Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Legacy systems to cloud migration: A review from the architectural perspective[☆]

Muhammad Hafiz Hasan [*], Mohd Hafeez Osman, Novia Indriaty Admodisastro, Muhamad Sufri Muhammad

Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia

ARTICLE INFO

Article history:
Received 15 June 2022
Received in revised form 26 February 2023
Accepted 6 April 2023
Available online 17 April 2023

Keywords:
Monolith
Cloud migration
Software architecture
Systematic review

ABSTRACT

Legacy systems are business-critical systems that hold the organization's core business functions developed in a traditional way using monolith architecture and usually deployed on-premises. Through time, this system is exposed to improvement changes, increasing its size and number of functionalities, thus increasing its complexity, and maintaining it becomes a disadvantage to the organization. Migration to the cloud environment becomes the primary option to improve legacy application agility, maintainability, and flexibility. However, to take advantage of the cloud environment, monolith legacy application needs to be rearchitected as microservice architecture to fully benefit from cloud advantages. This paper aims to understand the motivation for cloud migration, investigate existing cloud migration frameworks, identify the target architecture for the cloud, and establish any empirical quality issues in cloud migration from the implementation point of view. To achieve those objectives, we conducted a systematic literature review (SLR) of 47 selected studies from the most relevant scientific digital libraries covering pre-migration, migration, and post-migration stages. The SLR outcome provided us with the primary motivation for the cloud migration, existing cloud migration frameworks, targeted migration architecture patterns, and migration challenges. The results also highlight areas where more research is needed and suggest future research in this field. Furthermore, our analysis shows that current migration approaches lack quality consideration, thus contributing to post-migration quality concerns.

1. Introduction

Software modernization is a challenging and complex exercise (Knoche and Hasselbring, 2018). It involves multi-domain considerations such as business aspect, technical capability, economic features, and organizational culture (Efremovska and Lago, 2017; Pianini and Neri, 2021). A typical legacy software involves an organization's core business process. Thus, migrating legacy software to a new environment such as the cloud is risky and not straightforward. However, this transformation is needed from the business perspective as organizations and businesses see cloud platforms as promising future business strategies to remain competitive. They could utilize cloud benefits such as scalability, maintainability, reliability, and interoperability for business continuity and advancement.

A legacy system is a mission-critical system that supports the core business process of the organization with the restriction of outdated hardware, software, and resources to maintain. These systems often have monolithic software architecture, which does not consider modularity as a design principle (Megargel et al., 2020), and the systems work in silos (Ganesan and Chithralekha, 2016). Due to the years of usage, development, and improvement, legacy application's size and complexity are increasing, making the disadvantages of legacy monolith architecture outweigh its advantages (Kazanavicius and Mazeika, 2019). From the business perspective, existing core functionality systems developed using the latest technologies are defined as legacy systems if the system does not satisfy new business requirements and is unfit for the future needs of the organizations (Khadka et al., 2012).

Researchers have proposed various legacy to cloud migration frameworks. Approaches such as industrial domain-driven (Megargel et al., 2020), architecture-driven modernization (Grieger et al., 2016), horseshoe reengineering model (Marquez et al., 2015), and dependency graph (Fritzsch et al., 2018) help the migration execution process. However, from our initial observation, end-to-end migration frameworks are still limited (Yadav

[☆] Editor: Laurence Duchien.
[*] Corresponding author.
E-mail addresses: hafiz.hasan@student.upm.edu.my (M.H. Hasan), hafeez@upm.edu.my (M.H. Osman), novia@upm.edu.my (N.I. Admodisastro), sufry@upm.edu.my (M.S. Muhammad).

https://doi.org/10.1016/j.jss.2023.111702
0164-1212/© 2023 Elsevier Inc. All rights reserved.

et al., 2020), and little work has been done to evaluate them. For this reason, it is difficult to assess the quality of these migration frameworks. Thus, we initiate a systematic literature review (SLR) to further investigate this area from a quality perspective.

This paper reviews the current legacy to cloud migration approaches encompassing *pre-migration*, *migration*, and *post-migration* stages (Althani and Khaddaj, 2017). We aim to understand the motivation behind the cloud migration for legacy applications, identify and evaluate existing legacy to cloud migration framework and targeted cloud architecture patterns, and then derive *post-migration* quality issues. We have conducted an SLR to provide a balanced and objective summary primarily concerned with discovering a particular need for information related to research objectives using various techniques as proposed by Kitchenham and Charters (2007). We also consolidate legacy to cloud migration motivations from previous studies, a list of available cloud migration frameworks and architecture patterns, common cloud migration issues, and *post-migration* quality problems. This work benefits researchers as it helps to identify any limitations with cloud migration frameworks and identifies areas worth further investigation.

The remainder of this paper is structured as follows. Section 2 explains related works in this field. Section 3 describes the research methodology applied in conducting this research. Section 4 presents the research methodology's findings and answers research questions. Section 5 discusses the main findings identified and the research analysis. Section 6 addresses threats to our SLR's validity, and finally, Section 7 summarizes and concludes the research findings.

2. Related work

A review of existing legacy system modernization frameworks has been conducted by Jamshidi et al. (2013). Jamshidi et al. (2013) systematically reviewed 23 primary studies on cloud migration framework by classifying 12 framework characteristics categorized into four themes: migration maturity level, migration characterization, migration support, and migration constraints. The authors introduced a conceptual model called *Cloud-Reference Migration Model* (cloud-RMM) to present distinct migration processes and crosscutting concerns. The author's findings indicate that while cloud migration research has progressed and is becoming more established, there is still a need for a comprehensive, systematic framework to guide the migration process. Additionally, there is a lack of tools available to facilitate cloud migration. These findings are also supported by Rai et al. (2015), Rana and Rahman (2018) and Shuaib et al. (2019) in their works. Meanwhile, the crosscutting concerns and architecture adaptation are the least popular tasks, thus representing the potential future research directions.

Another systematic review of cloud migration research consisting of 23 primary studies done by Rai et al. (2015) is the most similar work to ours. However, it focuses on a different issue which is security. The authors proposed a conceptual model known as the *5-Phased Cloud Migration Model* inspired by Waterfall Model to classify and categorize the migration process. The most reported critical drivers for cloud migration are cost-saving, resource optimization, scalability, and maintainability. However, only business and technical factors are being considered for the cloud migration process challenges while overlooking crosscutting concerns such as quality factors.

Alkhalil et al. (2016) brought a different perspective on cloud migration around the decision-making process for migration. The authors reviewed the current level of support to aid the decision-making process offered by existing migration approaches and identified areas that require decision-making support. The review revealed that the level of support offered is insufficient to aid decision-makers in making informed decisions regarding cloud migration. Those decisions are made without considering the whole migration steps and tasks, resulting in higher risks of failure. New approaches should pay specific attention to the tasks during the design phase to assist in decision-making and migration planning.

Existing legacy system modernization strategies for the cloud have been discussed by Althani and Khaddaj (2018) from the quality aspects of the migration. Based on the systematic literature review, the authors proposed integrating quality into the migration process, considering the impact on migration cost and risk. Based on this concern, cloud migration strategies can be organized into three types of migration, i.e., *complete migration*, *incremental migration,* and *partial migration*. In *complete migration,* developers redeploy the system from scratch with defined migration goals and run the modernized system on new hardware using modern technology, platform, or architecture. The migration is completed when the target system is fully prepared to decommission the legacy system.

In comparison, the *incremental migration* type applied a reengineering approach through continuous improvement with incremental modifications. Based on the Horseshoe model, an *Architecture-Driven Modernization* method (ADM) transforms legacy applications by focusing on interoperability between related domains. Finally, *partial migration* involves some parts or components of the application to specific cloud-based services. This type of migration focuses on integrating legacy systems with provided cloud services. Althani and Khaddaj also suggested advancing modern technology to increase agility, reduce risks and enable organizations to benefit from new software technologies and paradigms. Besides, quality is a critical component that influences the choice of migration strategy based on business value or existing system quality.

Rana and Rahman (2018) emphasized the importance of appropriate cloud service model selection as a vital consideration for cloud migration. They suggested a hybrid approach for legacy application components that are hard to shift. As for other potential concerns, some consideration needs close attention, such as cloud vendor lock-in, restriction of legacy software licencing, cloud security, and cloud provider's SLA.

Shuaib et al. (2019) elaborated on the organization's challenges and strategies in adopting the cloud. Authors have identified several factors that affect cloud migration, including virtualization technology, interoperability, service level agreements (SLAs), and organizational readiness. At the same time, security and privacy have become the major hindrance to cloud adoption. In information system adoption, the *Technological, Organizational, and Environment* (TOE) is the most suitable adoption theory compared to the *Diffusion of Innovation* (DOI) theory. TOE covers the three primary concrete contexts of the author's work, consisting of *technological, organizational, economical,* and *external factors*.

Ponce et al. (2019) reported a rapid review of microservice architecture for legacy cloud migration. The authors analysed 20 migration techniques mainly applied to object-oriented software with design elements as inputs. The authors identified migration approaches as model-driven, static analysis, and dynamic analysis. These approaches can be used together in the migration without any problems.

Our work differs from Lenarduzzi et al. (2020), focusing on code quality and security (Marquez et al., 2015). We found out little previous works (Althani et al., 2017; Pigazzini et al., 2019; Sabiri et al., 2018) highlighted migration from architectural and reengineering quality aspects, while others discussed challenges (Ghofrani and Bozorgmehr, 2019) and high-level techniques (Fritzsch et al., 2018). We are interested in comparing post-migration quality concerns with migration quality drivers to

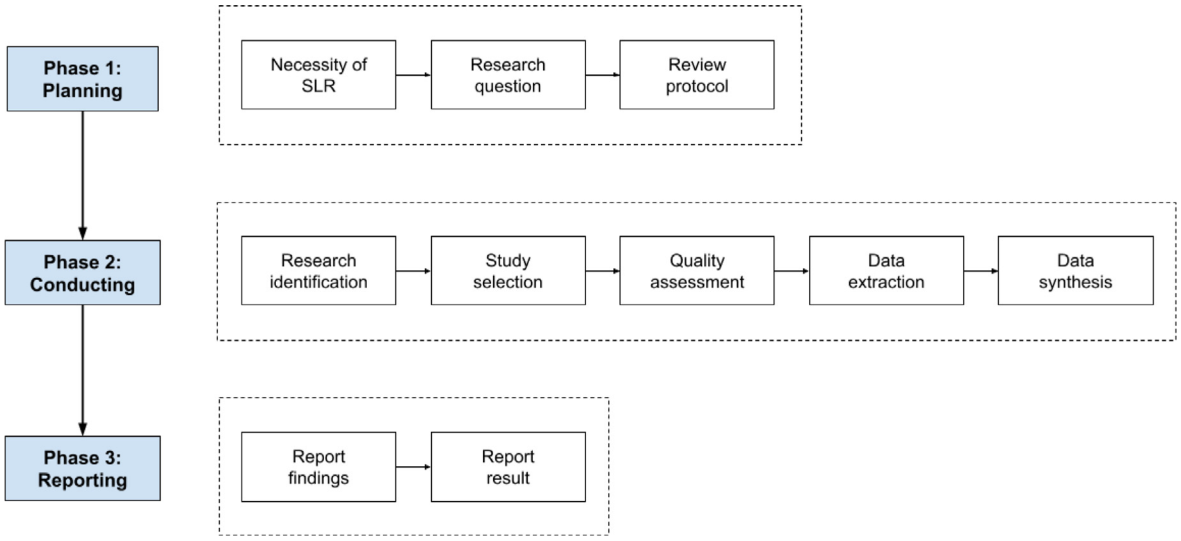

Fig. 1. Systematic literature review phases.

assess cloud migration benefits. We also gathered cloud migration challenges, trends, and techniques from architectural perspectives and showed targeted cloud architecture tendencies for the migration.

Vera-Rivera et al. (2021) carried out a systematic literature review on microservice architecture that focuses on the granularity of a microservice that directly affects application quality. They included metrics and quality attributes to determine the optimal microservice granularity. They also reported their microservices backlog approach, a genetic-programming technique that calculates proposed microservice granularity at design time (Vera-Rivera et al., 2020). Therefore, our work is complementary to their work. Instead of limiting purely to microservice architecture, our research is more cloud architecture-independent and generally emphasizes understanding the legacy to cloud migration. We also consider other product quality attributes, such as maintainability and reliability, from the software architecture quality perspective. Besides, this literature review used a different set of digital libraries as our source.

3. Research methodology

A systematic literature review is a method of reducing bias by following a structured process with defined steps. This type of review follows a three-phase process: planning, conducting, and documenting. It relies on a precise and thorough review protocol to extract, analyse, and document results in an organized and measured way (Kitchenham et al., 2009). Fig. 1 illustrates the overall review process. Additionally, we include critical reflections (Wieringa et al., 2006; Wohlin et al., 2013) into our research guideline during literature classification.

Conducted research phases and steps for our systematic literature review methodology are documented as follows.

3.1. Phase 1: Planning

Necessity of SLR

Over time, cloud computing evolves from the perspective of technological advancement and how people adopt it. Cloud technology improvement affects the growth and approach of migrating legacy applications to the cloud for the continuity of business and services (Pei Breivold, 2020). Therefore, besides findings on the cloud migration motivation from the architectural perspective, this SLR examines that migration achieved its main drivers. This research is designed to understand the quality of the migrated legacy application to the cloud architecture, cloud migration motivations, and cloud architecture trends that might lead to possible future research.

We establish the review concept in Table 1 to assist us in defining and evaluating the SLR review protocol. The study's general goal and scope have been formulated based on *population*, *intervention*, *comparison*, *outcome,* and *context* (PICOC) criteria (Petticrew and Roberts, 2008).

Research questions

Establishing the research questions is the most essential part of any literature review. This question helps to guide the entire literature review methodology comprised of searching related studies, data extraction, and data analysis process (Kitchenham and Charters, 2007). Based on the necessity of the review in Table 1, we constructed the research questions as shown in Table 2.

We started our research by understanding the motivations and drivers behind migrating legacy software to a cloud environment (RQ1). This question helps us understand the classification of factors from technical or business that contribute most as the drivers for migration. In RQ2, we would like to identify the existing approach and migration frameworks used or proposed by other researchers to guide the migration. We could then evaluate the strength and deficiencies of the existing approach. While migrating to the cloud involves re-architecting the legacy software, we are interested in identifying the target architectural pattern and its *state-of-the-art* characteristics (RQ3). Finally, in RQ4, we considered identifying common challenges faced during the migration and any quality concerns after the migration.

Review protocol

We specify the review protocol methods to reduce the possibility of researcher bias. We include the research questions and the review scope based on the objectives of formulating a search string for literature extraction. We listed five related keywords in our initial observation and established this research's inclusion and exclusion criteria.

Five digital databases that provide relevant and trustable resources have been selected as our data sources (Cavacini, 2015). These databases contain papers and publications related to software modernization and cloud computing that other researchers

Table 1
Definition of the general concept using PICOC.

Criterion	Scope description
Population	Legacy applications with monolith architecture pattern
Intervention	Migration drivers
Comparison	A holistic comparison among the population to analyse the collective impact of previous research on methods, findings, supporting tools, research coverage, and validation
Outcome	A classification, comparison, and characterization of proposed approaches with synthesized evidence to guide further research
Context	The academic, software industry, and all kinds of empirical studies

Table 2
Research questions.

Research question	Motivation
RQ1 — What are the main drivers to migrate legacy applications to the cloud?	The motivation and the need for cloud migration
RQ2 — How legacy to cloud migration has performed?	To identify and evaluate existing approaches and frameworks for cloud migration
RQ3 — What target architectural patterns are used to migrate legacy applications to the cloud?	To identify state-of-the-art of existing target architecture patterns
RQ4 — What are the common challenges during the migration and post-migration quality concerns?	To reveal the potential future directions in this area from a software quality perspective

Table 3
Inclusion and exclusion criteria.

Inclusion
1. In the context of migration from legacy or monolith to the cloud.
2. Studies that proposed solutions, evaluation, experience, opinion, and review of legacy to cloud migration.
3. Must provide the answer to at least one of the literature questions.
4. Published between 2015 and 2022.
5. Include maintenance or maintainability criteria as motivation or challenges during the migration.
6. Related to computer science, software engineering, cloud computing, and software architecture discipline.

Exclusion
1. Not written in English.
2. Studies are not available in full text.
3. Non-peer-reviewed studies or white papers.
4. Explicitly discuss on specific product or solution.

in the software engineering domain can refer to and use. In addition, other search engines such as Google Scholar have also been utilized in parallel for additional info gatherings. The selected digital data sources are *IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library,* and *Scopus*.

3.2. Phase 2: Conducting

Conducting is the second phase of the systematic review process, starting with research identification, literature selection and assessment, data extraction and synthesizing. The overview of the research identification process, study selection and assessment are illustrated in Fig. 2. We have formulated a search string based on identified keywords below. Besides implying word similarity checks to our main keywords to identify terminology diversification, we also utilized search strategies such as *Boolean* operators' manipulation to combine or exclude our specified criteria, quotation marks to get exact words or phrases, and wildcard characters to cover words that have spelling variations.

We applied our keywords search criteria to the title, abstract, and author keywords fields, thus resulting in a total of 10 502 papers from five databases. We initiate our study selection process by eliminating duplication. We further refined irrelevant studies by assessing them against inclusion and exclusion criteria (see Table 3). The assessment process includes evaluating titles and abstracts of the literature published between 2015 and 2022 with 571 studies. We then finalize our selection by repeating the previous step, focusing on full-text reading, and excluding all studies unrelated to migration and maintainability.

Forty-seven studies have been selected (refer to Appendix A) from 10 502 literature following the designated review protocol. Next, we present the distribution of selected studies according to digital libraries, as shown in Fig. 3. According to Petersen et al. (2015), the existing categorization scheme should be used for topic-specific categorization. Thus, we decided to categorize our selected literature research type into five categories according to systematic mapping classifications proposed by Wieringa et al. (2006). This classification schema was introduced for requirements engineering, but this scheme may be used for a broader field (Wohlin et al., 2013). We assumed it might be particularly interesting to compare studies using similar classifications to uncover some insights from previous works. We also replaced *philosophical* classification with *review* classification and combined *validation* and *evaluation* classification to adapt to our work mapping structure (see Table 4). It is important to highlight that some studies [P1][P7][P15][P17][P18][P21][P30][P33][P35][P37][P38][P41] falls under multiple category types.

As indicated in Table 5, most of these studies have been published in two service and cloud computing conferences and one software engineering and software architecture conference, namely ICSOC, CLOSER, ICSA, and ECSA. Among the 47 included studies, seven of them are published in the following software engineering journals JSSOFTWARE (impact factor: 4.476), INFSOF (impact factor: 4.769), SICS (impact factor: 2.108), and INFOSYS (impact factor: 3.648). The publication channel for each of the studies included in the list is described in Appendix D.

We further quantitatively synthesize our review data from the legacy systems migration perspective. Migration from legacy

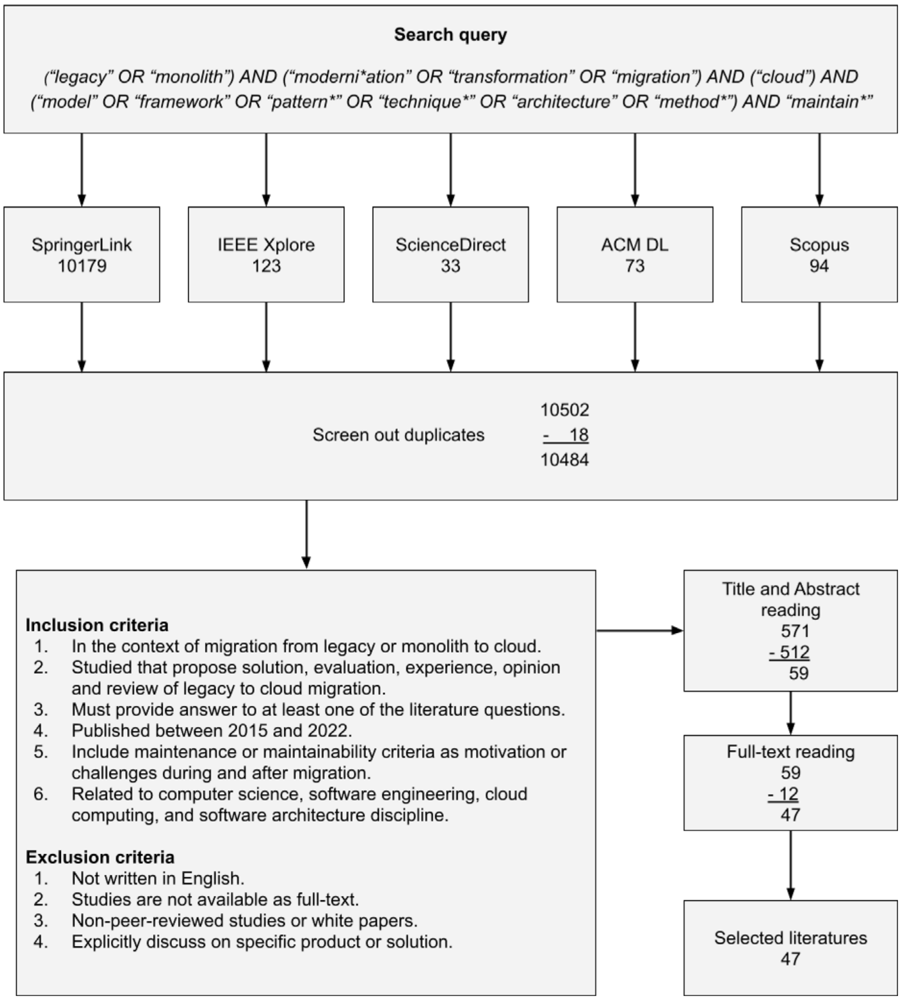
Search query
("legacy" OR "monolith") AND ("moderni*ation" OR "transformation" OR "migration") AND ("cloud") AND ("model" OR "framework" OR "pattern*" OR "technique*" OR "architecture" OR "method*") AND "maintain*"
SpringerLink 10179
IEEE Xplore 123
ScienceDirect 33
ACM DL 73
Scopus 94
Screen out duplicates 10502 - 18 10484
Inclusion criteria
1. In the context of migration from legacy or monolith to cloud.
2. Studied that propose solution, experience, evaluation, opinion and review of legacy to cloud migration.
3. Must provide answer to at least one of the literature questions.
4. Published between 2015 and 2022.
5. Include maintenance or maintainability criteria as motivation or challenges during and after migration.
6. Related to computer science, software engineering, cloud computing, and software architecture discipline.
Exclusion criteria
1. Not written in English.
2. Studies are not available as full-text.
3. Non-peer-reviewed studies or white papers.
4. Explicitly discuss on specific product or solution.
Title and Abstract reading 571 - 512 59
Full-text reading 59 - 12 47
Selected literatures 47

Fig. 2. Research identification process overview.

Table 4
Research type classification.

Type	Paper ID	Total
Solution	**P1**, P3, P5, P6, **P7**, P9, **P17**, **P18**, **P21**, P22, P27, **P30**, P31, P32, **P33**, P34, **P35**, **P37**, **P38**, P39, P40, **P41**, P42, P44, P46, P47	26
Review	**P1**, P4, P8, P10, P12, **P15**, P16, P19, P20, P23, P25, P36, P43, P45	14
Evaluation	**P7**, P13, **P15**, **P17**, **P18**, **P21**, P28, **P30**, **P33**, **P35**, **P37**, **P38**, **P41**	13
Opinion	P11, P14, P29	3
Experience	P2, P24, P26	3

Table 5
Study distribution by publication channel.

Publication channel	Abbreviation	Count	Category	Community	Paper ID
International Conference on Service-Oriented Computing	ICSOC	5	Conferences	Service/Cloud	P1, P8, P19, P20, P29
International Conference on Cloud Computing and Services Science	CLOSER	4	Conferences	Service/Cloud	P4, P10, P27, P39
International Conference on Software Architecture Companion	ICSA	4	Conferences	Software Engineering	P16, P24, P36, P43
European Conference on Software Architecture	ECSA	2	Conferences	Software Architecture	P32, P34

to the cloud involves three phases (Althani and Khaddaj, 2017; Gholami et al., 2017; Jamshidi et al., 2013). It consists of *migration planning, migration execution,* and *post-migration evaluation.* Detailed coverage for each study according to migration phases is described in Appendix B.

4. Findings

This section presents our findings on SLR according to the review protocol presented in Section 3. We report the findings, thus answering defined research questions.

Table 6
Legacy to cloud migration framework.

Paper ID	Framework
P3	System Migration Life Cycle (SMLC)
P6	Cloud Restriction Solver (CRS)
P18	Situational Method Engineering (SME)
P21	*FitScale*
P32	Secure Legacy Information System Migration to the Cloud (SMiLe2Cloud)
P35	*CloudMapper*
P38	Phase-wise Migration of Multiple Legacy Applications
P40	Smart Use-case Architecture Driven Modernization (ADM)
P46	Legacy Application Migration Process Framework to Cloud (LAMP2C)

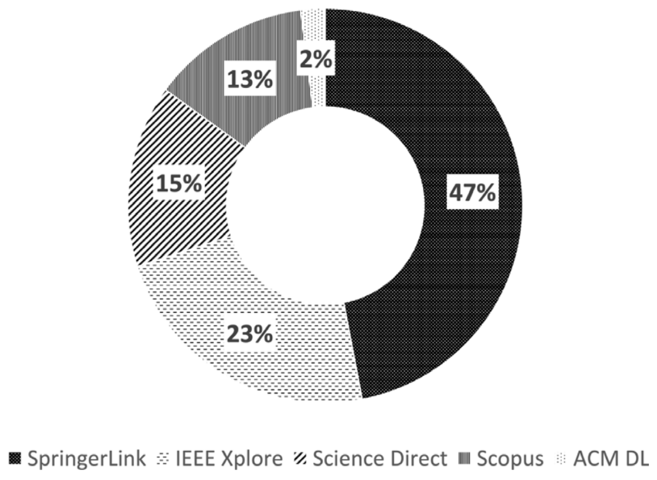

Fig. 3. Distribution of literature according to digital libraries.

4.1. RQ1: Main drivers in migrating legacy applications to the cloud

To answer this question, we present our findings regarding the motivation of the legacy to cloud migration in Fig. 4. From a product quality perspective, maintainability, scalability, and flexibility are the main drivers that motivate legacy to cloud migration. Twenty-eight studies reported maintainability as their main driver for migrating legacy applications to the cloud, followed by scalability 24 studies, flexibility 12 studies, availability 10 studies, manageability, resiliency, and reliability 4 studies respectively, cost 3 studies, and security 2 studies. According to Fehling et al. (2014), cloud-native ***I****solated state,* ***D****istribution,* ***E****lasticity,* ***A****utomated management, and* ***L****oose coupling* (IDEAL) properties are the main attraction for cloud migration for the legacy application. Those characteristics can only be achieved in a cloud environment, which is essential for legacy applications to fit future needs and satisfy new business requirements (Khadka et al., 2012).

4.2. RQ2: Available legacy to cloud migration framework

As mentioned by Althani and Khaddaj (2017), Gholami et al. (2017) and Jamshidi et al. (2013), a complete cycle of legacy to cloud migration involves three main phases *i. planning, ii. execution, and iii. evaluation*. To answer RQ2, we selected the proposed approach from the studies if it involves all those three phases in their approach. We present our findings regarding the available legacy to cloud migration framework in Table 6.

Regarding migration phase coverage, we found that most studies focus on the planning phase [P1][P4] [P21][P31][P32][P36], involving a monolith decomposition approach, service identification, and defining context boundary. As for the migration execution phase, only a high-level explanation has been described [P13][P24][P29], while limited studies include post-migration phase discussion [P10][P28][P44]. For the architecture reengineering process during the migration, the proposed frameworks refer to Architecture Driven Modernization (ADM) approach and Horseshoe Model (Kazman et al., 1998) [P15][P18][P30][P31][P32] methodology as architecture transformation guideline.

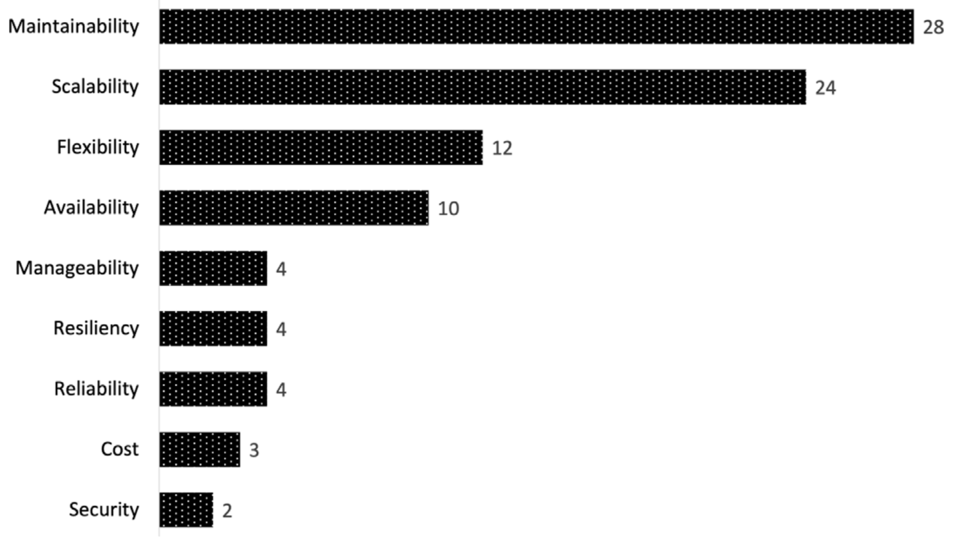

Fig. 4. Motivation to migrate legacy applications to the cloud.

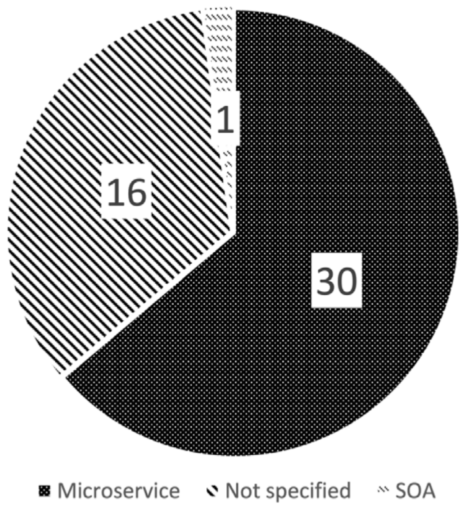

Fig. 5. Target architectural pattern for migration to the cloud.

4.3. RQ3: Migration target architectural pattern

RQ3 aims to gather available cloud architectural patterns and further understand the *state-of-the-art* of selected target architectural patterns on the cloud. Among 47 selected studies, only 31 studies stated targeted architecture, while the rest did not specify their target architecture on the cloud. This happened due to the author's work that covers different spectrums of cloud migration, such as business perspective [P22][P32], generic approach [P3][P18], cloud provider view [P6], migration assessment [P4][P47] and feasibility study [P11], evaluation of previous approaches [P15][P16][P29], architecture refactoring and reengineering [P40], specific cloud benefits [P21][P35], or partly covered the migration phases [P31][P46]. Of the 31 mentioned architectures, 30 are microservices architecture, while the other architecture [P1] is SOA. We illustrate our findings in Fig. 5.

4.4. RQ4: Common challenges during migration and post-migration quality concerns

Althani and Khaddaj (2017) categorize the migration lifecycle into *pre-migration*, *migration*, and *post-migration* stages. *Pre-migration* is the core stage that includes migration definition, assessment of software architecture, and migration planning phase. Meanwhile, the *migration* stage aims to accomplish migration execution with three phases involving software redesign, development, and deployment. The evaluation and release phase in the *post-migration* stage focuses on the functionality, operations, quality, and software correctness based on migration objectives.

Answering migration execution challenges and whether they contribute to post-migration quality concerns, we present our findings for the top legacy to cloud migration challenges comprised of legacy service identification or service boundary identification, lack of migration guidelines, refactoring or rearchitecting legacy application as a cloud application, legacy code complexity, not updated or no documentation artefacts for legacy application, and difficulty in conducting quality testing (see Fig. 6).

Meanwhile, to synthesize post-migration quality concerns, we follow ISO/IEC 25010:2011 (Estdale and Georgiadou, 2018) in categorizing our findings of product quality attributes consisting of maintainability, performance, security, reliability, compatibility, and portability. The number of mentioned challenges are maintainability, 28; performance, 24; security, 14; reliability, 13; compatibility, 11; and portability, 9 (see Fig. 7). We further describe papers for related quality concerns in Appendix C.

5. Discussion

After conducting the research, we can draw a perspective concerning the driver to migrate legacy applications to the cloud, available migration frameworks to guide the migration, common cloud target architecture, migration challenges, and post-migration quality concerns. Our findings from Section 4 help us to understand the whole migration process from *pre-migration planning*, *migration execution*, and *post-migration evaluation* phase.

5.1. RQ1 analysis of cloud migration drivers

As we can observe from the findings in Section 4.1, most factors that influence the migration of legacy applications to the cloud are related to the roadblock or limitation of legacy applications (Knoche and Hasselbring, 2018) and the benefits of cloud-native IDEAL properties in improving modifiability, scalability, maintainability, and deployability (Fehling et al., 2014). Thus, in general, we can describe that cloud offers an environment for software modernization that ensures legacy applications are fit to satisfy new business requirements and future needs.

Legacy applications that are commonly running on obsolete hardware and old programming language are often exposed to

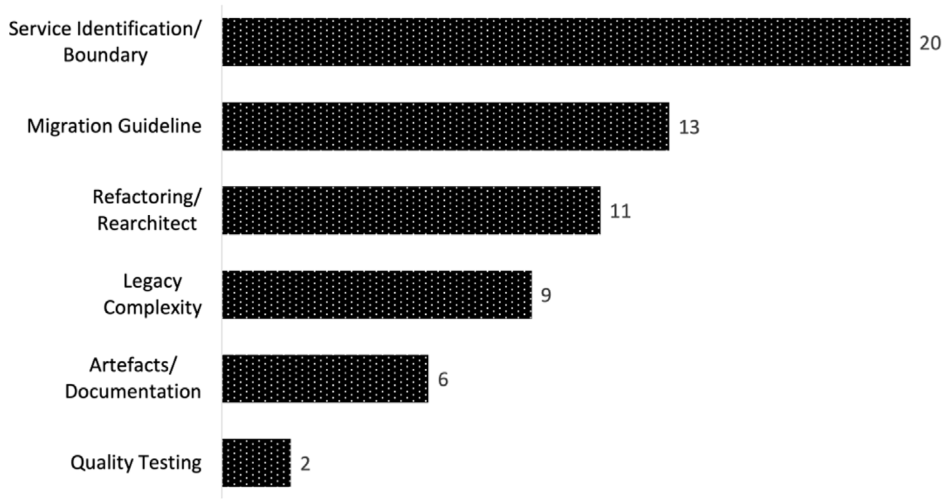

Fig. 6. Migration challenges.

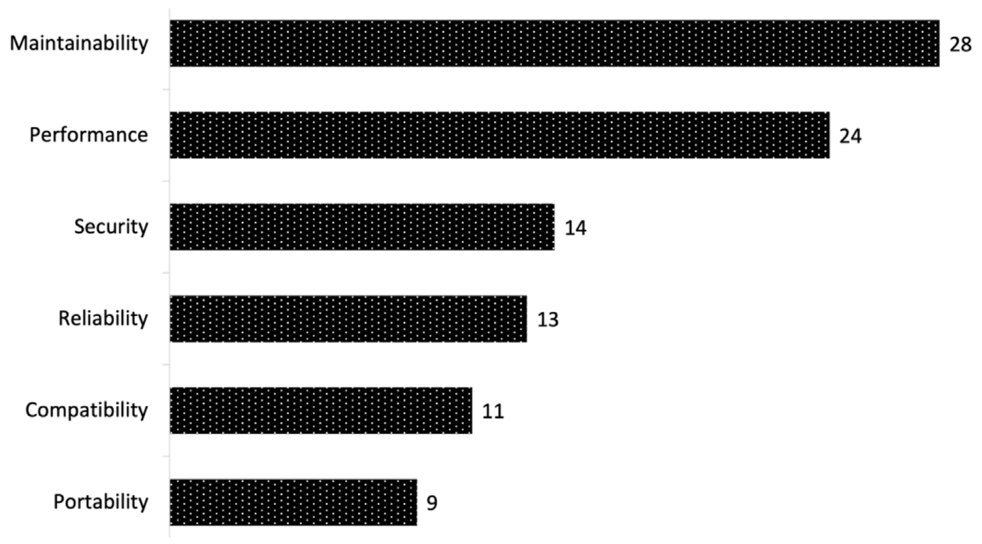

Fig. 7. Post-migration quality concerns.

degraded performance, limited support, and irreplaceable parts. Besides reducing operational costs and phasing out obsolete technology, migrating legacy applications to the cloud could eliminate infrastructure modernization overhead costs (Kazanavicius and Mazeika, 2019) and allow new opportunities for the company to offer a new business model such as Software-as-a-Service (SaaS) for their software (Khadka et al., 2015).

5.2. RQ2 analysis of cloud migration approach and framework

In this review, we observed that if a legacy application has a monolith architecture and does not consider modularity as a design principle (Megargel et al., 2020), the cloud migration strategy, such as rearchitecting (Clayton, 2018), should be performed. Therefore, the legacy application code should be modified, and the underlying architecture needs to be replaced based on the related migration approach (Jamshidi et al., 2015).

The identification of microservices from Object-Oriented applications was proposed by Selmadji et al. (2020). Author discussions are based on static analysis of legacy source code and software architect recommendations for identified services. However, decomposition results are not quantitatively assessed before migration has been executed.

Pigazzini et al. (2019) proposed service candidates based on the business domain model. The graph-based diagram has been generated to identify related legacy application features, implying a topic detection algorithm. However, the authors did not evaluate his proposed service candidate's quality, which might improve migration decisions and confidence.

The *System Migration Life Cycle* (SMLC) framework (Althani and Khaddaj, 2017) is a generic migration methodology focusing on technical and business alignment with quality aspects. However, as the authors focus on a generic methodology, the framework covers a high-level migration overview without discussing each migration phase. Therefore, there is a need for the author to refine quality attributes for each migration phase and extend his discussion on the quality evaluation approach.

An architecture-driven modernization (ADM) (Sabiri et al., 2018) approach is an iterative and incremental migration process based on a smart use-case. In this paper, the authors discuss how the architecture of legacy applications must be changed into cloud architecture and deployed. This process involves reverse engineering source code operation to generate Platform Specific Model (PSM) and transform it into the Platform Independent Model (PIM). PIM for legacy applications will then be transformed to PIM for the cloud and forward engineering it as PSM cloud before deploying into the cloud environment. However, there is a need for the author to validate generated PIM legacy source quality before transforming it to the PIM cloud to ensure that generated legacy artefact quality is acceptable.

A different approach has been proposed by Munisso and Chis (2017) for the *CloudMapper* framework. The authors proposed a model-driven framework by exploiting the intermediary layer as a service mapper for his cloud application. This approach focuses on cloud portability to prevent *"vendor lock-in"* problems due to applications tightly coupled to the underlying cloud provider. Our analysis for this approach is the need to evaluate quality-related impact by introducing an intermediary layer in the framework design.

5.3. RQ3 analysis of the targeted architectural pattern

As discussed in Section 4.3, microservices have been identified as the most targeted architectural pattern for legacy to cloud migration (see Fig. 5). We explored related cloud architectural patterns to understand the architectural pattern trends, as shown in Fig. 8. We plotted a bubble chart by mapping the cloud-targeted architecture's volume with the year of study. We noticed that microservices have become prominent in recent years. This trend shows that microservices have become the main architectural pattern for the cloud environment due to its characteristics that support cloud-native properties (Leymann et al., 2017). In addition, this increment was realized by virtue of the improvement of the cloud platform and cloud provider maturity (Kratzke, 2018), affecting business and industry's confidence in adopting cloud migration. Meanwhile, service-oriented architecture (SOA) remains the least software architectural pattern selected when migrating to the cloud.

5.4. RQ4 analysis of common challenges during migration and post-migration quality concerns

In Section 4.4, we have presented the challenges during the migration and post-migration quality concerns. Therefore, we enhance our analysis to observe whether legacy to cloud migration drivers (see Fig. 4) are satisfied with post-migration concerns (see Fig. 7). The observation is based on quality attribute perspectives.

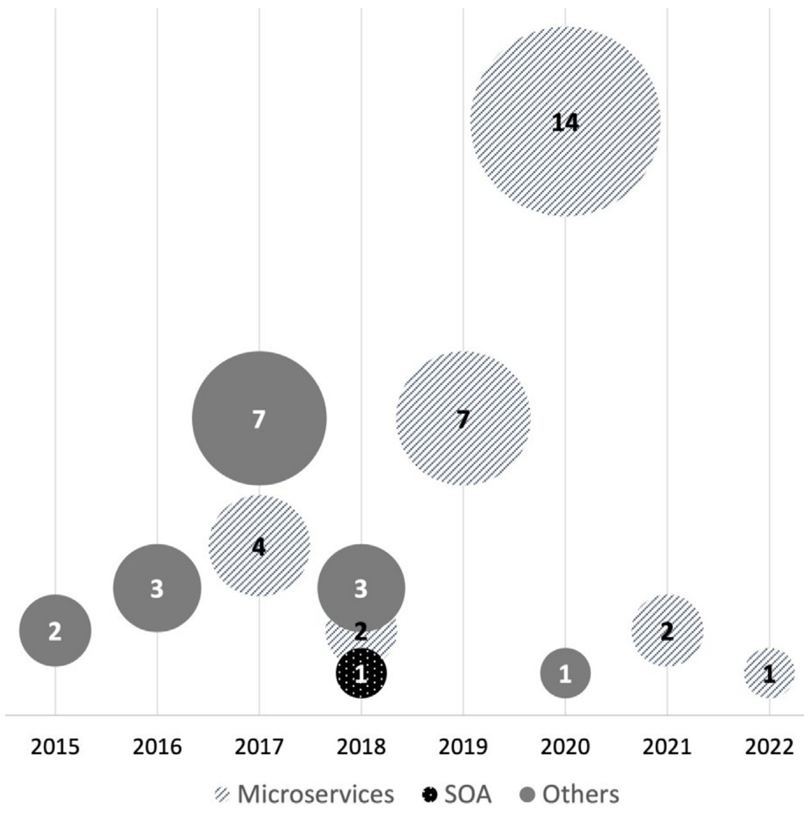

Fig. 8. Adoption of the cloud targeted architecture.

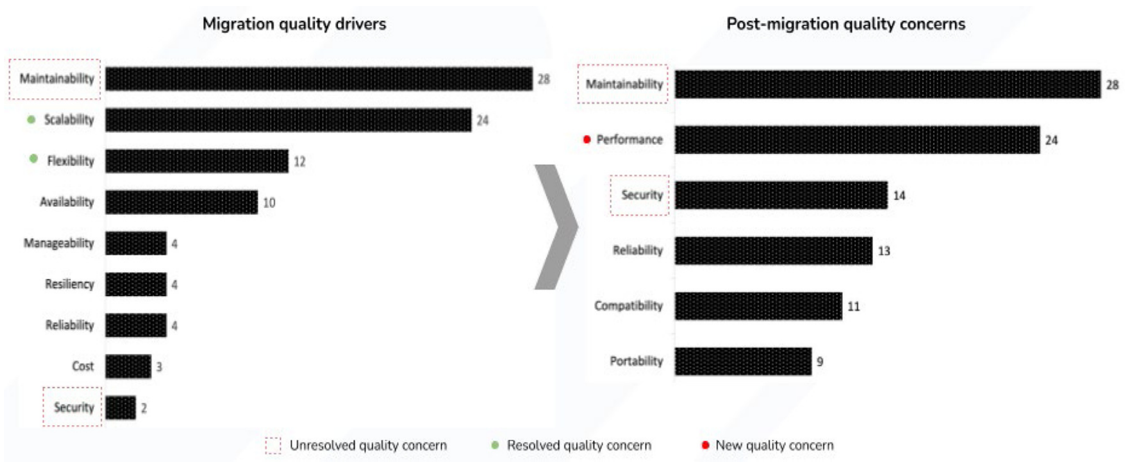

Fig. 9. Migration quality driver's relation vs. post-migration quality concerns.

We have identified the main motivations for the legacy system to cloud migration in RQ1. However, it is intriguing to investigate whether the migration fulfils its motivation. We correlate the issues and challenges from the quality perspective in the post-migration stage. Our analysis reveals that migration to the cloud satisfied other migration drivers, such as scalability, flexibility, manageability, and resiliency. Fig. 9 illustrates migration quality drivers and its relation with post-migration quality concerns. We defined three conditions consisting of (i) *unresolved*: where the quality concern still exists after the migration; (ii) *resolved*: where the migration solved the concern; and (iii) *new*: where the migration introduced new quality concerns. Unfortunately, migration to the cloud only accomplished scalability and flexibility motivations, while application maintainability quality remained unfixed. Security issue however, becomes more apparent than before the migration due to the dynamic cloud environment and rapidly evolving technologies (Patel et al., 2020; Shuaib et al., 2019). Application performance is visible as a new post-migration quality concern resulting from inadequately experienced system architect, incompatible migration strategy, opted application architecture, and difficulty in identifying microservice context boundary (Althani and Khaddaj, 2018; Sabiri et al., 2018; Shuaib et al., 2019).

Next, we present the top three quality concerns regarding motivation for migrating legacy applications to the cloud.

- *Maintainability* - From our previous findings in Section 3.1, maintainability becomes the main quality driver for cloud migration. However, post-migration quality concerns (RQ4)

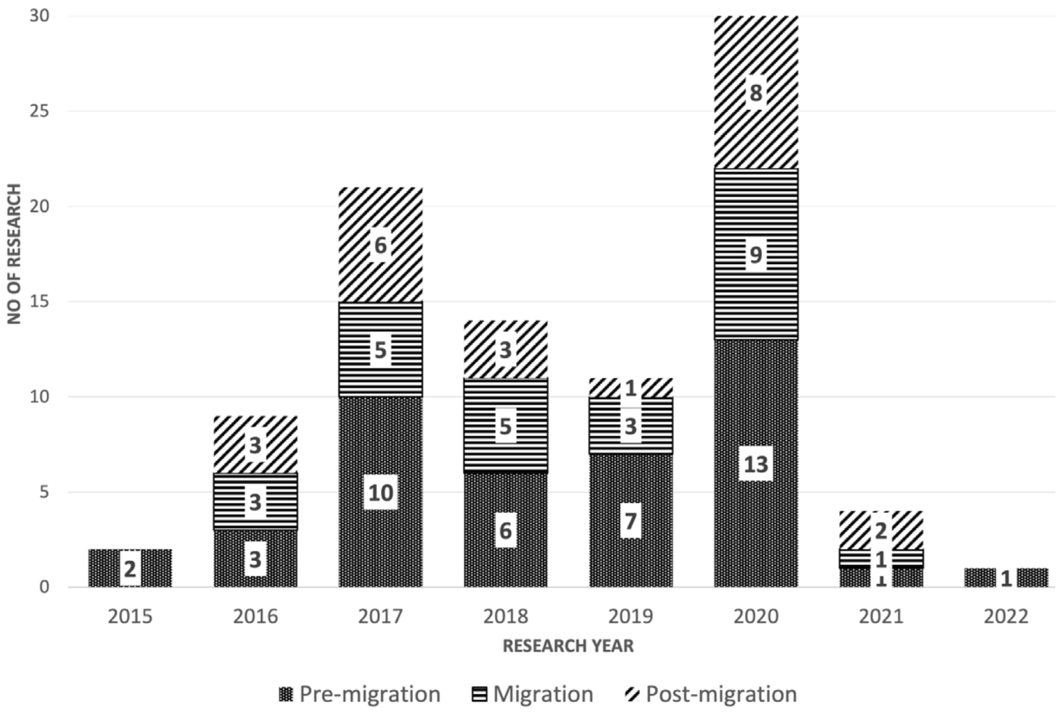

Fig. 10. Distribution of migration phases studies.

reported that maintainability still represents the most significant concern. This indicates that migrating to the cloud did not solve the software maintainability issue. We informed that quality metrics had received relatively limited attention in the existing migration approach. Similar findings were reported (Jamshidi et al., 2013) as crosscutting migration concerns.

- *Security —* Security is the most negligible factor motivating cloud migration (RQ1). Due to the hype on cloud computing's benefits and advantages (Amy Ann and Rob van der, 2016), cloud security received minimal attention during the pre-migration stage. Unfortunately, security concerns gain greater attention after the migration. Our analysis reveals two factors that are likely to contribute to this condition. First, a lack of understanding of the cloud provider's SLA limits client control on the security aspects of the cloud platform. As a result, cloud application security depends on services offered by cloud providers. Secondly, a legacy to cloud migration requires application architectural transformation. As a result, developers face new cloud architecture that requires new knowledge and perspective regarding security issues.
- *Performance —* Concern about performance only appears in the post-migration phase rather than as the driver for cloud migration. Previous work (Fritzsch et al., 2018) revealed insufficient structured standards for refactoring existing monolith applications. Consequently, poor architecture design due to an unsatisfactory migration strategy led to inefficient cloud application performance. Therefore, implementing a suitable application architecture design is crucial in a distributed cloud environment.

SMiLe2Cloud framework includes security concerns during migration execution (Marquez et al., 2015). Security attributes were implemented before the monolith's reverse engineering process. However, the authors did not consider multiple quality attributes during the process and analysed multi-attribute relations, interaction, and quality trade-offs.

Selection of ideal service granularity during decomposition and refactoring improved architecture complexity and maintainability (Fritzsch et al., 2018). The architectural quality of targeted migration patterns contributes to postponed migration activities (Lenarduzzi et al., 2020). During development, continuous quality monitoring approaches may decrease migration technical debt and increase code and architecture maintainability, reliability, and performance.

5.5. Other analysis

We plot the distribution of the cloud migration phases chart for our further exploration (see Fig. 10). From 2015 to 2022, the studies that included the post-migration phase received the least research coverage. This trend confirms our findings on limited attention given to the post-migration phase, comprising quality evaluation, thus contributing to reported migration quality issues. This finding indicates there is a grey area for researchers to explore.

Current research trends and the industry's adoption perspective (see Fig. 10) correspond with Moore's cloud adoption life cycle curve (Moore, 2014). During the 2015 to 2018 period, we describe the innovators and early adopters phase, which involves industry players aggressively pursuing the latest technologies to gain an early advantage and significant competitive lead in their field. Thus, research on cloud adoption and microservices interest started to arise before it slightly plummeted from 2018 to 2019 (see Fig. 10). This phenomenon is depicted as *Moore's Chasm —* where many newly introduced approaches have been proven too challenging to implement relative to their benefits (Moore, 2014). However, in 2020, a spike was recorded in this area's research. This confirms Moore's point of view that challenges faced by initial adopters have been slowly resolved in line with the maturity of cloud technology and infrastructures. As a result, others waiting companies will quickly join adopting it on a massive scale and being acceptable from the early majority group (see Fig. 11). Future research will focus on post-migration concerns to neutralize the current focus area, the pre-migration stage, as described in Appendix B.

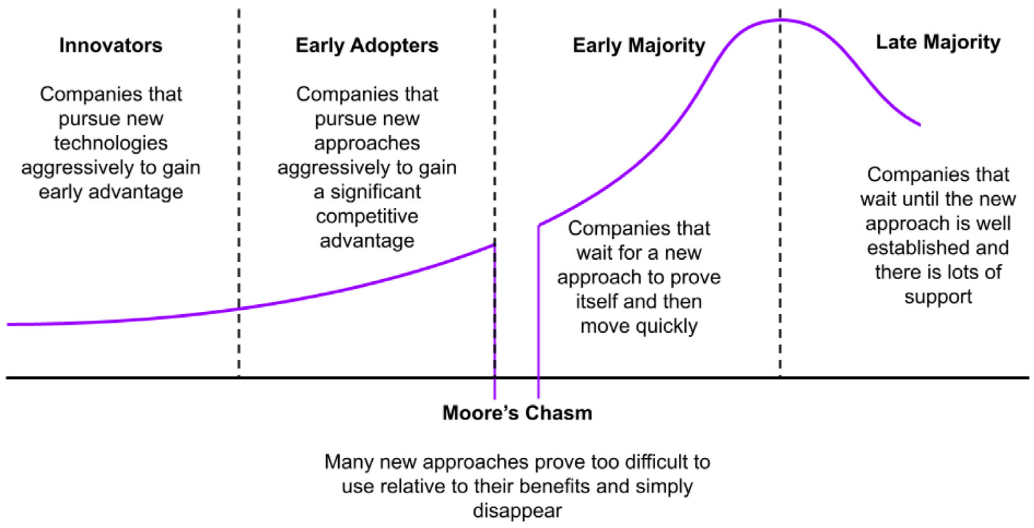
Innovators
Companies that pursue new technologies aggressively to gain early advantage
Early Adopters
Companies that pursue new approaches aggressively to gain a significant competitive advantage
Early Majority
Companies that wait for a new approach to prove itself and then move quickly
Late Majority
Companies that wait until the new approach is well established and there is lots of support
Moore's Chasm
Many new approaches prove too difficult to use relative to their benefits and simply disappear

Fig. 11. Moore's technology adoption life cycle curve (Moore, 2014).

6. Threats to validity

We adopted a well-defined systematic review for software engineering study as proposed by Kitchenham et al. (2009) with some deviations to match our approach as described in Section 3. Albeit the results and findings for this SLR are generally reliable, this review work cannot be excluded from possible threats to its validity. We map our SLR threats and validity as described by Zhou et al. (2016).

6.1. Construct validity

Our research questions may not be able to cover all related SLRs that describe our research domain completely. Thus, we conducted an initial study using related keywords identified from the research's primary objective to mitigate inappropriate research questions. This strategy helps reveal the current research state and sets a base context for this work so that pertinent research questions and review protocols have been constructed to avoid chaos in the research process.

6.2. Internal validity

We took several precautions to increase the internal validity of our work results. During the research identification process, our search strategy entirely depends on identified keywords to formulate the search query. As SLR involves numerous works from various authors, there are possibilities for them to use different terminologies on specific keywords. Therefore, we imply a word similarity check for each keyword to ensure no missing word. Our main goal is to capture as much available literature as possible to avoid favouritism. Since cloud migration relates to different domains, including information systems, software engineering, and networks, we combined related terms in our search string. This helps us to increase search results. Although we limit our search resources to five online library databases, those libraries are reliable enough to cover the most well-known and high-quality publications in the computer science field (Cavacini, 2015). In addition to our rigorous search strategy, the selected literature has gone multi-step selection process based on defined inclusion and exclusion criteria. These studies were then assessed through title and abstract reading followed by full-text reading. The selection and data extraction process relies heavily on the researcher's knowledge and understanding of the topic, guided by a defined review protocol. By referring to SLR's primary objective to understand the overview and current state of legacy migration to the cloud, researcher judgement is acceptable.

6.3. External validity

With the concern of generalizing the results, this study is sufficient to represent an overview of current research progress in this area. The research questions were designed in a way that would comprehensively answer common concerns regarding cloud migration. Besides, selected primary studies from the previous five years (2017 to 2022) are sufficient to produce general findings (Petticrew and Roberts, 2008).

6.4. Conclusion validity

Our analysis and findings from selected SLR are quantitatively based. Thus, we have constructed exclusion criteria in our review protocol to exclude primary study duplication in our work to ensure the correctness of the extracted data. However, reported analyses are subjected to our interpretation of the data. To mitigate bias and misclassification of these primary studies, each selected study has undergone an exhaustive reading process to establish understanding. Due to the quantitative nature of the data, this threat has already been reduced, which would be otherwise high in the case of qualitative data research.

7. Conclusion and future work

This study aims to understand the current state of legacy to cloud migration progress. We implemented an SLR approach in reviewing the literature to highlight possible research gaps. By analysing the collected data, we identified areas that have not yet been thoroughly studied, with a particular focus on maintainability and quality perspectives. Our contribution is consolidating existing drivers to migrate legacy applications to the cloud, identifying available cloud migration frameworks, *state-of-the-art* existing target architecture patterns, common migration issues, and post-migration quality concerns.

The results offer valuable insights for future work to explore. Our findings showed that maintainability, scalability, and flexibility are the main motivation for migrating legacy applications to the cloud. However, the existing legacy to the cloud migration framework focuses on the migration process with less attention to quality evaluation. Thus, post-migration's most reported quality issues are maintainability, performance, and security. Meanwhile, microservices are the most targeted architectural pattern for cloud migration to fully benefit from cloud advantage. Therefore, we believe that post-migration quality evaluation needs future exploration by researchers.

Future works include the development of a tool to facilitate the legacy to cloud migration process with consideration of quality concerns for migrated cloud architecture. We also plan to validate further the existing software reengineering approach to suit the legacy monolith to cloud architecture transformation approach and identify comparative measurement metrics for those architectures. Finally, we are also considering extending this work by proposing a quality-driven monolith legacy to cloud migration framework that covers a systematic, step-by-step, and comprehensive framework that is highly needed in this area of research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

Appendix A. List of selected literatures

See Table A.1.

Table A.1
List of selected literatures.

ID	Authors	Title	Library	Year
P1	Abdellatif, M. et al.	State of the Practice in Service Identification for SOA Migration in Industry	Springer Link	2018
P2	Alan, M., Venky, S., David, K.W.	Migrating from Monoliths to Cloud-Based Microservices: A Banking Industry Example	Springer Link	2020
P3	Althani, B., Khaddaj, S.	The Applicability of System Migration Life Cycle (SMLC) Framework	IEEE Xplore	2017
P4	Anastasija, E., Patricia, L.	From Legacy to Cloud: Risks and Benefits in Software Cloud Migration	Science Direct	2017
P5	Borges, M., Barros, E., Maia, P.	Cloud restriction solver: A refactoring-based approach to migrate applications to the cloud	Science Direct	2018
P6	Bucchiarone, A., Soysal, K., Guidi, C.	A Model-Driven Approach Towards Automatic Migration to Microservices	Springer Link	2020
P7	Carvalho, L., Garcia, A., Assuncao, W.	Extraction of configurable and reusable microservices from legacy systems: An exploratory study	Scopus	2019
P8	Christoforou, A. et al.	Supporting the Decision of Migrating to Microservices Through Multi-layer Fuzzy Cognitive Maps	Springer Link	2017
P9	Dragoni, N., et al.	Microservices: Yesterday, Today, and Tomorrow	Springer Link	2017
P10	Fowley, F. et al.	The Benefits of Using Experimental Exploration for Cloud Migration Analysis and Planning	Springer Link	2018
P11	Fritzsch, J. et al.	From Monolith to Microservices: A Classification of Refactoring Approaches	Springer Link	2019
P12	Fritzsch, J. et al.	Microservices Migration in Industry: Intentions, Strategies, and Challenges	IEEE Xplore	2019
P13	Ghofrani, J., Bozorgmehr, A.	Migration to Microservices: Barriers and Solutions	Springer Link	2019
P14	Gholami, M.F et al.	Cloud migration process — A survey, evaluation framework, and open challenges	Science Direct	2016
P15	Gholami, M.F et al.	Key challenges during legacy software system migration to cloud computing platforms — an empirical study	Science Direct	2017
P16	Gomes Barbosa, M., Maia, P.	Towards Identifying Microservice Candidates from Business Rules Implemented in Stored Procedures	Scopus	2020
P17	Grieger, M. et al.	Concept-Based Engineering of Situation-Specific Migration Methods	Springer Link	2016
P18	Henry, A., Ridene, Y.	Assessing Your Microservice Migration	Springer Link	2020
P19	Hwang, J., Vukovic, M., Anerousis, N.	Fitscale: Scalability of legacy applications through migration to cloud	Scopus	2016
P20	Jamshidi, P. et al.	Cloud Migration Patterns: A Multi-cloud Service Architecture Perspective	Springer Link	2015
P21	Kazanavicius, J., Mazeika, D.	Migrating Legacy Software to Microservices Architecture	IEEE Xplore	2019
P22	Knoche, H., Hasselbring, W.	Using Microservices for Legacy Software Modernization	IEEE Xplore	2018
P23	Kratze, N., Quint, P.	Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study	Science Direct	2017
P24	Krause, A. et al.	Microservice Decomposition via Static and Dynamic Analysis of the Monolith	IEEE Xplore	2020
P25	Lauretis, L.D	From Monolithic Architecture to Microservices Architecture	IEEE Xplore	2019
P26	Lenarduzzi, V. et al.	Does migrating a monolithic system to microservices decrease the technical debt?	Science Direct	2020

(*continued on next page*)

Table A.1 (*continued*).

ID	Authors	Title	Library	Year
P27	Leyman, F. et al.	Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation	Springer Link	2017
P28	Lichtenthaler, R. et al.	Requirements for a model-driven cloud-native migration of monolithic web-based applications	Springer Link	2020
P29	López-Sanz, M., et al.	Modernization of Information Systems at Red.es: An Approach Based on Gap Analysis and ADM	Springer Link	2017
P30	Luis, M. et al.	A Framework for Secure Migration Processes of Legacy Systems to the Cloud	Springer Link	2015
P31	Maisto, S.A., Di Martino, B., Nacchia, S.	From Monolith to Cloud Architecture Using Semi-automated Microservices Modernization	Scopus	2020
P32	Mishra, M., Kunde, S., Nambiar, M.	Cracking the Monolith: Challenges in Data Transitioning to Cloud Native Architectures	ACM Digital Library	2018
P33	Munisso, R., Chris, A.	CloudMapper: A Model-based Framework for Portability of Cloud Applications Consuming PaaS Services	IEEE Xplore	2017
P34	Pigazzini, I., Fontana, F.A., Maggioni, A.	Tool Support for the Migration to Microservices Architecture: An Industrial Case Study	Springer Link	2019
P35	Sabiri, K., Benabbou, F., Khammal, A.	Model Driven Modernization and Cloud Migration Framework with Smart Use Case	Springer Link	2018
P36	Santos, N., Silva, A.R.	A Complexity Metric for Microservices Architecture Migration	IEEE Xplore	2020
P37	Taibi, D., et al.	Microservices Anti-patterns: A Taxonomy	Springer Link	2020
P38	Taibi, D., Lenarduzzi, V., Pahl, C.	Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation	IEEE Xplore	2017
P39	Taibi, D., Systa, K.	A Decomposition and Metric-Based Evaluation Framework for Microservices	Springer Link	2020
P40	Valdivia, J.A. et al.	Patterns Related to Microservice Architecture: a Multivocal Literature Review	Springer Link	2020
P41	Yadav, S.K., Khare, A., Kavita, C.	ASK Approach: A Pre-migration Approach for Legacy Application Migration to Cloud	Springer Link	2020
P42	Zimmermann, O.	Architectural refactoring for the cloud: a decision-centric view on cloud migration	Springer Link	2017
P43	Romani et al.	Towards Migrating Legacy Software Systems to Microservice-based Architectures: A Data-Centric Process for Microservice Identification	IEEE Xplore	2022
P44	Guo & Wu	A Review of Bad Smells in Cloud-based Applications and Microservices	IEEE Xplore	2021
P45	Punnoose & De	Phase-wise migration of multiple legacy applications — A graph-theoretic approach	Science Direct	2021
P46	Pei Breivold	Towards factories of the future: migration of industrial legacy automation systems in the cloud computing and Internet-of-things context	Scopus	2020
P47	Bajaj et al.	Partial Migration for Re-architecting a Cloud Native Monolithic Application into Microservices and FaaS	Scopus	2020

Appendix B. Studies by migration phases

See Table B.1.

Table B.1
Studies by migration phases.

Paper ID	Year	Migration planning	Migration execution	Post-migration evaluation
P1	2018	x		
P2	2020	x	x	x
P3	2017	x	x	x
P4	2017	x		
P5	2020		x	
P6	2018	x	x	x
P7	2020	x	x	
P8	2019	x	x	x
P9	2017	x		
P10	2017			x
P11	2018	x	x	x
P12	2019	x		
P13	2019	x	x	
P14	2019	x		
P15	2016	x	x	x
P16	2017	x	x	x
P17	2020	x	x	
P18	2016	x	x	x
P19	2021			x
P20	2020	x		
P21	2016	x	x	x
P22	2015	x		
P23	2019	x		
P24	2018	x	x	
P25	2017	x		
P26	2020	x		
P27	2019	x	x	
P28	2020			x
P29	2017	x	x	x
P30	2020	x	x	
P31	2017	x		
P32	2015	x		
P33	2020	x		
P34	2018	x	x	x
P35	2017	x	x	x
P36	2020	x	x	x
P37	2019	x		
P38	2021	x	x	x
P39	2022	x		
P40	2018	x	x	
P41	2020	x	x	
P42	2020	x	x	x
P43	2017	x	x	x
P44	2020	x		x
P45	2020	x	x	x
P46	2020	x		
P47	2017	x		

Table C.1
Studies by quality concerns.

Quality concern	Paper ID
Maintainability	P1, P3, P7, P9, P10, P11, P12, P13, P14, P15, P16, P17, P18, P23, P24, P29, P30, P31, P32, P33, P37, P38, P40, P42, P43, P44, P45, P47
Performance	P3, P5, P6, P7, P8, P12, P13, P14, P15, P16, P18, P23, P26, P30, P31, P36, P37, P43, P44, P46
Security	P2, P4, P11, P14, P15, P19, P21, P24, P25, P26, P29, P33, P36, P43
Reliability	P6, P9, P12, P13, P14, P15, P17, P24, P26, P30, P33, P36, P46
Compatibility	P3, P12, P14, P24, P25, P26, P31, P32, P38, P43, P45
Portability	P1, P11, P21, P29, P31, P32, P39, P43, P47

Appendix C. Studies by quality concerns

See Table C.1.

Appendix D. Study distribution by publication channel

See Table D.1.

Table D.1
Study distribution by publication channel.

Publication channel	Abbreviation	Count	Category	Community	Paper ID
International Conference on Service-Oriented Computing	ICSOC	5	Conferences	Service/Cloud	P1, P8, P19, P20, P29
International Conference on Cloud Computing and Services Science	CLOSER	4		Service/Cloud	P4, P10, P27, P39
International Conference on Software Architecture Companion	ICSA	4		Software Engineering	P16, P24, P36, P43
European Conference on Software Architecture	ECSA	2		Software Architecture	P32, P34
International Conference on Software Maintenance and Evolution	ICSME	1		Software Engineering	P12
International Conference on Applied Informatics	ICAI	1		Software Engineering	P13
International Conference on Software Reuse	ICSR	1		Software Engineering	P17
Open Conference of Electrical, Electronic and Information Sciences	eStream	1		Information System	P21
International Conference on Advanced Information Systems Engineering	CAISE	1		Software Engineering	P30
International Conference on P2P, Parallel, Grid, Cloud and Internet Computing	3PGCIC	1		Service/Cloud	P31
International Conference on Parallel, Distributed and Network-based Processing	PDP	1		Software Architecture	P33
International Conference on Intelligent Computing, Automation and Systems	ICICAS	1		Intelligent Computing	P44
International Conference on Information, Communication and Computing Technology	ICOICT	1		Information System	P47
International Systems and Software Product Line Conference	SPLC	1		Software Engineering	P7
International Symposium on Distributed Computing and Applications to Business, Engineering and Sciences	DCABES	1	Symposiums	Software Engineering	P3
International Symposium on Software Reliability Engineering Workshops	ISSRE	1		Software Engineering	P25
Mediterranean Symposium on Smart City Applications	SCAMS	1		Service/Cloud	P35
Journal of Systems and Software	JSSOFTWARE	3	Journals	Software Engineering	P14, P23, P26
Information and Software Technology	INFSOF	2		Software Engineering	P5, P45
Software Intensive Cyber Physical Systems	SICS	1		Software Engineering	P28
Information Systems	INFOSYS	1		Information System	P15
International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment	DEVOPS	2	Workshops	Software Engineering	P6, P11

(continued on next page)

Table D.1 (*continued*).

Publication channel	Abbreviation	Count	Category	Community	Paper ID
Software Engineering in the Era of Cloud Computing	–	1	Article	Software Engineering	P2
Enterprise Information Systems	–	1	Article	Service/Cloud	P46
Software Architecture for Big Data and the Cloud	–	1	Article	Software Architecture	P4
Present and Ulterior Software Engineering	–	1	Article	Software Engineering	P9
Microservices	–	2	Article	Service/Cloud	P18, P37
IEEE Software	–	1	Article	Software Engineering	P22
IEEE Cloud Computing	–	1	Article	Service/Cloud	P38
Programming and Computer Software	–	1	Article	Software Engineering	P40
Data Management, Analytics and Innovation, Advances in Intelligent Systems and Computing	–	1	Article	Data Analytics	P41
Computing	–	1	Article	Software Architecture	P42

References

Alkhalil, A., Sahandi, R., John, D., 2016. A review of the current level of support to aid decisions for migrating to cloud computing. In: ACM Int. Conf. Proceeding Ser. 22-23-Marc. http://dx.doi.org/10.1145/2896387.2896443.

Althani, B., Khaddaj, S., 2017. The applicability of system migration life cycle (SMLC) framework. In: Proc. - 2017 16th Int. Symp. Distrib. Comput. Appl. to Business, Eng. Sci. DCABES 2017 2018-Septe. pp. 141–144. http://dx.doi.org/10.1109/DCABES.2017.38.

Althani, B., Khaddaj, S., 2018. Systematic review of legacy system migration. In: Proc. - 2017 16th Int. Symp. Distrib. Comput. Appl. to Business, Eng. Sci. DCABES 2017 2018-Septe. pp. 154–157. http://dx.doi.org/10.1109/DCABES.2017.41.

Althani, B., Khaddaj, S., Makoond, B., 2017. A quality assured framework for cloud adaptation and modernization of enterprise applications. In: Proc. - 19th IEEE Int. Conf. Comput. Sci. Eng. 14th IEEE Int. Conf. Embed. Ubiquitous Comput. 15th Int. Symp. Distrib. Comput. Appl. to Business, Engi. pp. 634–637. http://dx.doi.org/10.1109/CSE-EUC-DCABES.2016.251.

Amy Ann, F., Rob van der, M., 2016. Gartner Says by 2020? A Corporate No-Cloud Policy Will Be as Rare as a No- Internet Policy is Today [WWW Document]. Gartner, Inc, URL https://www.gartner.com/en/newsroom/press-releases/2016-06-22-gartner-says-by-2020-a-corporate-no-cloud-policy-will-be-as-rare-as-a-no-internet-policy-is-today (accessed 5.26.21).

Cavacini, A., 2015. What is the best database for computer science journal articles? Scientometrics 102, 2059–2071. http://dx.doi.org/10.1007/s11192-014-1506-1.

Clayton, T., 2018. Decision Point for Choosing a Cloud Migration Strategy for Applications [WWW Document]. Gartner, Inc, URL https://www.gartner.com/en/documents/3893681/decision-point-for-choosing-a-cloud-migration-strategy-f.

Efremovska, A., Lago, P., 2017. From Legacy to Cloud: Risks and Benefits in Software Cloud Migration, first ed. In: Software Architecture for Big Data and the Cloud, Elsevier Inc, http://dx.doi.org/10.1016/b978-0-12-805467-3.00009-0.

Estdale, J., Georgiadou, E., 2018. Applying the ISO/IEC 25010 quality models to software product. Commun. Comput. Inf. Sci. 896, 492–503. http://dx.doi.org/10.1007/978-3-319-97925-0_42.

Fehling, C., Leymann, F., Retter, R., Schupeck, W., Arbitter, P., 2014. Cloud Computing Patterns. Springer, Springer Vienna, Vienna, http://dx.doi.org/10.1007/978-3-7091-1568-8.

Fritzsch, J., Bogner, J., Zimmermann, A., Wagner, S., 2018. From Monolith to Microservices: A Classification of Refactoring Approaches. Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-06019-0, arXiv.

Ganesan, A.S., Chithralekha, T., 2016. A survey on survey of migration of legacy systems. In: ACM Int. Conf. Proceeding Ser. 25-26-Augu. http://dx.doi.org/10.1145/2980258.2980409.

Ghofrani, J., Bozorgmehr, A., 2019. Migration to microservices: Barriers and solutions. In: Commun. Comput. Inf. Sci. 1051 CCIS. pp. 269–281. http://dx.doi.org/10.1007/978-3-030-32475-9_20.

Gholami, M.F., Daneshgar, F., Beydoun, G., Rabhi, F., 2017. Key challenges in migrating legacy software systems to the cloud — an empirical study. Inf. Syst. 67, 100–113. http://dx.doi.org/10.1016/j.is.2017.03.008.

Grieger, M., Fazal-Baqaie, M., Engels, G., Klenke, M., 2016. Concept-based engineering of situation-specific migration methods. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 199–214. http://dx.doi.org/10.1007/978-3-319-35122-3_14.

Jamshidi, P., Ahmad, A., Pahl, C., 2013. Cloud migration research: A systematic review. IEEE Trans. Cloud Comput. 1, 142–157. http://dx.doi.org/10.1109/TCC.2013.10.

Jamshidi, P., Pahl, C., Chinenyeze, S., Liu, X., 2015. Cloud migration patterns: A multi-cloud service architecture perspective. In: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Vol. 8954. pp. 6–19. http://dx.doi.org/10.1007/978-3-319-22885-3_2.

Kazanavicius, J., Mazeika, D., 2019. Migrating legacy software to microservices architecture. In: 2019 Open Conf. Electr. Electron. Inf. Sci. EStream 2019 - Proc. http://dx.doi.org/10.1109/eStream.2019.8732170.

Kazman, R., Woods, S.G., Carriere, S.J., 1998. Requirements for integrating software architecture and reengineering models: CORUM II. In: Reverse Eng. - Work. Conf. Proc. pp. 154–163. http://dx.doi.org/10.1109/wcre.1998.723185.

Khadka, R., Saeidi, A., Idu, A., Hage, J., Jansen, S., 2012. Legacy to SOA evolution: A systematic literature review. In: Migrating Leg. Appl. Challenges Serv. Oriented Archit. Cloud Comput. Environ.. pp. 40–70. http://dx.doi.org/10.4018/978-1-4666-2488-7.ch003.

Khadka, R., Shrestha, P., Klein, B., Saeidi, A., Hage, J., Jansen, S., Van Dis, E., Bruntink, M., 2015. Does software modernization deliver what it aimed for? A post modernization analysis of five software modernization case studies. In: 2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015 - Proceedings. IEEE, pp. 477–486. http://dx.doi.org/10.1109/ICSM.2015.7332499.

Kitchenham, B.A., Charters, S., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Software Engineering Group School of Computer Science and Mathematics Keele University Keele, Staffs ST5 5BG, UK.

Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., Linkman, S., 2009. Systematic literature reviews in software engineering - A systematic literature review. Inf. Softw. Technol. 51, 7–15. http://dx.doi.org/10.1016/j.infsof.2008.09.009.

Knoche, H., Hasselbring, W., 2018. Using microservices for legacy software modernization. IEEE Softw. 35, 44–49.

Kratzke, N., 2018. A brief history of cloud application architectures. Appl. Sci. 8, 1–26. http://dx.doi.org/10.3390/app8081368.

Lenarduzzi, V., Lomio, F., Saarimäki, N., Taibi, D., 2020. Does migrating a monolithic system to microservices decrease the technical debt? J. Syst. Softw. 169, 110710. http://dx.doi.org/10.1016/j.jss.2020.110710.

Leymann, F., Breitenbücher, U., Wagner, S., Wettinger, J., 2017. Native cloud applications: Why monolithic virtualization is not their foundation. Commun. Comput. Inf. Sci. 740, 16–40. http://dx.doi.org/10.1007/978-3-319-62594-2_2.

Marquez, L., Rosado, D.G., Mouratidis, H., Mellado, D., Fernandez-Medina, E., 2015. A framework for secure migration processes of legacy systems to the cloud. Lect. Notes Bus. Inf. Process. 215, 507–517. http://dx.doi.org/10.1007/978-3-319-19243-7.

Megargel, A., Shankararaman, V., Walker, D.K., 2020. Migrating from monoliths to cloud-based microservices: A banking industry example. pp. 85–108. http://dx.doi.org/10.1007/978-3-030-33624-0_4.

Moore, A.G., 2014. Crossing the Chasm, third ed. In: Marketing and Selling Disruptive Products to Mainstream Customers, Harper Business. HarperCollins.

Munisso, R., Chis, A.E., 2017. CloudMapper: A model-based framework for portability of cloud applications consuming PaaS services. In: Proc. - 2017 25th Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. PDP. pp. 132–139. http://dx.doi.org/10.1109/PDP.2017.94.

Patel, A., Shah, N., Ramoliya, D., Nayak, A., 2020. A detailed review of cloud security: Issues, threats attacks. In: Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020. IEEE, pp. 758–764. http://dx.doi.org/10.1109/ICECA49313.2020.9297572.

Pei Breivold, H., 2020. Towards factories of the future: migration of industrial legacy automation systems in the cloud computing and Internet-of-things context. Enterp. Inf. Syst. 14, 542–562. http://dx.doi.org/10.1080/17517575.2018.1556814.

Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18. http://dx.doi.org/10.1016/j.infsof.2015.03.007.

Petticrew, M., Roberts, H., 2008. Systematic reviews in the social sciences: A practical guide. Syst. Rev. Soc. Sci.: Pract. Guide http://dx.doi.org/10.1002/9780470754887.

Pianini, D., Neri, A., 2021. Breaking down monoliths with microservices and DevOps: An industrial experience report. In: Proc. - 2021 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2021. pp. 505–514. http://dx.doi.org/10.1109/ICSME52107.2021.00051.

Pigazzini, I., Fontana, F.Arcelli., Maggioni, A., 2019. Tool support for the migration to microservice architecture: An industrial case study. In: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). In: LNCS, vol. 11681, pp. 247–263. http://dx.doi.org/10.1007/978-3-030-29983-5_17.

Ponce, F., Marquez, G., Astudillo, H., 2019. Migrating from monolithic architecture to microservices: A rapid review. In: Proc. - Int. Conf. Chil. Comput. Sci. Soc. SCCC 2019-Novem. pp. 7–13. http://dx.doi.org/10.1109/SCCC49216.2019.8966423.

Rai, R., Sahoo, G., Mehfuz, S., 2015. Exploring the factors influencing the cloud computing adoption: a systematic study on cloud migration. Springerplus 4, 1–12. http://dx.doi.org/10.1186/s40064-015-0962-2.

Rana, M.E., Rahman, W.N.W.A., 2018. A review of cloud migration techniques and models for legacy applications: Key considerations and potential concerns. Adv. Sci. Lett. 24, 1708–1711. http://dx.doi.org/10.1166/asl.2018.11142.

Sabiri, K., Benabbou, F., Khammal, A., 2018. Model driven modernization and cloud migration framework with smart use case. Lect. Notes Netw. Syst. 37, 17–27. http://dx.doi.org/10.1007/978-3-319-74500-8_2.

Selmadji, A., Seriai, A.D., Bouziane, H.L., Mahamane, R.Oumarou., Zaragoza, P., Dony, C., 2020. From monolithic architecture style to microservice one based on a semi-automatic approach. In: Proc. - IEEE 17th Int. Conf. Softw. Archit. ICSA. pp. 157–168. http://dx.doi.org/10.1109/ICSA47634.2020.00023.

Shuaib, M., Samad, A., Alam, S., Siddiqui, S.T., 2019. Why adopting cloud is still a challenge?—A review on issues and challenges for cloud migration in organizations. Adv. Intell. Syst. Comput. 904, 387–399. http://dx.doi.org/10.1007/978-981-13-5934-7_35.

Vera-Rivera, F.H., Gaona, C., Astudillo, H., 2021. Defining and measuring microservice granularity—a literature overview. PeerJ Comput. Sci. 7, e695. http://dx.doi.org/10.7717/peerj-cs.695.

Vera-Rivera, F.H., Puerto-Cuadros, E.G., Astudillo, H., Gaona-Cuevas, C.M., 2020. Microservices backlog - A model of granularity specification and microservice identification. In: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). In: LNCS, vol. 12409, pp. 85–102. http://dx.doi.org/10.1007/978-3-030-59592-0_6.

Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requir. Eng. 11, 102–107. http://dx.doi.org/10.1007/s00766-005-0021-6.

Wohlin, C., Runeson, P., Da Mota Silveira Neto, P.A., Engström, E., Do Carmo Machado, I., De Almeida, E.S., 2013. On the reliability of mapping studies in software engineering. J. Syst. Softw. 86, 2594–2610. http://dx.doi.org/10.1016/j.jss.2013.04.076.

Yadav, S.K., Khare, A., Kavita, C., 2020. ASK approach: A pre-migration approach for legacy application migration to cloud. Adv. Intell. Syst. Comput. 1042, 15–27. http://dx.doi.org/10.1007/978-981-32-9949-8_2.

Zhou, X., Jin, Y., Zhang, H., Li, S., Huang, X., 2016. A map of threats to validity of systematic literature reviews in software engineering. In: Proc. - Asia-Pacific Softw. Eng. Conf. APSEC 0. pp. 153–160. http://dx.doi.org/10.1109/APSEC.2016.031.