© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Microservice Testing Approaches: A Systematic Literature Review

Israr Ghani[1], Wan M.N. Wan-Kadir[2*], Ahmad Mustafa[3], Muhammad Imran Babir[4]

[1,2,3]Software Engineering Department School of Computing, Universiti Teknologi Malaysia, Skudai, Johor Baharu, 81310, Malaysia,

[4]Army Public College of Management and Science, Department of Computer Science, Rawalpindi, Pakistan

*Corresponding author

DOI: https://doi.org/10.30880/ijie.2019.11.08.008
Received 22 February 2019; Accepted 27 September 2019; Available online 15 September 2019

Abstract: Testing is a crucial part in software development process to which organizations devote extensive time and effort. The ever-changing industry needs of business necessitate that experts adopt and support themselves to meet these requirements especially in service-oriented software that uses microservice components. In order to introduce newer approaches and essential techniques of architecture in microservice testing, the advancement of "microservice" is the result of such an activity to make the testing quality better. Working on testing microservice has become a newer structure of this testing architecture. This study explains the challenges that the testing world has to deal with and the effective strategies that can be described to overcome them while testing for applications and its design with a microservice testing approaches. Testing approaches that can be developed and successfully applied while working within such a landscape. However, one of the major advantages of using microservice that it offers, efficient, flexible, effective, reusability mechanism. Furthermore, it is a secure way to reduce the development and testing time and cost. The security, performance, traceability, compatibility, complexity, effectiveness and scalability become some of the major concerns when testing approaches, frameworks, tools and models are applied for each microservice repository and no previous research addresses these concerns. In this review, we present some testing approaches, frameworks, tools and models to address all these concerns. As for the consequence, it can be said that microservice service testing technique are still broadly open for further enhancement.

Keywords: Microservice testing, Microservices-based application testing, empirical study, Quality assurance, Testing Approaches

1. Introduction

Microservices are growing in popularity in the software industry [1], [2]. Microservices are loosely coupled architectural elements that can be independently deployed by fully automated machinery. In order to make microservice more efficient and achieve higher success rates, several testing models, frameworks, tools and approaches have been proposed and widely used in selected 15 primary studies. While testing plays a vital role in these realizations, it is still a challenging task to test microservices. Therefore, a comprehensive systemic literature review is presented where we could see what the critical issues are, with a focus on quality attributes and existing solutions to resolve the testing issues. A sample illustration of all levels of testing (unit, end to end (E2E), contract testing, integration testing and component testing) of current micro service testing is used as a test strategy.

*Corresponding author: israrghani@outlook.com
2000 UTHM Publisher. All right reserved.

Microservices are software system utilized for coordination of various web applications and specific measures [3]. Microservices have become the most common sense approach utilizing minimal effort for conveying information that is disseminated between applications running on different operating structures, languages, and platforms. Microservices can be classified into two types: microservice utilized on Intranet and Internet. The service given by the intranet is utilized exclusively inside the concerned organization and are not accessible for general use while micro service via the Internet are available for all to utilize.

Internet and microservice security[5], performance [6][7], traceability [8], compatibility [9], complexity [10], effectiveness [11] and scalability [12] have become a leading issue and there are noticeable privacy concerns. As a result, analysts require programming skills and explicit tools [13], framework [14] , model [12] to test them. Many tools, frameworks, approaches [15] and models have been introduced to ensure microservice testing and a few methodologies have been suggested to examine the nature of these techniques. This study is focused on five essential testing approaches described in Fig 4.

The SLR reported by this paper aims at evaluating the state-of-the-art microservice testing approaches in terms of their support in addressing the related quality attributes. There are 465 papers identified from our initial search. After several stages of filtering, the number of papers is finally reduced to 15 papers that are selected as the primary studies for this SLR. The primary studies were systematically analyzed to answer the defined research questions. The contribution of this SLR is mainly on the identifying of the main issues related to microservice testing, the quality attributes that should be addressed by the microservice testing approaches, and the existing solutions and gaps that address the identified issues. This information is useful in setting the future directions in microservice testing.

2. Research method and motivation

Most of the studies use different models, frameworks and tools to recommend as a solution for microservice testing strategy. However, these tools, models and frameworks still have the extra overhead and are not appropriate for use on different platforms while testing microservice. Additionally, at business level microservice testing, there should be a solution that consolidates.

The bottleneck in microservice testing is the maturity of exclusive parameters like re-usability, performance, security, interoperability, maintainability, reliability and auditability. There is a need to conduct more empirical studies to test the most important mentioned parameters while microservice testing. Non functional requirement (NFR) in [16] microservice testing is neglected in previous literature reviews as a result of the environment of microservice testing process has been abandoned by the industry testers. There is a need for increasing experiential research in this area. Few studies have discussed automated microservice testing as an benefit of microservice testing; some studies [17] propose different automation testing approaches for microservice testing. However, there is insufficiency of comprehensive studies for microservice testing and more studies are required to accomplish more research with an alternate kind of testing in microservice. Research should be done on reusability, auditability, security, reliability, performance, interoperability and maintainability in microservice and about the mechanism of different testing proceedings.

Hence, existing tools and approaches [18] are not sufficient to reveal structural errors to test microservices [19]. This has motivated us to go through the literature and analyze the microservice testing with respect to security, performance, traceability, compatibility, complexity, effectiveness and scalability. In order to conduct a comprehensive analysis on testing of microservice, we have defined research questions and planning from Table 1 to Table 4 stepwise using research strategy consistent with the guidelines provided by Petersen et al. [20].

Table 1 - The Research Question and Motivation

No.	Question Statement	Motivation
RQ1	What are the issues in testing microservices?	The objective is to distinguish and expose the gaps in recent research and hence set the trend for future research.
RQ2	What are the quality related aspects and concerns with respect to testing of microservices?	The aim is to perceive and investigate what are the possible methodologies, strategies and models to best describe distinctive perspectives and level of microservices testing.
RQ3	What are the existing solutions to solve the microservice testing issues?	The purpose is to explore the available studies that were appropriate to microservices testing to highlight the gaps in them and look for future solution fundamentals.

2.1 Research Questions

The primary interest of researchers is to identify the scope of future research activities for testing of microservice. This systematic review identified the existing basis for the research work and make it clear where the proposed research conclusions fit into the current body of knowledge. Based on selected primary studies, the following research questions have been raised to make the microservice testing techniques and approaches more effective & useful for the industry

as well as for the future researcher. The following steps are outlined to specify the research questions. First, we analyze the data on the specified questions, which are most common testing approaches to address microservice testing issues in searched studies [3]. Second, we reviewed the previous SLRs use as a guideline. Following are the inclusion and exclusion criteria as determined by the research questions generated after selection of primary study.

Table 2 - The Selection Criteria

No.	Question Statement
Inclusion Study	• Papers which are published from 2010 to 2018. • A search based on testing of microservice. • A search which addresses specific testing problems with microservice. • A search which has the practical support for microservice SOAP and RESTful testing technique. • Papers which have quality related aspects and concerns with respect to testing of microservices. • Papers which have existing solutions to solve the microservice testing issues.
Exclusion Study	• Papers which are published before 2013. • Papers which address issues identified with microservice, but not accommodating quality assurance. • The articles are excluded based on title. Articles that are hard to distinguish dependent on the title are sent to the next stage. • After reading the abstract, the articles are identified. • Duplicate papers are also disregarded in this phase.

2.2 Study Selection

In the search procedure, all the research work that has been nearly done on testing of microservice has been collected. First, different digital databases are searched i.e. “"ACM Library", "IEEE Xplore", " ISI Web of Science", " Springer Link", " Science Direct ","Wiley” and "Scopus"”.

Table 3 - Database on-line sources

Library Source	ULR to access
Library source	http://www.acm.org
ACM Library	http://www.ieeexplore.ieee.org
IEEE Xplore	http://www.webofknowledge.com
ISI Web of Science	http://www.springerlink.com
Springer Link	http://www.sciencedirect.com
Science Direct	http://www.onlinelibrary.wiley.com
Wiley	http://www.scopus.com
Scopus	http://www.acm.org

Manual search was performed on various sources to enhance the outcomes, results of the coherent databases. Manual search was executed using Google Scholar as a meta-engine to search for papers that did not appear in the central databases. Also, the search was limited to papers published between 2013 and 2018, as it was during these years that terms such as Microservice Testing Techniques began to appear. As a result, we have obtained a total of 465 papers. The used sources contain almost all-important workshops, symposiums, journals, and conference proceedings within the software engineering domain. In Table 3 the commonly used selected data sources and search expressions are mentioned within the domain of software engineering and information system.

It is important to make sure that papers referring to microservice testing and conventions are also incorporated. As the search objective included the option (OR) Microservice AND Testing, the search is being extended rather than restricted. Search string in five systematic databases was submitted accordingly as shown in Table 4. The search was limited to the title, abstract and keywords which deliver sufficient material to indicate whether a paper is in the context of the work or not.

IEEE 2013 to 2018. Studies are searched by executing the search string in segment 1, 465. The studies were reduced to 173 after title-based selection. In phase 3, it is shortlisted for 49 after reading the abstract and removal of duplicate studies. The total studies are finalized to 15 after a full text reviewing. The selection criteria for all phases are explained in Fig 1, and Table 5 presents the list of identifying primary studies itemized. This section explains the review protocol and selection process of primary studies.

Fig 1 presents the results of steps which provide details of the selection of primary study. After examining the studies, 15 primary studies were identified, selected from conference proceedings, journals, and book chapters spanning 2013 to 2018. All four phases of the selection process are described in Fig 1. This segment clarifies the review protocol and determination procedure of primary studies.

Table 4 - Search sequences, data sources and designated studies

Name	Results	Query String
ACM Library	49	Issues OR Challenges in microservice testing
IEEE Xplore	112	Issues in microservice testing OR Quality assurance OR attributes OR issues in microservice AND challenges in microservice testing
Web of Science	83	Issues in microservice testing OR Quality assurance OR attributes OR issues in microservice AND challenges in microservice testing
Springer Link	36	Testing issues in microservice OR quality assurance OR attributes OR issues AND testing of microservice
Science Direct	56	Testing Issues in Microservice AND microservice OR Challenges in Microservice Testing Issues OR API Testing Issues
Wiley	73	Microservice Testing Issues OR Challenges in microservice Testing
Scopus	56	Issues in Microservice Testing OR API Service Testing Issues
Total	465	

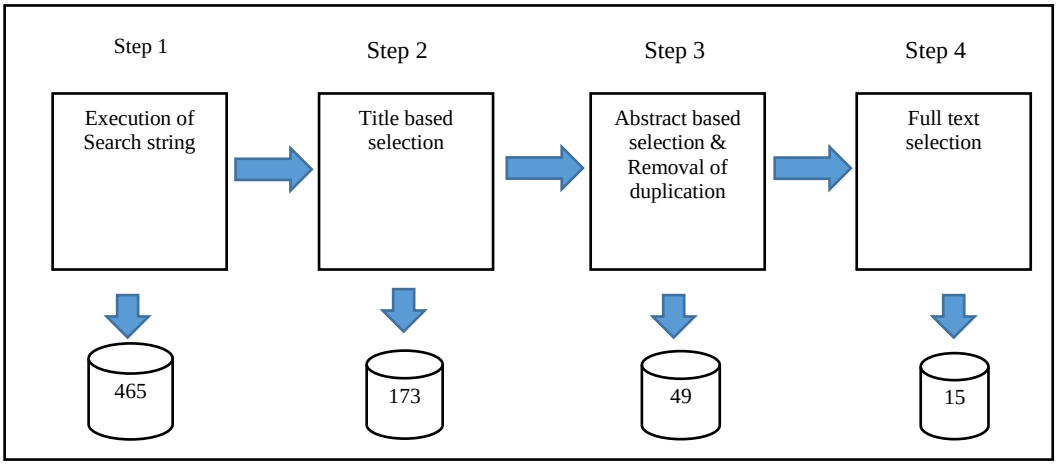

Figure 1 - Phase of Collection Method

By combining the manual and automatic exploration, we have identified 15 potential primary studies that are directly associated with the questions we are trying to answer. Including literature reviews [1], which have considered the role and approaches of microservice testing in the past few years, we have identified 15 technical papers. Focused on microservice testing, the study presents and offers an overview with research questions and classification of some approaches mentioned in Table 5 as a state of the art.

The subsequent phase is introduced to deal with recovered strategies relying upon their related work and importance of planned research conditions. By following these four steps, 15 recent and different studies are selected which were additionally contemplated one by one to perform the examination depending on our quality parameters, questions and research assessment criteria.

2.3 Data Extraction and synthesis

To extract appropriate material from selected studies, we have executed data extraction process which shows the synthesis of existing solutions. The result shows that most solutions are based on using existing tools, frameworks and approaches explain research questions on which research study is based. The result and discussion section in Fig. 4 shows the contribution of all solution approaches. It was preferable to use unit testing in the industry and due to the difficulty in conducting an industry survey instead of practical experimentation, some studies prefer conceptual solutions for automation in microservice testing. Following is the essential extracted material from all studies for that purpose, and guidelines for such studies in order to explore the introduction and methodology:

1. The source of studies, i.e. from journal or conference, the title, and the names.

2. The structure of each study conducted.
3. The method, pros and cons of microservice testing procedures and approaches.
4. Supportive facts of microservice testing technique.
5. Adaptive facts necessary for microservice Testing techniques.
6. The facts required to answer all research questions.

Table 5-Selected Primary Studies

Study ID	References	Description
S1	[2]	A Systematic Mapping Study in Microservice Architecture
S2	[3]	Computational Science and Its Applications
S3	[5]	A survey on security issues in services communication of Microservices-enabled fog applications
S4	[6]	Systematic Resilience Testing of Microservices Victor
S5	[7]	An Architecture to Automate Performance Tests on Microservices
S6	[8]	Migrating towards Microservices: Architecture Smells
S7	[10]	Microservices in Agile Software Development: a Workshop- Based Study into Issues, Advantages, and Disadvantages
S8	[11]	Contextual Understanding of Microservice Architecture: Current and Future Directions
S9	[12]	The pains and gains of microservices: A Systematic grey literature review
S10	[14]	Using Service Dependency Graph to Analyze and Test Microservices
S11	[15]	Microservices-Based Software Architecture and Approaches
S12	[16]	Microservice Architectures for Scalability, Agility and Reliability in E-Commerce
S13	[17]	Learning-Based Testing of Microservices
S14	[19]	Experience on a Microservice-based Reference Architecture for Measurement Systems
S15	[21]	A Reusable Automated Acceptance Testing Architecture for Microservices in Behavior-Driven Development

The outcomes of data extractions are treated in Result and Discussion section to elaborate their importance regarding research. Furthermore, other additional material to categorize the different studies was included and the categorization and execution process and its flow was described in Fig 2. To synthesize the selected studies, narrative synthesis was used which elaborated the research methodology performed to get selected primary studies for analysis in Table 5. To find the research questions (RQs), and based on the capacity or study of interest [23] to evaluate and interpret it, research methodology and strategy SLR is a systematic way. The preferences in a review process, authenticate that the existing work is practical, and seems reasonable. SLR has three (3) steps: planning, conducting and documenting according to guideline discussed by Kitchenham as presented in Fig 2. Requirement of research is previously described in Section 1, and Motivation where the specification of research questions is given in Table 1. Data extraction method is also discussed in Fig 1. Finally, the result of Primary study is given in Table 5 using research methodology in Fig 1 and 2.

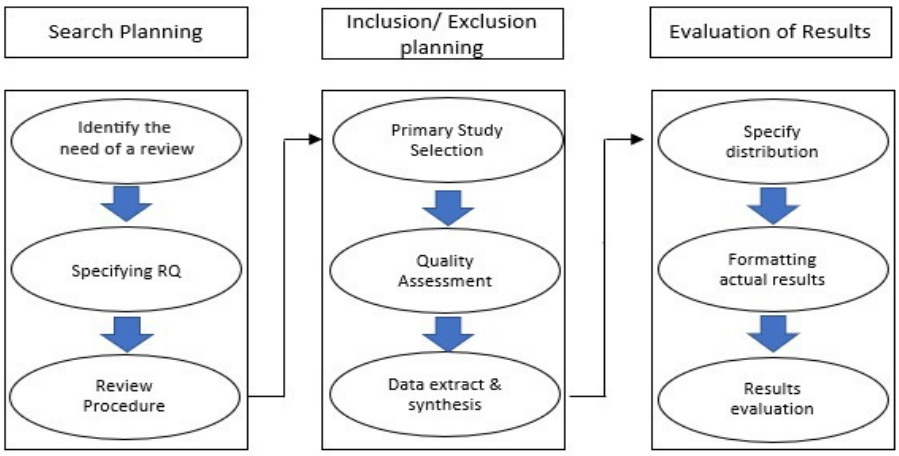

Figure 2. Research Methodology from [24]

Research methodology and all research questions are appropriate to quality assurance of microservice, which is being done through current techniques and approaches. Additionally, this study contains some parameters which are most used, common and significant from selected primary studies for quality assurance of microservice testing. In Fig 3, the classification of the concretization category over the years is shown. Analysis of microservice testing concentrates the greater number of papers in the years 2017 and 2018 that produce actual primary studies.

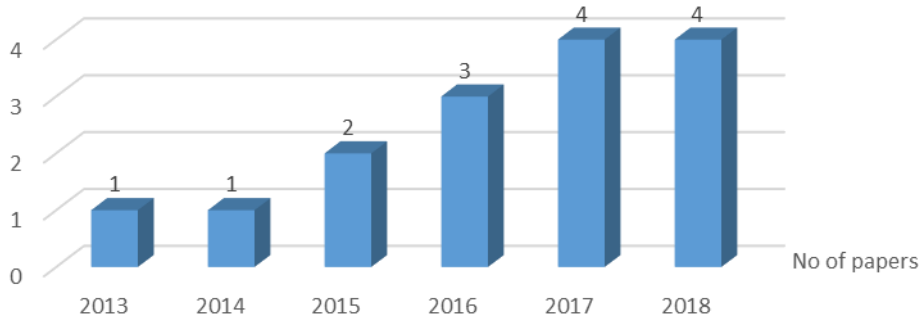

Figure 3 - Primary Study Year Wise Trend

The classification of the major contribution over the years is shown in Fig. 3. From the results of the selected primary study, it is noticed that most of the papers in selected primary study were published in 2017 and 2018.

The classification of primary studies that have tools, models and frameworks supported is available in Fig 3 studies. It is possible to see that most of the primary studies of the works are supported by tools, models and frameworks by the authors.

Also, we can see that there is not a great difference between the numbers of studies introduced between 2013 and 2015, were only 4 primary studies were introduced during these years. Furthermore, the majority of the studies were published between 2016 and 2018, when total 11 effective studies on microservice testing were introduced.

The outcomes of this classification are introduced in Fig 3. It seems that the majority of the literature focused around explanation and clarification research in the most recent six years. This classification isolates a lack of ethical and experience papers, which is a clear prospect for future study. Experience papers can only be written when the approaches are essentially realized in the industry and are used. We believe that the absence of this kind of papers shows that this field is still in an early stage and requires a substantial measure of attention.

This study is the analysis of following parameters to answer all three research questions. All three questions have one goal, which is to determine how quality of microservice testing can be progressed further according to RQ1. Selected parameters which come through using search strings from Table 4 and data sources as defined in phase of collection method are shown in Fig 1. Research methodology from a selected primary study is shown in Table 5, which are security, performance, traceability, compatibility, complexity, effectiveness and scalability. The actual aim of this study is to ensure a comprehensive evaluation of selected studies for the most common issues while analyzing these parameters found in a selected primary study in microservices and the following is the details for said parameters

Security: Is a basic term in software and microservice testing in context or microservice structure to play out its delivery by using a base measure of resources, either successfully linked within the system or other systems. Several times, a system needs to execute information interchange through various mechanisms to interact with other systems. For security testing microservices, this is a critical aspect, as the tool, technique, approach or model must be able to securely interact and then coordinate with components [1], [25].

Performance: Specifies the ability of the existing tools, approach and model to perform testing in microservice. It also represents how fast a service request can be completed. This parameter manages testing the affectability and execution of the micro service under the testing of explicit remaining workload and conditions. Stress and load testing are a vital part of performance testing in software and for microservice testing as well. Stress testing is focused on testing the execution of a framework under extraordinary load, expanding its most extreme limit. Load testing is done by executing the framework under specific predefined load to comprehend its conduct cooperation with the diverse segments of the different components of the system [7].

Traceability: Refers to the capacity of the testing structure to effectively communicate with effective tracking and its activity with different frameworks whenever required. Testing microservice should be traceable in context between the different development and deployment environments used to implement services. Usually a framework needs to perform data trade through different components to associate with other frameworks. For testing microservice benefits, this is a vital factor, as the instrument must have the capacity to collaborate and arrange with outer parts [8], [26].

Compatibility: Evaluates if the microservice can be compatible to address weaknesses during the testing, and can be altered or developed in the future. Code and test practicality can be accomplished by following great coding and testing standards, doing appropriate documentation of coding and testing to maintain compatibility in terms of

programming dialect and testing benchmarks. Code and test modularization and Object-Oriented Programming can likewise help in compatibility for current and future system [9].

Complexity: Specifies the ability of the techniques, approaches or models to accomplish complex testing. This deals with testing the complexity and sensitivity of the microservice under the same explicit embedded functionality and conditions. As large number of variable constraints can be complex from the earlier version's test cases. Therefore, testing of complexity can reduce a great deal of effort for resolving constraint standards for those variables while testing a new version of the application before deployment into production [10].

Effectiveness: Is the ability of the application to execute its task by consuming a lower capacity of resources and represents the ability of a microservice to perform its required functions after testing. It also includes earlier conditions of a service for a specified time interval and also represents how fast a service request can be completed. The effectiveness is the overall measure of a microservice to maintain its service quality while testing. This parameter estimates if the testing application makes the most enhanced use of microservice performance and effect. A tool, approach and framework consuming too much disk space or even system memory indicate a low effectiveness. Similarly, if a testing tool, approach, model, framework or technique takes too much time to produce the results, it can be observed as ineffective [28].

Scalability: Must handle a large number of devices that compose the service framework, support millions of users that use the platform services and a very large volume of service transportation related data that must be stored and processed after its quality assurance. The scalability is the ability of a system to increase capacity of workload by applying cost effective approaches for extending the system's capacity [29].Stated quality constraints and terms are the basic quality testing points which should be realized and achieved for assuring quality testing of microservice.

RQ2: What are the quality related aspects and concerns with respect to testing of microservices?

Microservice architecture is the bridge between corporate objectives and the software system. Selecting and planning an architecture that satisfies the functional and nonfunctional as well as the quality attribute requirements (security, performance, traceability, compatibility, complexity, effectiveness and scalability) are energetic to the success of the system. The quality attribute requirements, in particular, drive the software architecture design and testing. Mentioned parameters are concerned with analyzing selected studies to demonstrate the RQ2 quality related aspects and concerns which are security, performance, traceability, compatibility, complexity, effectiveness and scalability with respect to RQ1 presented in Table 1. It can be expected that when it comes to testing of microservice, additional emphasis is given to actual research leading to solutions than is given to hypothetical research similar to literature surveys. Because of the large volume of microservice testing, research studies, which identify with the trials executed in the lab and are not yet utilized in the genuine business environment. We can conclude that this study has a great deal of open research issues that are yet to be investigated.

RQ3: What are the existing solutions to solve the microservice testing issues?

Fig. 5 and Table 7 describe testing procedures that are being researched according to selected parameters security, performance, traceability, compatibility, complexity, effectiveness and scalability from Primary study in Table 5. There seems to be an overflow in microservice testing to address common quality related aspects in microservice which have been explained in the parameter section. As a result, many testing approaches which may be very practically identical at detailed level are reported with different names. This promising area of research and a lack of established terminologies has encouraged us to prolong this research to a systematic literature analysis. Results from this mapping study gave a detailed outline of the current state and flow condition of research in testing of microservice. Therefore, for definite experiences, this study should be extended into a systematic literature review.

Existing testing techniques, approaches, models and framework proposed solution in selected primary study is done through different methodologies, which are frameworks, models and tools available in Table 7. Other testing strategies, including unit testing, component testing, integration testing, contract testing and E2E (End to End testing) have been obtained from these selected primary studies and existing approaches. The RQ3 question is useful for the testers to select testing existing techniques and approaches before providing service. In response to this inquiry, our attention will be on parameters and testing techniques and approaches utilized in each study.

RQ3: "What are the existing solutions to solve the issues?" It is useful for researchers to recognize most recent testing solution areas which are not yet addressed thoroughly. This prerequisite is more concerned with the limitations, drawbacks and gaps of the existing testing techniques and approaches. The answer to RQ3 is explained in-depth in the following Result and Discussion section.

3. Result and discussion

3.1 Microservice Quality with Analysis

In order to search quality service containing microservice, security, performance, traceability, compatibility, complexity, effectiveness and scalability parameters, fifteen researches from primary studies are involved. Data abstraction is done to extract a satisfactory amount of data from each study to formulate the results to address research questions for microservice testing techniques and approaches with solution to evaluate. This is done in order to evaluate how each study addresses the most common issues found in selected primary and quality related aspects according to RQ1 and RQ2. With the intention of analyzing these parameters, we have formulated parameter consequences in tabular form. The symbols "✓" and "x" are used for “Existing” and “Lacking”, respectively, to show that the explicit techniques and approaches address what parameters in the context of microservice testing.

Table 6 provided the details which primary studies discussed and provide solutions according to the selected parameters from primary study. In the following segment, the results of the review are presented. Each subcategory tries to answer the questions we have defined in Table 1 above concerning the conditions used, how microservice has been tested and approaches by using different techniques, approaches, model and frameworks.

Table 6- Selected Studies for Analysis

Parameter	Selected Primary Studies for Analysis														
	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[10]	[11]	[12]	[14]	[15]	[16]	[17]	[21]
Security	✓	✓	✓	✓	x	x	x	x	✓	✓	✓	x	x	x	x
Performance	✓	✓	✓	✓	x	✓	✓	x	✓	✓	✓	x	x	x	x
Traceability	x	x	x	x	x	x	✓	x	x	x	x	x	x	x	x
Compatibility	x	x	✓	x	✓	✓	x	x	x	x	x	✓	x	x	x
Complexity	x	x	✓	✓	x	x	✓	✓	✓	✓	✓	✓	x	✓	✓
Effectiveness	x	x	✓	✓	x	x	x	✓	x	x	✓	x	x	x	x
Scalability	x	✓	✓	x	x	✓	x	✓	x	✓	x	x	✓	x	x

For quality assurance of microservice, the testing mechanism for security, performance, traceability, compatibility, complexity, effectiveness and scalability have been analyzed. Study with symbols "✓" have been discussed and proposed appropriate solutions and study with symbol "x" have not been highlighted or discussed according to RQ3 in Table 6 as per selected primary study in Table 5. Table 6 presents the trends of how many studies discussed this issue and proposed solutions against each parameter.

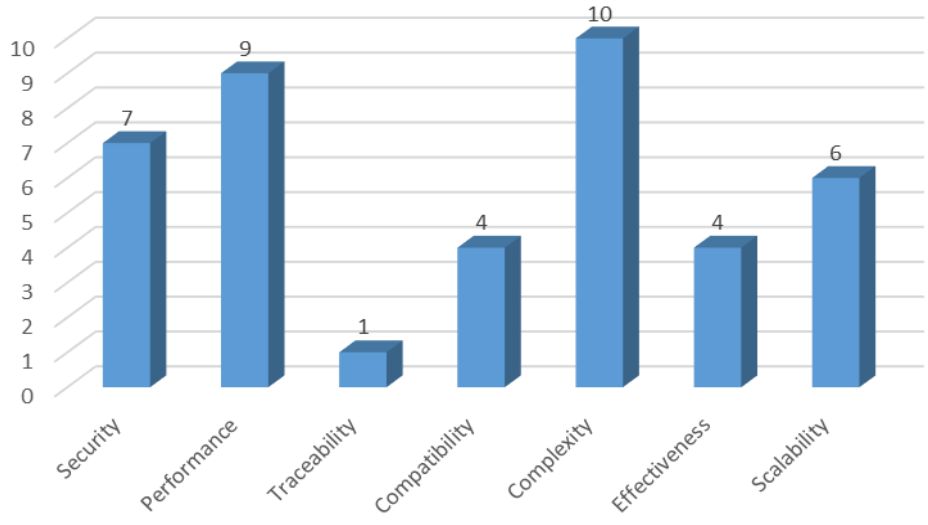

Figure 4 - Trend of selected primary study

Fig 4 shows the trend of the main influence for testing of microservices performed through maintaining the security, performance, traceability, compatibility, complexity, effectiveness and scalability issues. Fig 4 explains with the aim of using different techniques, tools, framework or models to test the mentioned parameters according to the selected primary studies in Table 5. Therefore, only 7 studies are focused on security and 9 studies are concentrated on performance while 10 studies out of 15 primary studies proposed solutions for complexity only. On the other hand, traceability, compatibility, effectiveness and scalability still need more concentration to make the microservice quality better according to the selected study mentioned above earlier. It will be better for upcoming studies to fulfil the requirements against each parameter while suggesting solutions. These testing approaches, tools or frameworks to make the service better as suggested parameter are the key indicators for quality testing associated with microservice and its components. Following are the major contribution of suggested approaches, tools and framework or models to meet the requirements of highlighted parameters mentioned in Fig 4 according to RQ3.

Table 7. Major Contribution in the Literature

Contribution Types	References
Approaches/Methods	[2],[4], [7],[8],[10],[11],[12],[16]
Tools	[3], [6], [15]
Frameworks/Models	[5], [14], [17],[21]

Table 7 presented the major division in this facet. Test generation methods, tools and techniques refers to the procedure used to generate the microservice test proposed to cover the test collection measures. Applied manual test, auto test, and graph search algorithms, model-checking, perceptive further manual or unit test is the process to make microservice testing better. To make it more prominent, it is important to know the major contribution for each selected analysis mentioned in Fig 4 using existing approaches, tools and frameworks suggested by authors in Table 7 which shows the trend against each parameter.

According to RQ3 microservice testing solutions are proposed by means of existing tools, frameworks and approaches. Due to the evolution in the microservice industry and the fluctuating behavior of business, the testing process becomes speedy using mentioned approaches framework and tool in Table 7. The primary studies based on microservice testing features and impact on measurement are listed in Fig 5 for analysis for mentioning techniques, tools and approaches discussed in Table 7.

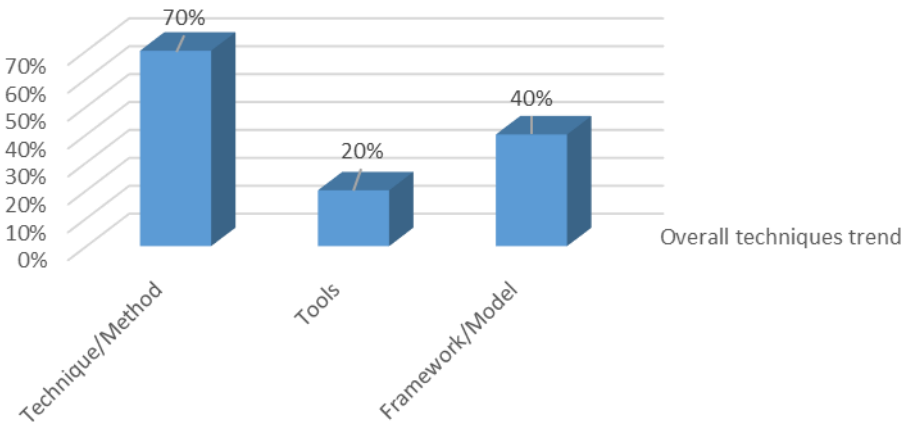

Figure 5. Classification of the Major Contribution

In Fig 5, the classification of primary studies that have tools, frameworks and approaches support have been analyzed to know what kind of technique or approach has been suggested most by the authors to test microservice. Results indicate almost 70% of the studies are supported by different approaches or methods, with 20% are supported by tools that were introduced for that particular resolution of microservice testing. However, 40% are supported by different framework models created by the authors to make the microservice better while testing of microservice security, performance, traceability, compatibility, complexity, effectiveness and scalability.

Selected conditions were distinguished by perusing the abstract of the papers in order to find keywords and point of view that better reflect the impact of the paper. Thus, we have arranged the papers into the classifications called (Major Contribution) presented in our ordering scheme shown in Fig 5. This methodology was completed with the

assistance of spreadsheets showing every feature and its dimensions. The results of this process are concisely shown in Tables 2, 3, 4 and 5 while testing strategy of microservice is shown in Fig 1.

3.2 Test Strategy for microservices

In selected primary study available in section 2 Review Method, microservice testing is on the basis of the latest approaches of microservice testing. The primary study we have selected contains unit testing, component testing, integration testing, contract testing, and end to end testing (E2E) used or discussed to evaluate security, performance, traceability, compatibility, complexity, effectiveness and scalability.

A microservices architecture contains many moving parts with different guarantees and failure modes. Testing and verification of these systems are expressively more nuanced and difficult than testing a traditional monolithic application. An effective and complex test strategy needs to account for both testing individual services in isolation and the verification of overall system behavior [30]. Following testing strategies have been proposed to validate the quality of suggested parameters in Table 6

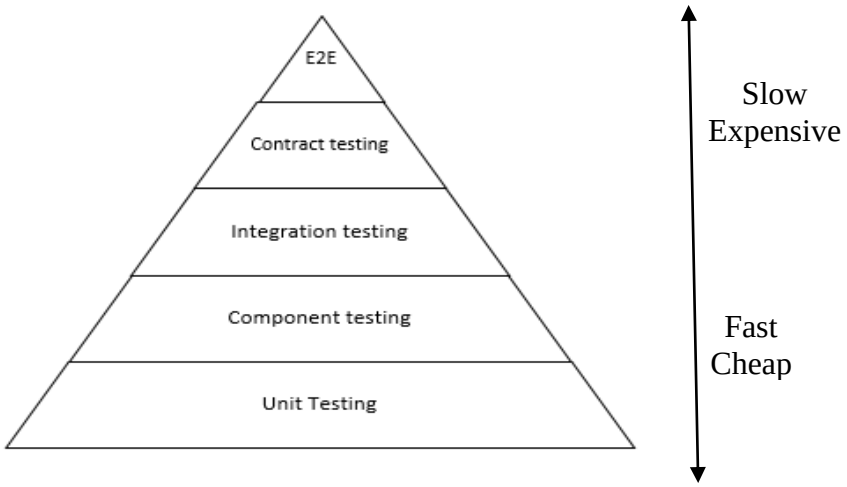

Figure 6. Level of Microservice Testing

With the intention to review to encapsulate the current state of the art of functional and nonfunctional requirements, Fig 6 describes the testing strategy. Unit testing, Component testing, Integration testing, Contract testing, E2E testing of microservice have been reviewed to identify challenges associated with using these existing approaches, tools and frameworks. Following is the purpose of each testing level to test microservice testing by the authors.

1. **Unit tests:** Tests that cover the smallest piece of testable functionality in microservices. It makes sense that unit testing would be a part of any microservices testing strategy. In a monolith architecture [31] there are also benefits to unit testing. Isolating the smallest usable components of an application and testing them one by one is conducted to ensure everything works on its own before integrating with the application as a whole. However, with microservices, it is a bit easier to do as all of those components are already separated out.

2. **Component tests:** When you look at component tests for microservices, a component is a service that exposes certain functionalities. Therefore, component tests for microservice can only be acceptance tests for services and your tests need to validate whether the service provides the functionality that it promises [32].

3. **Integration tests:** Integration tests, from microservice testing perspective, deal with testing integrations and interface defects for components within the service [33]. This involves testing the interfaces between them for defects that might interrupt the interactions between services. Numerous microservices environments rely on RESTful HTTP or other messaging protocols to interface with one another. Running gateway integration tests will locate any errors at the protocol level. Microservice integration testing will ensure that each component is behaving the way it should when contacted. The API interfaces between the components should also be tested to verify they are delivering the right information in the correct format.

4. **Contract tests:** Contracts testing of APIs of microservices are conducted to see if the API is valid or if the microservice reputations is API. A casual variation of this contract test is consumer driven contract test. This test is written by consumer service of an API. The consumers classify this contract in a suite of tests that get run on every change to the API. That way, if a change to the API breaks a contract that one of its consumers expect, this breaking change should be highlighted early in the testing pipeline while performing testing of microservice [34].

5. **End-to-end tests (E2E):** End-to-end tests are more coarse-grained and try to test the functionality of an overall system. Depending on the deployment architecture like microservice, while deploying all services in a pre-production environment after testing in an aggregate manner, end-to-end tests have been considered at this stage since end-to-end tests are usually delicate and take a long time to run. If microservices are completely independent and don not get deployed to a pre-production test environment, then consider approaches that test in production [21].

Fig 7 shows the classification of testing approaches and testing level facets according to RQ2 and RQ3. It is shown that most of the works deal with the Unit testing. However, in terms of integration testing, few studies are about component testing and some authors discussed contract testing to ensure microservice testing in selected studies as shown in Fig

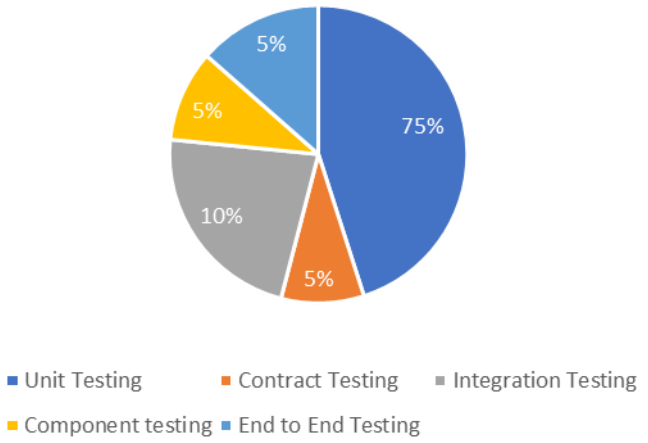

Figure 7 - Contribution of solution approaches with respect to existing studies

In Fig 7, the distribution and classification of testing levels are mentioned by the authors. Results indicate that most of the testing level suggested in primary study is about Unit testing while testing microservices. Unit testing has been preferred by almost 75% as most of the testers prefer to assure the quality testing manually instead of automatically or using tool, framework or model because of sensitivity of data and reliability of the service. Only 10% of selected primary studies suggested integration testing according to the service usability and interoperability while 5% of the studies preferred component, contrast and E2E testing levels because the use of these testing levels are less than the unit testing integration in microservice area.

The reason for using the suggested approaches from existing primary study technique can be summed up as:

1. **Relatively easy:** This order of testing is easy but difficult to use as the microservice analyzers don not need material of the inside structure of the system. Therefore the generation of test cases in microservice is a comparatively stress-free assignment.
2. **Faster testing:** In mentioned testing approaches in Table 7, the microservice testers only must interact with systems. For Graphic User Interface (GUI) with functional and nonfunctional requirements of microservice, no time is exploited in evaluating the structure of the microservice testing. This kind of practice saves the lot of time of testers and makes the testing process enormously easier and faster using said approaches, frameworks, tools and models.
3. **Easiness:** Sometime, testers must deal with bulky and composite systems. Some testing approaches or techniques like unit testing make the testing procedure easier. Mostly, testers are only dealing with the valid and invalid responses and the estimated output into microservice testing environment.

3.3 Challenges and testing levels of Microservice Testing

RQ1 according to the analysis of earlier studies, it has been observed that current and quality measures must be satisfied. Further to mentioned parameters in Table 6, some other system architectural parameters to make the best quality of microservice are also significant, for example interoperability, auditability and reusability measures. Most of the quality parameters are mentioned in Table 6, which can also be helpful in the improved testing of microservices.

The most energetic part which needs more concentration to progress is to close the gap between testers and consumer. This is because eventually only the end user is going to use the service and product by the developer and testers in the organization of the respective product. But this is not a simple activity. The other challenging issue is the expanding number of APIs, as there is a distinctive API for the respective service [35]. Moreover, the matter of

upgrading and maintenance of current service reasons are also quality complications as it is possible that not every user upgrades his service eventually. Thus, there is the need to produce such testing approaches that can work for all kinds of services rather than maintaining quality for the different kinds of services once, if at all. The most important points are to assure security, performance, traceability, compatibility, complexity, effectiveness and scalability of microservice through testing according to the research questions mentioned in Table 1.

RQ3 is geared toward assuring better quality of microservice. As previously mentioned, a broader study is required to:

- Define the quantity of test attempts in terms of frequency that should be required for reassuring quality in microservice testing.
- Define the required levels of testing according to each perception (i.e. Developer, Provider, Integrator and Third Party as well as for End User)
- Potential testing approaches at Integrator and third-party level
- Most used quality features in Table 6 are mentioned in selected studies which are useful in outcome of verifying domain to expand quality performance for each level of microservice testing.

However, the majority of the parameters were not properly validated or assessed while testing of microservice as mentioned in Table 6. The relation between the 15-selected paper's contributions are precisely explained in Figs 4 and 5. The results of this work against selected parameters using existing approaches tools, framework and models are shown in Fig 8.

3.4 Testing approaches Used for Testing microservice

This evaluation initiated a number of known testing approaches or techniques through which testing of the microservice is done such as Unit, Component, Integration, Contract and End to End Testing. Different test case methods, tools and models were introduced to improve the quality of microservice testing according to the selected parameters from a selected primary study in Table 6 and Fig 4 to fulfill the requirements according to RQ3.

Microservices testing involves different interpretations through less or no detail from user crosswise. As the services can be insignificant and bulky, an approach must be used which can be applied for every size service. Thus different testing approaches, techniques, tools and model mentioned in Table 7 are appropriate to use because of easiness and time proficient approaches.

With the intention of analysis between each primary study, Table 8 displays the quality attributes we have defined in our study for good and valuable quality testing techniques. When we discuss quality attributes of microservice testing, we are ultimately analyzing what has been claimed in the primary study, what the study method was and how the research question was answered using defined quality parameters.

To evaluate which primary study is more comprehensive and claimed to improve results that contribute to knowledge, the following comparison is presented to analyze quality study. This analysis shows the main contribution for quality studies of microservice testing approaches according to the selected primary studies using Table 5 as a research methodology in Fig 1 and 2.

The symbols "✓" and "x" are used for present and absent, respectively, to show which study satisfies the requirements according to the conducted analysis from the explicit selected primary study in Table 5. The major contribution of microservice quality testing studies are shown in Table 8 below.

The results of Table 8 using symbol "x" and "✓"show the absence and presence of the primary study according to the appropriate comparison questions. However, in results, we can see there are many limitations and weaknesses in selected primary studies. Some studies did not even discuss the solution inclusively in its related work and also no comparison was made with previous studies. Also, no comparison was presented during experiments, Presented Results and Claimed to expand the results sections

Most of the selected primary studies neither presented the outcomes of testing experiments using suggested tools, frameworks nor approaches available in the analysis. It shows that more concentration is needed in upcoming studies with comparative analysis between the previous and current studies to make the testing quality better in microservice.

Analysis of comparison of selected quality studies uncovered that only 6 primary studies are inclusive in the literature (related to the work), and 10 studies are proposing approaches or techniques. Therefore, 10 studies out of 15 primary studies conducted experiments while testing microservice and 7 presented outcome as a result. These results were not compared with previous studies which shows that most of the studies contain experiments without contributing knowledge and only 1 study compared the results with previous studies. This analysis shows that studies that compare the results with previous studies require more attention with the purpose of contributing to knowledge in the study to expand the final results. So, the results show that 7 studies claimed to improve approaches and techniques where 12 primary studies contributed knowledge and suggests some appropriate solution frameworks mentioned in Table 8. It will help the testing industry to use existing techniques and approaches, models and frameworks for microservice testing.

It is observed in Fig 8 that there are some research trends like inclusive literature review, presented results, comparison with previous studies and claims to improve outcomes have not been satisfactory. Most of the article's study proposed solutions. However, there is a lack of assessment in selected studies to concentrate more in conducting experiments before proposing appropriate solutions for microservice testing. Furthermore, in proposed approaches,

there is a lack of assessment and standard research results to make the testing quality better. Hence, further case studies are needed to explore the real-time systems in microservice testing by following quality aspects.

Table 8- Major Comparison of Quality Studies

Selected Primary Study	Inclusive literature review	Proposed approaches	Conducted Experimentation	Presented Outcomes	Associated outcome with earlier studies	Claimed to Expand Results	Contributed to knowledge
[2]	x	✓	x	x	x	✓	✓
[3]	✓	✓	x	✓	x	✓	✓
[4]	x	x	✓	x	x	x	✓
[5]	✓	✓	✓	x	x	✓	✓
[6]	x	✓	✓	✓	x	✓	✓
[7]	✓	✓	✓	✓	x	x	✓
[8]	x	x	x	x	x	✓	✓
[10]	x	✓	x	x	x	✓	✓
[12]	✓	✓	✓	✓	✓	x	✓
[14]	x	✓	✓	x	x	x	x
[15]	x	✓	✓	x	x	x	x
[16]	x	x	✓	✓	x	x	x
[17]	x	x	x	x	x	x	✓
[18]	✓	x	✓	✓	x	✓	✓
[21]	✓	✓	✓	✓	x	x	✓

With the purpose of analysis of major comparison in quality studies, Fig 8 shows the trend of the major contribution of quality studies which is explained in the summary presented in Table 8.

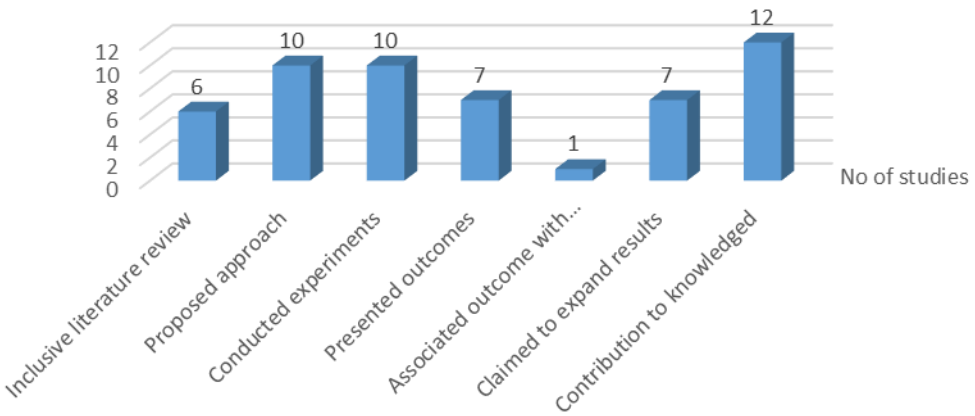

Figure 8 - Major Contribution of Quality Study

3.5 Quality Aspects

Consistent with RQ2, the following is the detailed explanation of each quality attribute to make the microservice testing better. There might be some other quality attributes for analysis inclusive of literature review. However, according to our selected primary study, we are considering said quality aspects as shown in Table 8. This is to understand which study provides better quality for testing microservice results in order to provide better resolution to testers to make the service improved for end users.

Microservice is a combination of interfaces, which motivates the use of different suggested approaches and methods, frameworks or tools under the defined quality attributes. Based on methods that have been defined in Table 8, there are differences in the Comparison between Previous Studies SLR's from primary study and Current Study. We uncovered that previous SLR's did not cater to all parameters mentioned above. We have seen that selected common issues as shown in Fig 5 associated with RQ1 while testing were not discussed in their study, while some studies are

providing detailed experiments and approaches as a solution using existing techniques as explained in Table 7. In order to provide better service we need to cater to all existing quality parameters to make the quality of microservice testing better. However, we suggest following the implications to improve the microservice quality.

1. The researcher will obtain awareness of different Microservice testing approaches in the testing environment.
2. The solutions so far developed for microservice testing using different frameworks, model and tools will help testers in implementing while testing microservices Security, Performance, Traceability, Compatibility, Complexity, Effectiveness and Scalability.
3. Use of tools and frameworks increase the transparency in microservice testing activities.
4. The positive and negative influence of microservice testing will help professionals and industries in the adoption of testing frameworks, models and tools.
5. In microservice testing, tools and frameworks support have an essential role.
6. Different testing approaches, models, frameworks have a vital role in microservice testing.
7. The SLR helps the testers in the selection of frameworks, tools, models, testing environment required to establish microservice testing.

4. Threats to Validity

The SLR presents the classification of solution for microservice testing and the different approaches to solve the parameters mentioned in Table 6. There is a chance of bias and validity threats during selection of primary study and analysis or conclusion level of the SLR work [23] , though the consequence of SLR study is reliable. The literature, by avoiding the bias step of collection primary study or search strategy, identifies the maximum possible studies available. Due to some gray literature, there is possibility to miss some studies, for example contribution of solution approaches and relationship between primary study, testing techniques and testing levels. This is due to the selected primary study communities such as software engineering, quality assurance and microservice testing. The method of isolating primary studies is recommended by the Cochrane Reviewers Handbook [36] and suggested by the study [23] considering bias, internal validity and external validity. Based on the implementation of existing techniques to conduct the review [37], the study has also been validated. The transparency of the study will allow the researchers to judge the consistency [38] of outcomes objectively.

The outcome of SRL indicates that justification research is appropriate in all types of contribution as shown in Table 8. The assessment includes practical case studies and is capable to increase the assurance of researcher and testers reports to reduce testing time and cost in this capacity [39].

5. Conclusion

Microservice testing is a crucial issue that should be deliberated carefully. The testing must be broad, inclusive and automated to entirely essential levels (unit, component, integration, contract and end to end testing). In this paper, selected microservice testing approaches in primary study are deliberated based on quite a few known important quality assurance parameters. Microservice quality parameters are tested through different approaches, techniques, frameworks, models and tools. Among these major influences are seven parameters which are security, performance, traceability, compatibility, complexity, effectiveness and scalability. Along with comprehensive survey, 70% of microservice testing performed are approach based, hence the better-quality assurances are those which are established on some model and framework. Such approaches and models distributed in a better way the utilization of microservice testing data with respect to the different functions and not functional requirements of the system. For quality assurance of microservice, testing at three main levels, such as an end to end, service to service and interface to interface testing is important due to their exceptional structure. This is because the most concerned factor is the user who needs a better GUI with new alteration and provide security, performance, traceability, compatibility, complexity, effectiveness and scalability with user machine. The other important aspects which require more focus include auditability, reliability, interoperability, flexibility and accuracy which should cater for a better quality of microservices. This study can be further improved by adapting remaining parameters by the increasing the number of selected studies and by covering all the digital databases. Furthermore, it can be prolonged by including testing on the remaining concerns of interoperability, auditability, integrity and inconsistencies in microservices.

References

[1] D. Nagothu, R. Xu, S. Y. Nikouei, and Y. Chen, “A Microservice-enabled Architecture for Smart Surveillance using Blockchain Technology,” 2018.

[2] N. Alshuqayran, N. Ali, and R. Evans, “A systematic mapping study in microservice architecture,” *Proc. - 2016 IEEE 9th Int. Conf. Serv. Comput. Appl. SOCA 2016*, pp. 44–51, 2016.

[3] H. Vural, M. Koyuncu, and S. Guney, “Computational Science and Its Applications – ICCSA 2013,” vol. 7971, pp. 203–217, 2013.

[4] Y. Sun, S. Nanda, and T. Jaeger, "Security-as-a-service for microservices-based cloud applications," *Proc. - IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2015*, pp. 50–57, 2016.

[5] D. Yu, Y. Jin, Y. Zhang, and X. Zheng, "A survey on security issues in services communication of Microservices-enabled fog applications," *Concurr. Comput.* , no. January, pp. 1–19, 2017.

[6] V. Heorhiadi, S. Rajagopalan, H. Jamjoom, M. K. Reiter, and V. Sekar, "Gremlin: Systematic Resilience Testing of Microservices," *Proc. - Int. Conf. Distrib. Comput. Syst.*, vol. 2016–Augus, pp. 57–66, 2016.

[7] A. de Camargo, I. Salvadori, R. dos S. Mello, and F. Siqueira, "An architecture to automate performance tests on microservices," *Proc. 18th Int. Conf. Inf. Integr. Web-based Appl. Serv. - iiWAS '16*, pp. 422–429, 2016.

[8] A. Carrasco, B. van Bladel, and S. Demeyer, "Migrating towards microservices: migration and architecture smells," *Proc. 2nd Int. Work. Refactoring - IWoR 2018*, no. June, pp. 1–6, 2018.

[9] TOGAF, *Microservices Architecture*. 2016.

[10] D. Taibi, V. Lenarduzzi, C. Pahl, and A. Janes, "Microservices in agile software development: a workshop based study into issues, advantages, and disadvantages," *Proc. XP2017 Sci. Work. - XP '17*, no. October, pp. 1–5, 2017.

[11] T. Cerny, M. J. Donahoo, and M. Trnka, "Contextual understanding of microservice architecture," *ACM SIGAPP Appl. Comput. Rev.*, vol. 17, no. 4, pp. 29–45, 2018.

[12] J. Soldani, D. A. Tamburri, and W. J. Van Den Heuvel, "The pains and gains of microservices: A Systematic grey literature review," *J. Syst. Softw.*, vol. 146, pp. 215–232, 2018.

[13] A. Chaturvedi and A. Gupta, "A tool supported approach to perform efficient regression testing of web services," *c2013 IEEE 7th Int. Symp. Maint. Evol. Serv. Cloud-Based Syst. MESOCA 2013*, pp. 50–55, 2013.

[14] S. P. Ma, C. Y. Fan, Y. Chuang, W. T. Lee, S. J. Lee, and N. L. Hsueh, "Using Service Dependency Graph to Analyze and Test Microservices," *Proc. - Int. Comput. Softw. Appl. Conf.*, vol. 2, pp. 81–86, 2018.

[15] K. Bakshi, "Microservices-based software architecture and approaches," *IEEE Aerosp. Conf. Proc.*, pp. 1–8, 2017.

[16] W. Hasselbring and G. Steinacker, "Microservice architectures for scalability, agility and reliability in e-commerce," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, pp. 243–246, 2017.

[17] A. N. Exploratory and C. Study, "Learning-Based Testing of Microservices AN EXPLORATORY CASE STUDY USING Learning-Based Testing of Microservices," 2015.

[18] A. Arcuri, "RESTful API automated test case generation," in *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, 2017, pp. 9–20.

[19] M. Vianden, H. Lichter, and A. Steffens, "Experience on a Microservice-based Reference Architecture for Measurement Systems," *2014 21st Asia-Pacific Softw. Eng. Conf.*, vol. 1, pp. 183–190, 2014.

[20] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic Mapping Studies in Software Engineering," pp. 1–10, 2007.

[21] M. Rahman and J. Gao, "A reusable automated acceptance testing architecture for microservices in behavior-driven development," *Proc. - 9th IEEE Int. Symp. Serv. Syst. Eng. IEEE SOSE 2015*, vol. 30, pp. 321–325, 2015.

[22] D. Moher *et al.*, "Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement," pp. 1–9, 2015.

[23] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," vol. 80, no. 4, pp. 571–583, 2007.

[24] D. Budgen and P. Brereton, "Performing Systematic Literature Reviews in Software Engineering," pp. 1051–1052, 2006.

[25] D. Savchenko and G. Radchenko, "Microservices validation: Methodology and implementation," *CEUR Workshop Proc.*, vol. 1513, pp. 21–28, 2015.

[26] S. Hassan and R. Bahsoon, "Microservices and their design trade-offs: A self-adaptive roadmap," *Proc. - 2016 IEEE Int. Conf. Serv. Comput. SCC 2016*, pp. 813–818, 2016.

[27] M. J. Kargar and A. Hanifizade, "Automation of regression test in microservice architecture," *2018 4th Int. Conf. Web Res. ICWR 2018*, pp. 133–137, 2018.

[28] W. Ha, "Reliability Prediction for Web Service Composition," pp. 7–10, 2017.

[29] G. Cherradi, A. El Bouziri, and A. Boulmakoul, "A Generic microservice-based architecture for Smart HazMat Transportation A Generic microservice-based architecture for Smart HazMat 1 Introduction," no. November, 2017.

[30] X. Wan, X. Guan, T. Wang, G. Bai, and B. Choi, "Journal of Network and Computer Applications Application deployment using Microservice and Docker containers : Framework and optimization," *J. Netw. Comput. Appl.*, vol. 119, no. May, pp. 97–109, 2018.

[31] J. P. Gouigoux and D. Tamzalit, "From monolith to microservices: Lessons learned on an industrial migration to a web oriented architecture," *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, pp. 62–65, 2017.

[32] D. Jaramillo, D. V. Nguyen, and R. Smart, “Leveraging microservices architecture by using Docker technology,” *Conf. Proc. - IEEE SOUTHEASTCON*, vol. 2016–July, pp. 1–5, 2016.
[33] O. Zimmermann, “Microservices in Practice, Part 1,” vol. 2, no. 1, pp. 91–98, 2016.
[34] W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand, “Using contract-based component specifications for virtual integration testing and architecture design,” *2011 Des. Autom. Test Eur.*, pp. 1–6, 2011.
[35] S. Segura, J. A. Parejo, J. Troya, and A. Ruiz-Cortes, “Metamorphic testing of RESTful Web APIs,” *IEEE Trans. Softw. Eng.*, vol. 44, no. 11, pp. 1083–1099, 2018.
[36] M. Van Tulder, A. Furlan, and C. Bombardier, “Updated Method Guidelines for Systematic Reviews in the Cochrane Collaboration Back Review Group,” vol. 28, no. 12, pp. 1290–1299, 2003.
[37] N. Niknejad, Y. A. Prasetyo, I. Ghani, and A. A. N. Fajrillah, "Service Oriented Architecture Adoption: A Systematic Review," International Journal of Integrated Engineering, vol. 10, no. 6, 2018.
[38] M. D. M. Suffian, D. N. A. Jawawi, R. R. Saedudin, and M. A. Isa, "Towards Formulating Dynamic Model for Predicting Defects in System Testing using Metrics in Prior Phases," International Journal of Integrated Engineering, vol. 10, no. 6, 2018.
[39] M. F. Darmawan, N. I. Jamahir, R. R. Saedudin, and S. Kasim, "Comparison between ANN and Multiple Linear Regression Models for Prediction of Warranty Cost," International Journal of Integrated Engineering, vol. 10, no. 6, 2018.