of the migration process and the challenges and advantages related to migration	List of the migration process	Identification of a framework based on a comparison of migration procedures	Issues for migrating to MSA	S-42
43	Conference	Case study	Analyzing microservice architecture patterns	Performance test execution results	Benefits & challenges		S-43
44	Conference	Qualitative analysis, interviews	identify risks challenges and solutions associated with the technical debt of architecture in microservices.	Solutions, issues, and possible risks	list of Technical Debt issues of Architecture, list of possible risks related to funding and inadequate prioritization	To constantly provide value to their customers through continuous delivery	S-44
45	Conference	Case study	data synchronization and improvement from monolithic data to microservices	automated data streaming	To address data synchronization, an automated data streaming system between databases	data communication management is a challenge	S-45

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
46	Journal	Experience report	Benefits after migration from monolithic to MSA	microservices have developed and have proven to be cost-efficient	the novel idea of infrastructure to support the CI & CD	migration challenges	S-46
47	Conference	Survey	Target to find the need for a method to measure the microservice's criticality	To measure the criticality was observed on reputable architectures by lack of an ashore method	Guidance about the evolution of architecture and maintenance	To measure the criticality by lack of stranded method	S-47
48	Conference	Online questionnaire	Systematic techniques have benefits for maintainability	Need to improve quality control in results of suggestions by industries	Applied processes, tools, and metrics	little scientific research on the industry	S-48
49	Conference	Empirical analysis	In industry, the adoption of microservices' migration practices	Industry future directions appropriate migrations and problems towards microservices	Challenges to address the organizational and technical perspectives	Technical & organizational migration issues	S-49
50	Conference	Interview	Shifting to MSA is needed to explore the importance of several areas of microservice	assessments of MSA design importance, and rationales for the providing valuations	relevant strategy for MSA	Investigation importance of MSA design	S-50
51	Conference	Qualitative analysis	Reducing the risk of suppliers and increasing potential coalition	To facilitate by team formation and joint manufacturing in Industry 4.0	strategy and the architecture of a B2B microservice	Business to Business (B2B) interoperability platforms	S-51
52	Workshop	Online survey	the improper decision during microservice extraction is mentioned as a lack of synthesized information to relevant criteria	suggest four criteria to support decisions	suggest academic techniques	Migration limited knowledge	S-52
53	Workshop	Case study	To acquire quality, productivity, and efficiency	MSA for I4.0 applications	provides an integrated, distributed, and flexible architecture	problems of cross-layer integration & interoperability in industrial systems	S-53
54	Conference	Qualitative methods (interviews)	Investigation of the microservice benefits gap between real industrial and vision	characterized the gaps in typical industrial practices and characteristics of microservice	confirmed the benefits of the microservices from practice	The gap between versions & industrial practices	S-54

(Continues)

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
55	Conference	A qualitative study (interviews)	Investigation & understanding of microservice in practice	findings into account when focused methods are used in industrial design.	Finding maintainability & software quality	practitioners and academia-based investigation gap	S-55
56	Conference	Systematic literature	An industrial and academic-based systematic review on tactics and architectural patterns for microservices	attributes: scalability, flexibility, testability, performance, and elasticity.	most investigated tactics and patterns of architecture	No well-clear perspectives on design decisions and patterns of architecture in the context of academia and industry for microservice	S-56
57	Conference	Experience report	Successful migration lesson learned	technical and integration outcomes	best practices		S-57
58	Conference	Containerized architecture	Adopting microservices-based containerized design by migrating the complex and distributed industrial automation systems	scalability and availability	workload handling, resource utilization, and reliability	Need flexible systems	S-58
59	Journal	Empirical evaluation analysis	Microservices Performance & cost evaluation	Cost reduction, scalability, and agility should be leveled with efforts of development and technical challenges	Scalability, independent development, and agility	infrastructure costs	S-59
60	Journal	Case study	Reveal how scalability effects positive by migrating the monolithic architecture to MSA	Achieve scalability	implemented a scalable microservice architecture	Monolithic scalability issue	S-60
61	Journal	Experience report	migrating a monolithic on-premise software architecture to microservices	After migration, availability & scalability	Experience report & lesson learned	distribution complexities	S-61
62	Workshop	Review paper	migration to microservices can be high costs for a mistake. need to understand bad smells related to migration	Potential solutions	Identify 9 bad smells of migration	lack of migration guidelines	S-62
63	Conference	Empirical study	To get monolithic system characteristics behavior of	Assistance and guidance of migration phase and	Transforming the monolithic legacy systems and data to microservices	Less focus on dynamic characteristics	S-63

TABLE A9 (Continued)

#	Type:	Technique/ Empirical/Survey	Objective	Results	Contribution	Limitation/Challenge	Ref.
			runtime and static structure by combining both	achieving low-performance cost and high accuracy			
64	Conference	Qualitative analysis	A part of the application stack to be migrated, the procedure to follow during the migration,	Best practices	Implementation of approach	distribution of application functionality issue	S-64
65	Journal	Experience report	Decomposing system transforming process to microservices	High maintainability	best practices	Migration challenges	S-65
66	Journal	Experience report	To understand the refactoring to MSA	Best practices characteristics	best-practice approach	Lack of knowledge of Migration awareness	S-66
67	Journal	Experience report	Awareness of decomposing a monolithic system to MSA	Scaling, statelessness, and interoperability	Comparison monolithic system, SOA, MSA	Integration issue in distributed systems	S-67
68	Workshop	Experience report	Growing need for efficient development of software and deployment strategies	SA enables ways for continuous experimentation, deployments, and integration	most investigated best practices based on MSA containerized systems	Increasing Dependency & maintenance issues	S-68