Multi-tenancy in Cloud-native Architecture: A Systematic Mapping Study

DANIEL OLABANJI, TINEKE FITCH, OLUMUYIWA MATTHEW
University of Portsmouth, Portsmouth,
UNITED KINGDOM

Abstract: - Cloud-native architectures has become an essential part of the cloud computing paradigm with the capacity of improved horizontal and vertical scalability, automation, usability and multi-tenancy. However, there are parts that are yet to be fully discovered like multi-tenancy. Multi-tenancy an essential part of the cloud computing, has not been fully. The purpose of this study is to survey existing research on multi-tenancy in cloud-native architecture in order to identify useful trends, opportunity, challenges and finally the needs for further researches. A systematic mapping method was used to systematically compare, classify, analyse, evaluate and appraise existing works of literature on multi-tenancy in cloud-native. We started from over 921 potentially relevant peer reviewed publications. We applied a selection procedure resulting in 64 peer reviewed publications over the last six years between 2015 to 2022 and the selected studies were classified through the characterisation framework. The review shows the emerging challenges and trending concepts in multi-tenancy within cloud native architecture, but also discusses the improvement in multi-tenancy while considering cloud native architecture in the recent years.

Key-words: Cloud-Native, Multi-tenancy, Isolation, Cloud Computing, Systematic Mapping study.

Received: March 26, 2022. Revised: January 2, 2023. Accepted: February 5, 2023. Published: March 7, 2023.

1 Introduction

Cloud computing has several essential characteristics that make it more robust and attractive to several users. The technology enables the pay-per-use business model and moves local storage to cloud-based storage for average internet users and almost every commercial entity, [1]. Cloud computing is classified into private, public, hybrid, and community deployment models. These deployment model classifications are based on the infrastructure's ownership, management, and operation. Similarly cloud computing is also classified by service models namely: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [2], and each service model has different management and control that makes each unique. The SaaS delivers applications that are accessible to different users online, although, the user does not manage or control the underlying cloud infrastructure. While in the PaaS, the user has access and controls their data, the application, and the application development lifecycle, without control over the infrastructure. Whereas in the IaaS the user owns and manages the applications, data, operating system and application runtime. The cloud service model is thus, the classification of the services that a service provider can offer in a public cloud. This study focused mainly on the IaaS model and private cloud deployment model, where both multi-tenancy and virtualisation are implemented.

Multi-tenancy is an architectural tactic to increase cost-efficiency by sharing the available resources maximally among several users, [3]. According to [4], there are three methods for achieving multi-tenancy in cloud computing: using a database, virtualisation, and physical separation. Of the three options, virtualisation is the most used options in achieving multi-tenancy in cloud computing. Virtualisation is achieved in cloud computing through the implementation of either virtual machines or containers [5]. In a virtual machine (VM) setup, multi-tenancy is achieved through the use of a hypervisor, which allows service providers, developers, engineers and designers to make a single instance of an application, hardware, middleware, and database to be shared between several entities by isolating each tenant from the others [5]. However, this comes with several limitations such as; resource management, scalability and lack of automation and to solve these issues another virtualisation technology was introduced.

The industry developed container virtualisation to provide on-demand scalability, optimal resource usage, fault tolerance, and automation. Container virtualisation has been seen as an alternative to VMs in the IaaS cloud computing development model, and it is becoming a vital part [7]. Containers are lightweight virtualisation that makes use of fewer resources and less time to provide scalable, portable and interoperable applications in cloud computing [6]. The container virtualisation, Microservice, DevOps and other improvements in cloud computing architecture have made the term 'cloud-native' popular in industries and academic.

It is noteworthy that the term cloud-native was early mentioned at the early stage of cloud computing to means applications developed solely for cloud computing [8]. However, as more ideas and innovation emerged, the term cloud-native gained a more comprehensive meaning and popularity from the year 2015 [9] to mean cloud architecture that uses microservice and containers virtualisation to provide a scalable application further details in Section 2.1. However, this improvement comes with issues of implementing multi-tenancy specifically the sharing of available resources in cloud-native architecture environment. The understanding of performance bottlenecks in multi-tenancy based cloud-native environment is critical in achieving performance improvements in cloud native adoptability, application level fairness and resource management, [7].

Kubernetes which is the main container orchestration system lacks sufficient multi-tenant supports by design, [10]. Furthermore, the inadequate support for multi-tenancy during the development of kubernetes orchestrator brings the lack of guaranty secure isolation between tenant, [11], diminish the benefits of cloud computing and makes is difficult to adopt. In order to provide detail understanding of multi-tenancy in cloud-native architecture and trends, this study conducted a systematic mapping study which intended to identify, evaluate and summarise the findings about multi-tenancy in cloud-native architecture. Which will help to provide more insight on the existing academic work on cloud-native architecture, its challenges and improvement. Sixty-four peer-reviewed publications were methodically selected years ranging from 2015 to 2022 in which cloud-native architecture and multi-tenancy are mentioned and discussed in their topic, keywords and metadata from different online academic database.

The remainder of this paper is structured as follows: Section 2 describes the background and related research. Section 3 explains the research methodology, research question and screening; Section 4 provides a classification scheme, followed by the mapping; Section 5 discusses findings, and Section 6 provides the review's conclusions.

2 Multi-tenancy

Multi-tenancy is a computing architectural concept concerned with information sharing among multiple users referred to as tenants. In case cloud computing, improvement of resource utilisation and service availability in cloud computing are based on multi-tenancy, [12], [13]. [12], explained that multi-tenancy is a concept that enables sharing the same service instance, scaling up and down the resources allocated among different tenants. Both characteristics improve resource utilisation, cost and service availability. In addition, apart from the capacity of multi-tenancy to share resources as a strategy in cloud computing, it also enables service providers to maximise resource utilisation and, thus, reduce the servicing costs per tenant, [14]. In the database perspective, multi-tenancy as a principle where a single instance of the DBMS runs on a server, serving multiple clients (tenants), [15]. The multi-tenancy in database systems supports several separate and distinct groups of users, the users are referred to as tenants.

Multi-tenancy can thus be defined as an architectural concept that makes resource sharing possible and enforce isolation between tenants. These tenants could be applications, users, physical or virtual infrastructure or systems. Furthermore, as [16], observed, multi-tenancy is an essential property of cloud computing that optimises resource utilisation by allowing multiple consumers and multiple workloads to share computing and network infrastructure using virtualisation technology. Multi-tenancy in cloud computing has its challenges: the lack of filtration of the inside part of the servers because both the client and the attackers are in the same server [13], access control and resource allocation. Multi-tenancy enables computation instances from different tenants running on the same physical server in an IaaS service model [17]. Those identified challenges affecting multi-tenancy were transferred into the cloud-native architecture.

2.1 Cloud-native Architecture (CNA)

Cloud-Native Computing Foundation (CNCF), the sole convener of the architecture, defines cloud-native as a set of technologies that empower organisations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. The approach is exemplified by "containers, service meshes, microservices, immutable infrastructure, and declarative APIs". CNCF further claims that "these techniques enable loosely coupled systems that are resilient, manageable, and observable combined with robust automation, that allow engineers to make

high-impact changes frequently and predictably with minimal toil" , [18]. In [19], it is observed that cloud-native technologies are used to develop applications built with services packaged in containers deployed as microservices and managed on elastic infrastructures through agile DevOps processes and continuous delivery workflows.
To further understand CNA and containers, there is a need to briefly discuss about 'Microservice'. Microservice architecture divides applications into smaller self-contained components, called microservice and the microservice serves specific business functions and communicates via lightweight language-agnostic APIs, [20]. Microservice architecture necessitates a virtualisation technique that can provide a better level of isolation, scalability, deployment, updating and elastic resources; these can be achieved using a different virtualisation technique than the virtual machine. These needs produce the success of the containers virtualisation mechanism which are now widely used in cloud computing environment and configuration. Containers virtualise the operating system and spin up multiple containers within milliseconds, whereas, virtual machine are based on running software on physical hardware to simulate physical computer and different operating systems need to be installed on virtual devices, [6]. However, as discuss above the CNA consist of the container virtualisation and microservice architecture and need to be study to shows the gaps and add to knowledge in cloud-native architecture.

3 Research Methodology

In this study, systematic mapping study method was adopted to ascertain the trends, research areas and challenges of multi-tenancy in CNA. A Systematic Mapping Study (SMS) is useful in determining the structure of the study in a research area where there is a lack of high-quality primary papers. Systematic mapping as an approach designed to give an overview of a research area through classification and quantification of contributions in the categories arising [21]. A systematic mapping study helps to achieves a broad review of primary studies in a specific topic area and to identify the available evidence and research gaps.

This study follows the guidelines for systematic mapping in software engineering which provides insight through a rigorous and methodical approach to searching, classifying literature and extracting evidence with analytical evaluation [21]. Based on the above definition and description of the SMS, it is a suitable method for the topic area. There are other literature review methods, but they lack the systematic analysis method in SMS. Table 1 presents the step-by-step process taken to achieve the research outcome and present the contribution in a methodical way.

Table 1. Systemic Mapping process

Process step	Outcomes	Section
Research question	Review scope	3.1
Conduct search	Gather all papers	3.2
Screening of papers	Relevant papers	3.3
Topic Classification	Classification and Keywording	4.1
Data extraction and mapping	Systematic mapping	4.2

3.1 Research Questions

The following research questions were answered in this study and they are methodically formulated to guide the study.

[RQ1]: What does the term 'multi-tenancy' in cloud-native architecture mean?
[RQ2]: What are the existing trends in cloud-native multi-tenancy?
[RQ3]: What are the foreseeable challenges of multi-tenancy in cloud-native architecture?

3.2 Conducting the Primary Search

Publications and papers were extracted from relevant electronic databases through the use of search terms and keywords. The selection of search terms were drew on the keyword identification guidance of [21]. This is done by grouping keywords and synonyms to formulate a search string that covered a larger space, unbiased and avoid incompleteness. Similar relative terms were used to construct the search string. Research papers such as [9] have considered key 'cloud-native' search words. Surprisingly, several other papers use 'containers' to represent CNA. Although there are other virtualisation techniques in cloud computing, container technology is the virtualisation method used in the CNA, hence, including 'container' in the search string. Also, some of the literature could be under-represented without the addition of the word. The search string used to initiate this study was:

("Cloud native" OR "cloud-native" OR "container" OR "native cloud") AND ("multi-tenant" OR "multi-tenancy")

This study considered the PICO (Population, Intervention, Comparison and Outcome) keyword formulation search approach. Table 2 shows the online databases used and the 921 papers generated from the online database.

Table 2. Papers considered from the initial search

Sources	***No.***	***Oldest***	***New***	***Accessed***
IEEE	64	2015	2022	4.9.2022
ScienceDirect	360	2015	2022	7.9.2022
ACM	159	2015	2022	7.9.2022
SpringerLink	338	2015	2022	7.9.2022
Total	921			

3.3 Literature Screening

The screening mechanism was applied for scrutinising the selected papers with inclusion and exclusion criteria. The design and methodological guidelines from [21] were used to screen the literature. Table 3 shows both the inclusion and the exclusion criteria that were applied.

Table 3. Literature screening criteria

Inclusion	• Journals or conferences that are peer-reviewed and researching multi-tenancy and cloud-native or cloud-native architecture. • Studies published between year 2015 to 2021
Exclusion	• Studies that did not relate to cloud computing • Studies that were not presented in English • Studies that were not accessible in full text • Studies that were non-peer-reviewed

The literature screening criteria helped remove the papers that did not focus on cloud-native and multi-tenancy and those that were not related to the inclusion criteria. Likewise, a specific year gap was selected as the target years. The reason for these target years has been discussed earlier in this publication. Based on the criteria in Table 3 and publications indexed in multiple databases resulting in duplication, 867 publications were removed. Which bring the remaining categorises 64 publications comprised only peer-reviewed publications. The selected papers can be found in the Table 4 with classifications. Also, publication in the table was not sorted to avoid any bias. The second and third authors contributed to this work by providing a quality performance control to each of the section in the research methodology.

4 Keywording and Topic Classification

4.1 Topic Classification

This process create a classification scheme for clear understanding and straightforward design of the systematic mapping. The selected papers are classified into two main categories:

- Topic-independent classification
- Topic-specific classification

4.1.1 Topic-independent Classification

Topic-independent classification classifies the papers based on the research approach and not by topic or keyword. According to [21], adopting an existing classification scheme in a systematic mapping study is advisable. This study drew on the classification proposed by [84], with a few adjustments to their categories. For instance, [9] suggested that philosophical papers are rare in software engineering. Thus, this study replaced philosophical category with the survey category to provide a clearer classification. Details of this classification of the selected papers can be found in Table 4.

Table 4. Included and categorised publications

Study No	***Title***	***Year***	***Type***	***Validation***	***Evaluation***	***Solution***	***Survey***	***Option***	***Experience***
S1	Evaluating the Effect of Multi-tenancy Patterns in Containerized Cloud-hosted content management system[22]	2018	Conference		X	X		X	X
S2	Scone: Secure Linux Containers with intel SGX[23]	2016	Conference	X	X	X			X
S3	Leveraging Kernel security Mechanisms to improve container security: a survey[24]	2019	Conference				X	X	
S4	KubeSphere: An Approach to Multi-Tenant Fair Scheduling for Kubernetes Clusters[25]	2019	Conference	X	X	X			X
S5	KubeSphere: An Approach to Multi-Tenant Fair Scheduling for Kubernetes Clusters [26]	2018	Conference				X	X	
S6	A Case for Performance-Aware Deployment of Containers [27]	2019	Journal		X	X			X
S7	Design and Implementation of Multi-tenant Vehicle Monitoring Architecture Based on Microservices and Spark Streaming [28]	2020	Conference		X	X		X	
S8	Preemptive and low latency Datacenter Scheduling via lightweight Containers [29]	2019	Journal		X			X	X
S9	Multi-tenant utility computing with compute containers. [30]	2015	Conference		X	X			X
S10	Studying the Applicability of Intrusion Detection to Multi-tenant Container Environments [31]	2019	Conference		X	X			X
S11	Using Attack Injection to Evaluate Intrusion Detection Effectiveness in Container-based Systems [32]	2020	Conference	X		X			X
S12	A Study on the Security Implications of Information Leakages in Container Clouds [33]	2018	Journal			X			X
S13	ContainerLeaks: Emerging Security threats of Information Leakages in Container Clouds [34]	2017	Conference	X	X	X			X
S14	Houdini's Escape: Breaking the resource rein of linux control groups [35]	2019	Conference		X	X			X
S15	Towards a Taxonomy of Microservices Architectures [36]	2017	Conference			X	X		
S16	Singularity: Simple,secure containers for compute-driven workloads [37]	2019	Conference		X				
S17	Software-defined object storage in multi-tenant environments [38]	2019	Journal	X		X			X
S18	SCoPe: A Decision System for Large Scale Container Provisioning Management [39]	2016	Conference			X		X	X
S19	An Improved Kubernetes Scheduling Algorithm for Deep Learning Platform [40]	2020	Conference	X		X			X

Table 4. Included and categorised publications

Study No	***Title***	*Year*	*Type*	*Validation*	*Evaluation*	*Solution*	*Survey*	*Option*	*Experience*
S20	PARES: Packet Rewriting on SDN-Enabled Edge Switches for Network Virtualization in Multi-Tenant Cloud Data Centers [41]	2017	Conference	X	X	X			X
S21	Energy efficiency comparison of hypervisors [42]	2019	Journal	X					X
S22	A framework for black-box SLO tuning of multi-tenant applications in Kubernetes [43]	2019	Conference	X	X	X		X	X
S23	Securing Cloud Containers Using Quantum Networking Channels [44]	2016	Conference	X	X	X		X	X
S24	Native Cloud Applications:Why Monolithic Virtualization Is Not Their Foundation [45]	2017	Conference				X	X	
S25	Customizing Multi-Tenant SaaS by Microservices: A Reference Architecture[46]	2019	Conference			X			X
S26	Thread-level resource consumption control of tenant custom code in a shared JVM for multi-tenant SaaS [47]	2021	Journal		X				X
S27	Evaluation of Virtualization and traffic filtering methods for container networks [48]	2019	Journal		X				X
S28	The Nas Benchmark Kernels for Single and Multi-tenant cloud Instances with LXC/K [49]	2018	Conference		X	X			X
S29	Reprint:Legiot: A Lightweight Edge Gataway for the Internet of Things [50]	2019	Journal	X	X	X			X
S30	Using Microservices for Non-intrusive Customization of Multi-tenant Saas [51]	2019	Conference	X	X			X	X
S31	Service-oriented Multi-tenancy (SO-MT): Enabling Multi-tenancy for Existing Service Composition Engines with Docker [52]	2016	Conference		X	X		X	X
S32	An Open Sharing PatternDesign of Massive Power Big Data [53]	2019	Conference		X	X		X	X
S33	Docker Cluster Management for the Cloud - Survey Results and Own Solution [54]	2016	Journal				X	X	
S34	Profiling distribution systems in lightweight virtualized environments with logs and resource metrics [55]	2018	Conference		X				X
S35	A Machine Learning Model for Detection of Docker-based APP Overbooking on Kubernetes [56]	2021	Conference	X		X			X
S36	SynAPTIC: Secure And Persistent connecTIvity for Containers [57]	2017	Conference	X	X	X			X
S37	Scheduling dynamic workloads in multi-tenant scientific workflow as a service platforms [58]	2018	Journal	X		X			X

Table 4. Included and categorised publications

Study No	Title	Year	Type	Validation	Evaluation	Solution	Survey	Option	Experience
S38	Right Scaling for Right Pricing: A Case Study on Total Cost of Ownership Measurement for Cloud Migration [59]	2019	Conference				X		
S39	Containers and Virtual Machines at scale: A Comparative study [60]	2016	Conference	X	X	X			X
S40	Building a multi-tenant cloud service from legacy code with Docker containers [61]	2015	Journal			X			X
S41	Cloud Native Databases: An Application Perspective [62]	2017	Conference			X	X		X
S42	Towards Vulnerability Assessment as a Service in OpenStack Clouds [63]	2016	Conference	X	X	X			X
S43	SWITCH-ing from multi-tenant to event-driven video conferencing services [64]	2017	Workshop		X			X	X
S44	Towards a container-based architecture for multi-tenant SaaS applications [3]	2016	Workshop				X	X	
S45	Performance overhead of container orchestration frameworks for management of multi-tenant database deployments [65]	2019	Conference		X		X		X
S46	Network Virtualization: Proof of Concept for Remote Management of Multi-Tenant Infrastructure [66]	2020	Conference	X		X			X
S47	Challenges for Building a Cloud Native Scalable and Trustable Multi-tenant AIoT Platform [67]	2020	Conference	X	X	X			X
S48	A Latency-driven Availability Assessment for Multi-Tenant Service Chains [68]	2022	Journal		X	X			X
S49	A Multi-Tenant Framework for Cloud Container Services [10]	2021	Journal		X	X			X
S50	A Secure Container Placement Strategy Using Deep Reinforcement Learning in Cloud [69]	2022	Conference			X		X	X
S51	Advocating isolation of resources among multi-tenants by containerization in IaaS cloud model [70]	2017	Conference	X	X				X
S52	Container-Based Service Chaining: A Performance Perspective [71]	2016	Conference	X	X	X			X
S53	Containers Resource Allocation in Dynamic Cloud Environments [72]	2021	Conference		X	X			X
S54	Enhancing Proportional IO Sharing on Containerized Big Data File Systems [73]	2021	Journal	X	X	X			X
S55	Framework for Analysing a Policy-driven Multi-Tenant Kubernetes Environment [74]	2021	Conference			X	X		X
S56	Improving the Security of Microservice Systems by Detecting and Tolerating Intrusions [75]	2020	Workshop						X
S57	Migrating Monoliths to Microservices-based Customizable Multi-tenant Cloud-native Apps [76]	2021	Journal	X	X	X		X	
S58	Feasibility of container orchestration for adaptive performance isolation in multi-tenant SaaS applications [77]	2020	Conference	X	X	X		X	X

Table 4. Included and categorised publications

Study No	*Title*	*Year*	*Type*	*Validation*	*Evaluation*	*Solution*	*Survey*	*Option*	*Experience*
S59	Deep customization of multi-tenant SaaS using intrusive microservices [78]	2018	Conference			X			X
S60	EdgeNet: A Multi-Tenant and Multi-Provider Edge Cloud [79]	2021	Workshop		X				X
S61	LogStore: A Cloud-Native and Multi-Tenant Log Database [80]	2021	Conference	X		X			X
S62	Multi-Tenant Machine Learning Platform Based on Kubernetes [81]	2020	Conference		X	X			X
S63	Performance Evaluation of Container-Level Anomaly-Based Intrusion Detection Systems for Multi-Tenant Applications Using Machine Learning Algorithms [82]	2021	Conference	X	X	X			X
S64	Reinforcement Learning for Resource Management in Multi-tenant Serverless Platforms [83]	2022	Conference		X	X			X

4.1.2 Topic-specific Classification

This research did not achieve topic-specific classification through the IEE, ISO/IEC Swebok classification [85], so the researchers generated their topic-specific classification. In agreement with [21], who stated that most mapping studies design their classification scheme. For this purpose, developing a bespoke topic-specific classification was decided. Authors' keywords from our selected papers were extracted first, as most papers have keywords. For those without keywords, such as [3], [65], and [23], the approach of [21], was followed, which involved an adaptive reading of the abstract in picking the keywords, and when there is no abstract, reading the introduction and conclusion to provide valuable keywords for the study. The first reviewer extracted the data by considering the inclusive and exclusive criteria, and co-authors checked the outcome from the extracted data.

4.1.2.1 Steps used in Developing Topic-Specific Classification:

1. Extracted the keywords from each selected paper
   i. Identified those without keywords
   ii. Adaptively read the abstract, introduction, and conclusion to generate keywords for those without
2. Grouped the keywords cohesively
3. Aggregated the grouped keywords
4. Grouped the papers into each aggregated keyword.

4.2 Data Extraction and Mapping

The data extraction and mapping process is based on the above classification techniques to develop the keyword classification scheme. The data mapping is also divided into the same two types: topic-independent and topic-specific.

Data mapping simplifies the mapping system and answers the research questions

Category	Description
Validation	Validations are done for techniques that might be novel or have not yet been implemented in practice. These techniques are validated using experiments, i.e., work done in the lab.
Evaluation	Techniques are implemented in practice, and the method is evaluated. It shows how the process is executed (solution implementation) in practice and the consequences of the implementation evaluation.
Solution	A solution to a problem is proposed. The solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good argumentation line.
Survey	A survey reviews other primary or secondary studies relating to a specific research question to integrate/synthesise evidence associated with a particular research question.
Opinion	These papers express personal opinions on whether a specific technique is good or bad or how things should be done. They do not rely on related work and research methodologies.
Experience	Experience papers explain what and how something has been done in practice. It is related to the personal experience of the author.

4.2.1 Topic-Independent Data Extraction and Mapping

In this type of data mapping, the information considered is not closely related to the publications' keywords or content; instead, it includes other factors, such as the year of publication, the venue type, and the research approach.

4.2.1.1 Year of Publication

Firstly, the collated literature was separated into the publication years (see Figure 1). In 2015, just two publications considered multi-tenancy in CNA, while in the following year, there was an increase in the number, which remained steady until the year 2019. In 2019, there was another and more significant rise, possibly because there was more awareness of multi-tenancy needs and capabilities in cloud-native orchestration software, such as Kubernetes. The efforts put into such research through the CNCF, and other secondary foundations and organisations, focus on multi-tenancy in cloud-native computing. Furthermore, security and isolation have become critical in cloud-native research for 5G, IoT, and artificial intelligence technologies. Research into those technologies has increased since 2019.

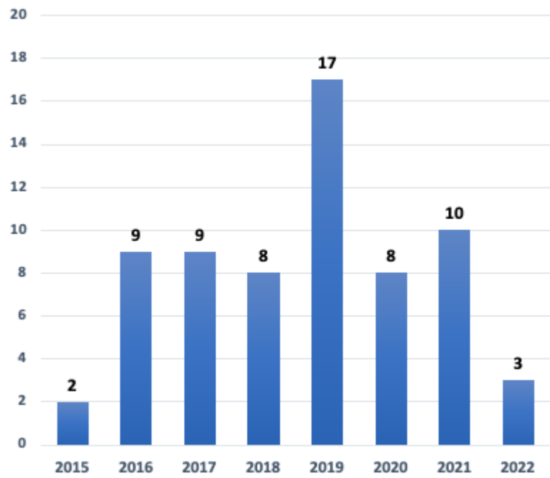

Fig. 1: Year of publication

(Note that the apparent significant drop in numbers after 2019 was due very probably to the impact of the Covid-19 global pandemic in 2020 and 2022)

4.2.1.2 Research Format of publications

The publication format is another criterion that can map the selected literature into different categories. Figure 2 below shows the contribution based on the format type in which the papers are published. The primary format of this study was conference papers, which stand at 74% of the total selected literature, while 22% were articles in journals, and only 4% were workshops.

Fig. 2: Research Format

4.2.1.3 Research Approach

The papers were grouped based on the contribution type (as noted in Table 4 above): experience, validation, evaluation, solution, survey, and opinion. Figure 3, below, visualises the categories and clearly shows that the minor contribution is the 'survey' type of research approach; more survey research is needed in this field and other research approaches.

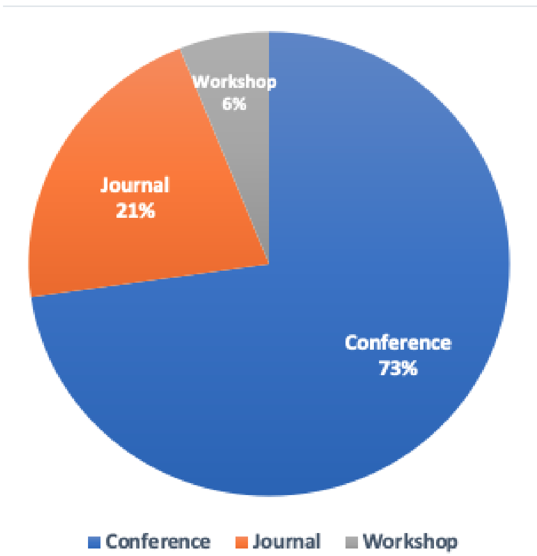

Fig. 3: Research categories

4.2.2 Topic-specific Data Extraction

This extraction answers the systematic mapping research questions and focuses on the selected papers' keywords and content. These are directly keyword-related extractions, the correlation of the research topics and research approaches.

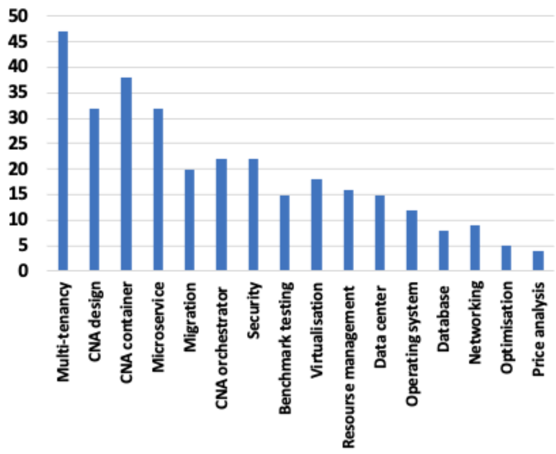

Fig. 4: Keywords extraction and frequency

4.2.2.1 Keyword Extraction and Frequency

Keyword extraction categorises the selected papers based on their topics. Another category was developed that mapped the selected papers into a group either because they had a similar research topic or similar keywords. This yielded 16 groups, as shown in Figure 4.

4.2.2.2 Main Research Topic

The keywords of the already collated and selected publications were further regrouped. Figure 5 visualises the main research topics from the selected papers. It could be observed that the majority of the research topic focused on architecture and less on multi-tenancy or security.

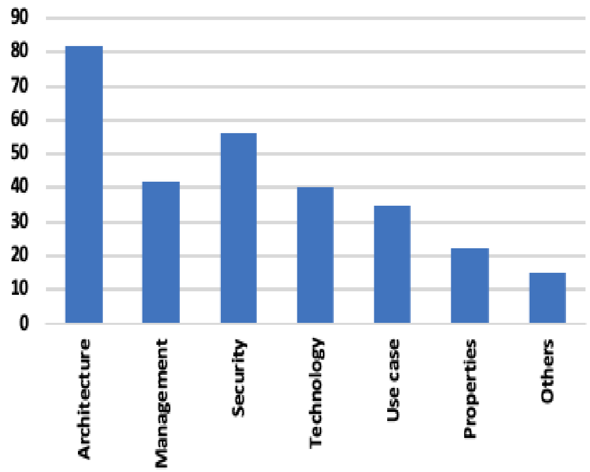

Fig. 5: Keywords grouping extraction and frequency

4.2.2.3 Correlation between Studies based on the Extracted Keywords

The correlation between studies and extracted keywords was done by correlating them to investigate linearly related or correlated. Figure 6 visualises the correlation between the keywords and the topics. It should be noted that none of the considered studies included topics concerning cloud quality. In addition, this keyword categorisation may not be detailed enough for use in some cases and detailed and specific analyses, such as cost-effectiveness, energy usage, forensics, or cybersecurity.

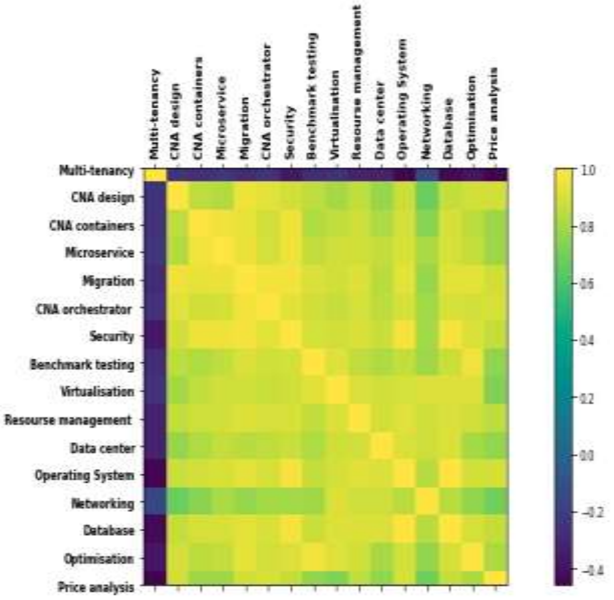

Fig. 6: Further keywords extraction and correlation

Figure 7 shows the mapping of the research approach to the research topic and the specific area to which the study contributes. Both figures show what has been done and can still be improved.

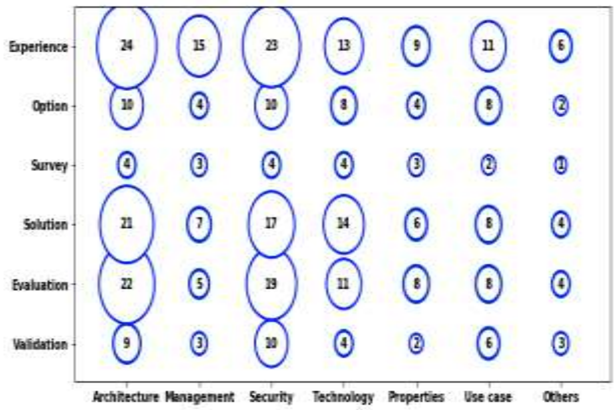

Fig. 7: Mapping the research approach to the research topics

4.3 Study Quality

The quality of the study is based on the selection of the contributing papers. The selected research papers, including journal articles and conference and workshop papers, were accessed from reputable databases and were peer-reviewed. Books, keynotes and magazines were not used in this research because they are often based on the writers' views or past reviews. This does not suggest that those materials do not contribute to research, but they did not meet the inclusion criteria here.

4.4 Threats of Validity

Combining cloud-native architecture and multi-tenancy raises validity threats, common in systematic mapping. The threats that might have affected this research include theoretical validity, which might have been evident in the study during data selection and extraction. This was reduced by allowing the second researcher to review the extraction, but, given human judgment, this kind of threat cannot be eliminated, [21]. Another threat is interpretive validity, which relates to the conclusion, and the value of the information provided. Data and information provided in the research interpretation may be biased because the study focuses on multi-tenancy in cloud-native architecture rather than other cloud computing properties.

The study used easy-to-read guidelines, suggested in [21], which help reduce personal and technical bias about the content and the systematic mapping data.

Descriptive validity could be less evident because it does not consider detailed observations and the studies' objectives. In addition, readers need to understand that it is possible to be biased in the research scope or focus here because the study targeted specific properties in the mapping. This may provide a bias in the general mapping of the study. This study focuses more on the security issues related to multi-tenancy in cloud-native architecture rather than the cost, energy usage and computer hardware usage.

5 Discussion

5.1 Research Question 1: What Does the Term Multi-Tenancy in Cloud-Native Architecture (CNA) Mean?

Cloud-native technology considers the isolation from the physical system, kernel-based orchestration, software development methodology (DevOps), and micro-servicing architecture. Many early academic papers such as (S40), (S39), and (S44) discussed how to migrate from the legacy cloud computing characteristics to the cloud-native paradigm. However, less of the research shows the challenges of the cloud-native paradigm when considering isolation

and sharing. Multi-tenancy in cloud-native architecture was explained in this research as an architecture that enables sharing the same service instance among different tenants, which is done by sharing hardware, instances, clusters, namespace, pods or microservices between different tenants. The research indicates that the contributions focus of multi-tenancy in cloud-native architecture falls mainly into the research approach categories' experiment' and 'solution'. (S1), (S2), (S4), (S6), (S7), and several other study papers, as indicated in Table 7 below, covered the experiment and solutions. In [9] the author indicated that the contribution type of their selected papers are mostly solution and experience papers. In agreement, [86], also show the same in their review of the Cloud container technologies review, which shows fewer opinion or survey types of contribution. To further understand multi-tenancy in cloud-native architecture, there is a need to understand the differences between multi-tenancy in cloud-native architecture and software orientation architecture. Both isolation and sharing in CNA can be addressed based on different forms of role, permission, and access, which are different in other software multi-tenancy. Table 6 shows the differences between software-oriented and cloud-native architecture, and these differences introduce the ideas about the classification of tenants more than the 'owner' and 'tenant' relationship.

Table 6. Sharing mode and Isolation in cloud computing

Sharing Mode	*Isolation*	*Software orientation*	*CNA*
Shared Hardware	VM	x	x
Shared VM	OS User	x	x
Shared OS	DB Instance	x	x
Shared instance	Database	x	x
Shared database	Scheme	x	x
Sharing network	Network	x	x
Sharing API	API		x
Sharing cluster	Node		x
Sharing container	Container		x
Sharing pod	Pod		x

According to [70], the container provides better isolation than the virtual machine. The virtual machine isolation is operating-system-based, while containers provide isolation at every instance of virtualisation, such as the process level, the file system level, the network level and the inter-process communication level. Multi-tenancy in CNA also provides enhanced sharing capability based on the provided isolation. Containers enable significant resource savings by isolating the application process while sharing part of the operating system such as the kernel, libraries and other processes concurrently running on the machine because it was built on top of namespaces (S15) and (S35).

5.2 Research Question 2: What are the Existing Trends in Cloud-Native Multi-Tenancy?

CNA properties, such as elasticity, auto-scaling, horizontal and vertical scaling and automation, and including multi-tenancy, bring several trending characteristics and technologies to empirical research. For this reason, this research question considered cloud-native multi-tenancy with other trending concepts. Table 6 identifies some of the trending concepts found in the research. Readers should note that these were not necessarily the only trending considerations at the research time. Table 7 serves to identify a few of the critical examples related to this research and the combination of both cloud-native and multi-tenant architecture.

5.3 Research Question 3: What are the Foreseeable Challenges of Multi-Tenancy in Cloud-Native Architecture?

The challenges of multi-tenancy in CNA are inherited from the SOA multi-tenancy issues and some additional challenges. Challenges such as the incomplete implementation of system resource isolation mechanisms in the Linux kernel posed security concerns for multiple container sharing in an operating system kernel (S12), (S35). Listed below are some of the challenges of multi-tenancy in CNA.

Table 7. Trending concepts

Trending concepts	Definition
Hard multi-tenancy	Hard multi-tenancy means that multiple tenants in the same cluster should not have access to anything from other tenants [87]. This concept is extensively discussed in Kubernetes, a container orchestrator. Hard multi-tenancy can be explained as the outright isolation of one tenant from the other, and the method considers any tenant in a given cluster as a potential malicious tenant.
Soft multi-tenancy	This is another concept popularly known in the Kubernetes multi-tenant consideration, where users are not considered actively malicious since they are within the same organisation. However, as soon as they leave the organisation, they become a potential risk that brings security threats to the cluster members. It is worthy of note that soft multi-tenancy mainly focus on the preventive mechanism and not the defensive [87].
Stateless isolation	Isolation is one of the basic requirements to be met when customising multi-tenant SaaS (S32) in cloud-native architecture. This concept became essential due to the stateless protocol in cloud computing, in which a client request is dealt with by the server with the previous configuration and provides automatic scalability, which reduces resource usage. This creates the need for a specific type of isolation that will make commands sent from a given tenant to the cloud computing server without affecting another tenant or instance.
Lodger	Cloud-native architecture has stretched the multi-tenancy of cloud computing to a depth that brings into reality a different kind of tenant, called the 'lodger'. Cloud-native architecture has become the latest upgrade that cloud computing service providers and cloud computing users consider due to its ability to improve productivity through scalability and automation. In an IaaS model of cloud computing, the service provider can host an instant either in containers or virtual machines resold to the reseller (partner) customers, which involves a bilateral, multi-round negotiation [88]. Furthermore, the customer (tenant of tenant) of the reseller can be addressed as a 'lodger' if considered from the perspective of the real estate industry from which the term 'multi-tenant' was initially borrowed. Lodgers pose a security threat in architecture.
Migration	Migration into the cloud-native architecture has had limited research [9]. (S38) described the migration of existing legacy software and associated customers with perpetual licences and the adoption of cloud-native Software by new customers with no existing economic relationship with the service provider. Recent research is working on this area as multi-tenancy affects how instances can be moved in a data centre (S9), (S7) and (S45).

5.3.1 Isolation

Isolation is a significant dimension in cloud security issues, requiring a vertical solution from the SaaS layer down to physical infrastructure to develop physical-like boundaries among tenants, instead of the virtual limitations currently applied (S12). According to (S12) incomplete implementation of isolation in containers is another major challenge in container-based virtualisation. The more inadequate isolation makes it more susceptible to access into the bare-metal host system from the containers than from the hypervisor in a virtual machine (S43), (S32). Kubernetes, one of the orchestration software packages, cannot guarantee secured isolation between tenants; it offers features that may be sufficient for specific use cases with multi-tenancy in mind. Nevertheless, isolation is achievable in container technology through Namespace as the implementor and Cgroups as the Control (S19), (S23). It needs a more complex configuration of policies such as the pod security policy and network policy, improved scheduling policy/algorithms (S19), and the use of namespaces to provide a clear boundary between nodes.

5.3.2 Malicious User

Multi-tenancy security is complex because both the malicious user and the actual user are on the same server. This is why avoidance of security is possible in multi-tenancy, as it is not designed to infiltrate the inside part of servers, and is ability is limited to the boundaries of the network layer [13]. In agreement, (S10) described the imminent threat around the container-based cloud deployments in a multi-tenant environment as a threat posed by an instance where a malicious or attacker container and non-malicious container reside in the same host operating system and are running on the same container engine. Cloud security deployment could be based on how strongly the container engine can implement identity management.

5.3.3 Migration

In cloud computing, migration can be described as moving from a legacy system to cloud computing or moving from one deployment model to another in a CNA. It can also be considered as the movement of part of the data in the cloud computing paradigm, called portability, through the moving of pods, clusters, containers, and users. Based on isolation and multi-tenancy, migration becomes a challenge. Moving different parts of the prominent structure takes time and resources (S9). However, migration should be a first-class notion in the system, having the same stature as scalability, consistency, fault-tolerance, and functionality. Nonetheless, migration will not be easily achieved if multi-tenancy is not solved in the CNA implementation's architecture stage. Those challenges are being solved by modification and customisation. Organisations using CNA provide their own do-it-yourself (DIY) approaches, such as the Alibaba virtual cluster, which implements multi-tenancy in a containerised orchestration application, Kubernetes. Alibaba virtual cluster achieves this by implementing complex customisation and the application's adjustments, including interface, API, and design parameters. According to the [18], Kubernetes is not multi-tenancy-enabled at default and to achieve a multi-tenancy containerised cloud architecture, multiple configurations and customisation are needed, and so is the use of tools such as namespaces, network policies, resource quotas and isolation systems, such as the sandbox and sole tenant nodes provided and being improved.

6 Conclusion and Future Research

Multi-tenancy is a key characteristic of software-oriented architecture, but according to the review it is not thoroughly considered in the cloud-native architecture which seems that the multi-tenancy in cloud-native seem to be different to concept in software-oriented architecture because of the differences in style, component, and principle of cloud native architecture. This review may not be able to justify the differences based on the scope of this research, further research work in this area is needed. Our discussion shows some the trending concepts in multi-tenancy within cloud-native architecture that brings about improvement to the cloud computing paradigm in general and also challenges that are introduce. We identify challenges such as migration, malicious user and isolation issues which has some interest but not maturity of research area is quite low. Overall, based on analysis and classification, further research is needed to provide a detailed understanding of multi-tenancy in cloud-native architecture through experiments and case studies. The experiment-based findings will make cloud computing security, isolation and migration in cloud-native architecture less complicated. Secondly, future research should investigate other cloud computing properties such as interoperability and portability in cloud-native architecture. Which will remove the gap in knowledge about the adoption of cloud-native architecture and technology and increase the adoption of cloud computing and cloud-native-based technology in small, medium and large-scale enterprises.

References:

[1] G. Ramachandra, M. Iftikhar, and F. A. Khan, ‘A Comprehensive Survey on Security in Cloud Computing’, *Procedia Comput. Sci.*, vol. 110, pp. 465–472, 2017, doi: 10.1016/j.procs.2017.06.124.

[2] P. Mell and T. Grance, ‘The NIST Definition of Cloud Computing’, p. 7.

[3] E. Truyen, D. Van Landuyt, V. Reniers, A. Rafique, B. Lagaisse, and W. Joosen, ‘Towards a container-based architecture for multi-tenant SaaS applications’, in *Proceedings of the 15th International Workshop on Adaptive and Reflective Middleware - ARM 2016*, Trento, Italy, 2016, pp. 1–6. doi: 10.1145/3008167.3008173.

[4] J. Fiaidhi, I. Bojanova, J. Zhang, and L.-J. Zhang, ‘Enforcing Multitenancy for Cloud Computing Environments’, *IT Prof.*, vol. 14, no. 1, pp. 16–18, Jan. 2012, doi: 10.1109/MITP.2012.6.

[5] O. M. Okonor, M. Adda, and A. Gegov, ‘Intelligent Agent-based Technique For Virtual Machine Resource Allocation For Energy-Efficient Cloud Data Centres’, *WSEAS Trans. Commun.*, vol. 19, pp. 37–46, Apr. 2020, doi: 10.37394/23204.2020.19.5.

[6] C. Pahl, ‘Containerization and the PaaS Cloud’, *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 24–31, May 2015, doi: 10.1109/MCC.2015.51.

[7] J. Bhimani *et al.*, ‘Understanding performance of I/O intensive containerized applications for NVMe SSDs’, in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, Las Vegas, NV, USA, Dec. 2016, pp. 1–8. doi: 10.1109/PCCC.2016.7820650.

[8] V. Andrikopoulos, T. Binz, F. Leymann, and S. Strauch, ‘How to adapt applications for the Cloud environment: Challenges and solutions in migrating applications to the Cloud’, *Computing*, vol. 95, no. 6, pp. 493–535, Jun. 2013, doi: 10.1007/s00607-012-0248-2.

[9] N. Kratzke and P.-C. Quint, ‘Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study’, *J. Syst. Softw.*, vol. 126, pp. 1–16, Apr. 2017, doi: 10.1016/j.jss.2017.01.001.

[10] C. Zheng, Q. Zhuang, and F. Guo, ‘A Multi-Tenant Framework for Cloud Container Services’, in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, DC, USA, Jul. 2021, pp. 359–369. doi: 10.1109/ICDCS51616.2021.00042.

[11] Google cloud, ‘Cluster multi-tenancy’, 2018. https://cloud.google.com/kubernetes-engine/docs/concepts/multitenancy-overview

[12] M. Almorsy, J. Grundy, and I. Müller, ‘An Analysis of the Cloud Computing Security Problem’, *ArXiv160901107 Cs*, Sep. 2016, Accessed: Jan. 24, 2022. [Online]. Available: http://arxiv.org/abs/1609.01107

[13] H. Dey, R. Islam, and H. Arif, ‘An Integrated Model To Make Cloud Authentication And Multi-Tenancy More Secure’, in *2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, Jan. 2019, pp. 502–506. doi: 10.1109/ICREST.2019.8644077.

[14] F. Masmoudi, M. Sellami, M. Loulou, and A. H. Kacem, ‘From Event to Evidence: An Approach for Multi-tenant Cloud Services’ Accountability’, in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, Taipei, Taiwan, Mar. 2017, pp. 1082–1089. doi: 10.1109/AINA.2017.48.

[15] Research Student, University of Wolverhampton, UK., O. Matthew, K. Buckley, and M. Garvey, ‘A Framework for Multi-Tenant Database Adoption based on the Influencing Factors’, *Int. J. Inf. Technol. Comput. Sci.*, vol. 8, no. 3, pp. 1–9, Mar. 2016, doi: 10.5815/ijitcs.2016.03.01.

[16] B. Medeiros, M. A. Simplicio, and E. R. Andrade, ‘Designing and Assessing Multi-tenant Isolation Strategies for Cloud Networks’, in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Paris, France, Feb. 2019, pp. 214–221. doi: 10.1109/ICIN.2019.8685898.

[17] X. Gao, Z. Gu, M. Kayaalp, D. Pendarakis, and H. Wang, ‘ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds’, in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Denver, CO, USA, Jun. 2017, pp. 237–248. doi: 10.1109/DSN.2017.49.

[18] CNCF, ‘Cloud-native Definition v1.0’, 2019. https://github.com/cncf/toc/blob/master/DEFINITION.md (accessed Dec. 11, 2021).

[19] G. Toffetti, S. Brunner, M. Blöchlinger, J. Spillner, and T. M. Bohnert, ‘Self-managing cloud-native applications: Design, implementation, and experience’, *Future Gener. Comput. Syst.*, vol. 72, pp. 165–179, Jul. 2017, doi: 10.1016/j.future.2016.09.002.

[20] J. Rahman and P. Lama, ‘Predicting the End-to-End Tail Latency of Containerized Microservices in the Cloud’, in *2019 IEEE International Conference on Cloud Engineering (IC2E)*, Prague, Czech Republic, Jun. 2019, pp. 200–210. doi: 10.1109/IC2E.2019.00034.

[21] K. Petersen, S. Vakkalanka, and L. Kuzniarz, ‘Guidelines for conducting systematic mapping studies in software engineering: An update’, *Inf. Softw. Technol.*, vol. 64, pp. 1–18, Aug. 2015, doi: 10.1016/j.infsof.2015.03.007.

[22] A. A. Adewojo and J. M. Bass, ‘Evaluating the Effect of Multi-Tenancy Patterns in Containerized Cloud-Hosted Content Management System’, in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, Cambridge, Mar. 2018, pp. 278–282. doi: 10.1109/PDP2018.2018.00047.

[23] S. Arnautov *et al.*, ‘SCONE: Secure Linux Containers with Intel SGX’, p. 17.

[24] M. Bélair, S. Laniepce, and J.-M. Menaud, ‘Leveraging Kernel Security Mechanisms to Improve Container Security: a Survey’, in *Proceedings of the 14th International Conference on Availability, Reliability and Security - ARES ’19*, Canterbury, CA, United Kingdom, 2019, pp. 1–6. doi: 10.1145/3339252.3340502.

[25] A. Beltre, P. Saha, and M. Govindaraju, ‘KubeSphere: An Approach to Multi-Tenant Fair Scheduling for Kubernetes Clusters’, in *2019 IEEE Cloud Summit*, Washington, DC, USA, Aug. 2019, pp. 14–20. doi: 10.1109/CloudSummit47114.2019.00009.

[26] M. Beranek, V. Kovar, and G. Feuerlicht, ‘Framework for Management of Multi-tenant Cloud Environments’, in *Cloud Computing – CLOUD 2018*, vol. 10967, M. Luo and L.-J. Zhang, Eds. Cham: Springer International Publishing, 2018, pp. 309–322. doi: 10.1007/978-3-319-94295-7_21.

[27] E. F. Boza, C. L. Abad, S. P. Narayanan, B. Balasubramanian, and M. Jang, ‘A Case for Performance-Aware Deployment of Containers’, in

Proceedings of the 5th International Workshop on Container Technologies and Container Clouds - WOC '19, Davis, CA, USA, 2019, pp. 25–30. doi: 10.1145/3366615.3368355.

[28] C. Chen, J. Cai, N. Ren, and X. Cheng, 'Design and Implementation of Multi-tenant Vehicle Monitoring Architecture Based on Microservices and Spark Streaming', in *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, Kuala Lumpur, Malaysia, Jul. 2020, pp. 169–172. doi: 10.1109/CISCE50729.2020.00040.

[29] W. Chen, X. Zhou, and J. Rao, 'Preemptive and Low Latency Datacenter Scheduling via Lightweight Containers', *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 12, pp. 2749–2762, Dec. 2020, doi: 10.1109/TPDS.2019.2957754.

[30] G. Collins and Y. Biran, 'Multi-tenant utility computing with compute containers', in *2015 IEEE 5th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, Berlin, Germany, Sep. 2015, pp. 213–217. doi: 10.1109/ICCE-Berlin.2015.7391238.

[31] J. Flora and N. Antunes, 'Studying the Applicability of Intrusion Detection to Multi-Tenant Container Environments', in *2019 15th European Dependable Computing Conference (EDCC)*, Naples, Italy, Sep. 2019, pp. 133–136. doi: 10.1109/EDCC.2019.00033.

[32] J. Flora, P. Goncalves, and N. Antunes, 'Using Attack Injection to Evaluate Intrusion Detection Effectiveness in Container-based Systems', in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*, Perth, WA, Australia, Dec. 2020, pp. 60–69. doi: 10.1109/PRDC50213.2020.00017.

[33] X. Gao, B. Steenkamer, Z. Gu, M. Kayaalp, D. Pendarakis, and H. Wang, 'A Study on the Security Implications of Information Leakages in Container Clouds', *IEEE Trans. Dependable Secure Comput.*, pp. 1–1, 2018, doi: 10.1109/TDSC.2018.2879605.

[34] X. Gao, Z. Gu, M. Kayaalp, D. Pendarakis, and H. Wang, 'ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds', in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Denver, CO, USA, Jun. 2017, pp. 237–248. doi: 10.1109/DSN.2017.49.

[35] X. Gao, Z. Gu, Z. Li, H. Jamjoom, and C. Wang, 'Houdini's Escape: Breaking the Resource Rein of Linux Control Groups', in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, London United Kingdom, Nov. 2019, pp. 1073–1086. doi: 10.1145/3319535.3354227.

[36] M. Garriga, 'Towards a Taxonomy of Microservices Architectures', in *Software Engineering and Formal Methods*, vol. 10729, A. Cerone and M. Roveri, Eds. Cham: Springer International Publishing, 2018, pp. 203–218. doi: 10.1007/978-3-319-74781-1_15.

[37] D. Godlove, 'Singularity: Simple, secure containers for compute-driven workloads', in *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*, Chicago IL USA, Jul. 2019, pp. 1–4. doi: 10.1145/3332186.3332192.

[38] R. Gracia-Tinedo, J. Sampé, G. París, M. Sánchez-Artigas, P. García-López, and Y. Moatti, 'Software-defined object storage in multi-tenant environments', *Future Gener. Comput. Syst.*, vol. 99, pp. 54–72, Oct. 2019, doi: 10.1016/j.future.2019.03.020.

[39] A. Hegde, R. Ghosh, T. Mukherjee, and V. Sharma, 'SCoPe: A Decision System for Large Scale Container Provisioning Management', in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, Jun. 2016, pp. 220–227. doi: 10.1109/CLOUD.2016.0038.

[40] S. Huaxin, X. Gu, K. Ping, and H. Hongyu, 'An Improved Kubernetes Scheduling Algorithm for Deep Learning Platform', in *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, Dec. 2020, pp. 113–116. doi: 10.1109/ICCWAMTIP51612.2020.9317317.

[41] K. Jeong, R. Figueiredo, and K. Ichikawa, 'PARES: Packet Rewriting on SDN-Enabled Edge Switches for Network Virtualization in Multi-Tenant Cloud Data Centers', in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, Honolulu, CA, USA, Jun. 2017, pp. 9–17. doi: 10.1109/CLOUD.2017.11.

[42] C. Jiang *et al.*, 'Energy efficiency comparison of hypervisors', *Sustain. Comput. Inform. Syst.*, vol. 22, pp. 311–321, 2019, doi: https://doi.org/10.1016/j.suscom.2017.09.005.

[43] M. Kaminski, E. Truyen, E. H. Beni, B. Lagaisse, and W. Joosen, 'A framework for black-box SLO tuning of multi-tenant applications in Kubernetes', in *Proceedings of the 5th International Workshop on Container Technologies and Container Clouds - WOC '19*, Davis, CA, USA, 2019, pp. 7–12. doi: 10.1145/3366615.3368352.

[44] B. Kelley, J. J. Prevost, P. Rad, and A. Fatima, 'Securing Cloud Containers Using Quantum Networking Channels', in *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, New York, NY, USA, Nov. 2016, pp. 103–111. doi: 10.1109/SmartCloud.2016.58.

[45] F. Leymann, U. Breitenbücher, S. Wagner, and J. Wettinger, 'Native Cloud Applications: Why Monolithic Virtualization Is Not Their Foundation', in *Cloud Computing and Services Science*, vol. 740, M. Helfert, D. Ferguson, V. Méndez Muñoz, and J. Cardoso, Eds. Cham: Springer International

Publishing, 2017, pp. 16–40. doi: 10.1007/978-3-319-62594-2_2.

[46] H. Song, P. H. Nguyen, F. Chauvel, J. Glattetre, and T. Schjerpen, 'Customizing Multi-Tenant SaaS by Microservices: A Reference Architecture', in *2019 IEEE International Conference on Web Services (ICWS)*, Jul. 2019, pp. 446–448. doi: 10.1109/ICWS.2019.00081.

[47] M. Makki, D. V. Landuyt, B. Lagaisse, and W. Joosen, 'Thread-level resource consumption control of tenant custom code in a shared JVM for multi-tenant SaaS', *Future Gener. Comput. Syst.*, vol. 115, pp. 351–364, 2021, doi: https://doi.org/10.1016/j.future.2020.09.025.

[48] Ł. Makowski and P. Grosso, 'Evaluation of virtualization and traffic filtering methods for container networks', *Future Gener. Comput. Syst.*, vol. 93, pp. 345–357, Apr. 2019, doi: 10.1016/j.future.2018.08.012.

[49] A. M. Maliszewski, D. Griebler, C. Schepke, A. Ditter, D. Fey, and L. G. Fernandes, 'The NAS Benchmark Kernels for Single and Multi-Tenant Cloud Instances with LXC/KVM', in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, Orleans, Jul. 2018, pp. 359–366. doi: 10.1109/HPCS.2018.00066.

[50] R. Morabito, R. Petrolo, V. Loscrì, and N. Mitton, 'Reprint of : LEGIoT: A Lightweight Edge Gateway for the Internet of Things', *Future Gener. Comput. Syst.*, vol. 92, pp. 1157–1171, Mar. 2019, doi: 10.1016/j.future.2018.10.020.

[51] P. H. Nguyen, H. Song, F. Chauvel, R. Muller, S. Boyar, and E. Levin, 'Using microservices for non-intrusive customization of multi-tenant SaaS', in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Tallinn Estonia, Aug. 2019, pp. 905–915. doi: 10.1145/3338906.3340452.

[52] G. Nikol, M. Trager, S. Harrer, and G. Wirtz, 'Service-Oriented Multi-tenancy (SO-MT): Enabling Multi-tenancy for Existing Service Composition Engines with Docker', in *2016 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, Oxford, United Kingdom, Mar. 2016, pp. 238–243. doi: 10.1109/SOSE.2016.40.

[53] S. Pan, L. Zhu, and J. Qiao, 'An Open Sharing Pattern Design of Massive Power Big Data', in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, Chengdu, China, Apr. 2019, pp. 5–9. doi: 10.1109/ICCCBDA.2019.8725750.

[54] R. Peinl, F. Holzschuher, and F. Pfitzer, 'Docker Cluster Management for the Cloud - Survey Results and Own Solution', *J. Grid Comput.*, vol. 14, no. 2, pp. 265–282, Jun. 2016, doi: 10.1007/s10723-016-9366-y.

[55] A. Pi, W. Chen, X. Zhou, and M. Ji, 'Profiling distributed systems in lightweight virtualized environments with logs and resource metrics', in *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing - HPDC '18*, Tempe, Arizona, 2018, pp. 168–179. doi: 10.1145/3208040.3208044.

[56] F. Ramos, E. Viegas, A. Santin, P. Horchulhack, R. R. dos Santos, and A. Espindola, 'A Machine Learning Model for Detection of Docker-based APP Overbooking on Kubernetes', in *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, Jun. 2021, pp. 1–6. doi: 10.1109/ICC42927.2021.9500259.

[57] A. Ranjbar, M. Komu, P. Salmela, and T. Aura, 'SynAPTIC: Secure and Persistent Connectivity for Containers', in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, Spain, May 2017, pp. 262–267. doi: 10.1109/CCGRID.2017.62.

[58] M. A. Rodriguez and R. Buyya, 'Scheduling dynamic workloads in multi-tenant scientific workflow as a service platforms', *Future Gener. Comput. Syst.*, vol. 79, pp. 739–750, 2018, doi: https://doi.org/10.1016/j.future.2017.05.009.

[59] P. Rosati, F. Fowley, C. Pahl, D. Taibi, and T. Lynn, 'Right Scaling for Right Pricing: A Case Study on Total Cost of Ownership Measurement for Cloud Migration', in *Cloud Computing and Services Science*, vol. 1073, V. M. Muñoz, D. Ferguson, M. Helfert, and C. Pahl, Eds. Cham: Springer International Publishing, 2019, pp. 190–214. doi: 10.1007/978-3-030-29193-8_10.

[60] P. Sharma, L. Chaufournier, P. Shenoy, and Y. C. Tay, 'Containers and Virtual Machines at Scale: A Comparative Study', in *Proceedings of the 17th International Middleware Conference*, Trento Italy, Nov. 2016, pp. 1–13. doi: 10.1145/2988336.2988337.

[61] A. Slominski, V. Muthusamy, and R. Khalaf, 'Building a Multi-tenant Cloud Service from Legacy Code with Docker Containers', in *2015 IEEE International Conference on Cloud Engineering*, Tempe, AZ, USA, Mar. 2015, pp. 394–396. doi: 10.1109/IC2E.2015.66.

[62] J. Spillner, 'Self-balancing architectures based on liquid functions across computing continuums', in *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing Companion*, Leicester United Kingdom, Dec. 2021, pp. 1–6. doi: 10.1145/3492323.3495589.

[63] K. A. Torkura and C. Meinel, 'Towards Vulnerability Assessment as a Service in OpenStack Clouds', in *2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops)*, Dubai, Nov. 2016, pp. 1–8. doi: 10.1109/LCN.2016.022.

[64] J. Trnkoczy, U. Pascinski, S. Gec, and V. Stankovski, 'SWITCH-ing from Multi-Tenant to Event-Driven Videoconferencing Services', in *2017*

IEEE 2nd International Workshops on Foundations and Applications of Self Systems (FAS*W)*, Tucson, AZ, USA, Sep. 2017, pp. 219–226. doi: 10.1109/FAS-W.2017.151.

[65] E. Truyen, D. Van Landuyt, B. Lagaisse, and W. Joosen, ‘Performance overhead of container orchestration frameworks for management of multi-tenant database deployments’, in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, Limassol Cyprus, Apr. 2019, pp. 156–159. doi: 10.1145/3297280.3297536.

[66] S. Ugwuanyi, R. Asif, and J. Irvine, ‘Network Virtualization: Proof of Concept for Remote Management of Multi-Tenant Infrastructure’, in *2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Application (DependSys)*, Nadi, Fiji, Dec. 2020, pp. 98–105. doi: 10.1109/DependSys51298.2020.00023.

[67] J. Xiong and H. Chen, ‘Challenges for building a cloud native scalable and trustable multi-tenant AIoT platform’, in *Proceedings of the 39th International Conference on Computer-Aided Design*, Virtual Event USA, Nov. 2020, pp. 1–8. doi: 10.1145/3400302.3415756.

[68] L. De Simone, M. D. Mauro, R. Natella, and F. Postiglione, ‘A Latency-Driven Availability Assessment for Multi-Tenant Service Chains’, *IEEE Trans. Serv. Comput.*, pp. 1–14, 2022, doi: 10.1109/TSC.2022.3183938.

[69] Q. Deng, X. Tan, J. Yang, C. Zheng, L. Wang, and Z. Xu, ‘A Secure Container Placement Strategy Using Deep Reinforcement Learning in Cloud’, in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Hangzhou, China, May 2022, pp. 1299–1304. doi: 10.1109/CSCWD54268.2022.9776226.

[70] J. A. Samo, Z. Ahmed, and A. Shaikh, ‘Advocating isolation of resources among multi-tenants by containerization in IaaS cloud model’, in *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, Karachi, Pakistan, Apr. 2017, pp. 1–17. doi: 10.1109/ICIEECT.2017.7916567.

[71] S. Livi, Q. Jacquemart, D. L. Pacheco, and G. Urvoy-Keller, ‘Container-Based Service Chaining: A Performance Perspective’, in *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*, Pisa, Italy, Oct. 2016, pp. 176–181. doi: 10.1109/CloudNet.2016.51.

[72] O. Katz, D. Rawitz, and D. Raz, ‘Containers Resource Allocation in Dynamic Cloud Environments’, in *2021 IFIP Networking Conference (IFIP Networking)*, Espoo and Helsinki, Finland, Jun. 2021, pp. 1–9. doi: 10.23919/IFIPNetworking52078.2021.9472812

[73] D. Huang, J. Wang, Q. Liu, N. Xiao, H. Wu, and J. Yin, ‘Enhancing Proportional IO Sharing on Containerized Big Data File Systems’, *IEEE Trans. Comput.*, pp. 1–1, 2021, doi: 10.1109/TC.2020.3037078.

[74] A. Beltre, P. Saha, and M. Govindaraju, ‘Framework for Analysing a Policy-driven Multi-Tenant Kubernetes Environment’, in *2021 IEEE Cloud Summit (Cloud Summit)*, Hempstead, NY, USA, Oct. 2021, pp. 49–56. doi: 10.1109/IEEECloudSummit52029.2021.00016.

[75] J. Flora, ‘Improving the Security of Microservice Systems by Detecting and Tolerating Intrusions’, in *2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, Coimbra, Portugal, Oct. 2020, pp. 131–134. doi: 10.1109/ISSREW51248.2020.00051.

[76] S. G. Haugeland, P. H. Nguyen, H. Song, and F. Chauvel, ‘Migrating Monoliths to Microservices-based Customizable Multi-tenant Cloud-native Apps’, in *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Palermo, Italy, Sep. 2021, pp. 170–177. doi: 10.1109/SEAA53835.2021.00030.

[77] E. Truyen, A. Jacobs, S. Verreydt, E. H. Beni, B. Lagaisse, and W. Joosen, ‘Feasibility of Container Orchestration for Adaptive Performance Isolation in Multi-Tenant SaaS Applications’, in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2020, pp. 162–169. doi: 10.1145/3341105.3374034.

[78] H. Song, F. Chauvel, and A. Solberg, ‘Deep Customization of Multi-Tenant SaaS Using Intrusive Microservices’, in *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results*, New York, NY, USA, 2018, pp. 97–100. doi: 10.1145/3183399.3183407.

[79] B. C. Şenel, M. Mouchet, J. Cappos, O. Fourmaux, T. Friedman, and R. McGeer, ‘EdgeNet: A Multi-Tenant and Multi-Provider Edge Cloud’, in *Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking*, New York, NY, USA, 2021, pp. 49–54. doi: 10.1145/3434770.3459737.

[80] W. Cao *et al.*, ‘LogStore: A Cloud-Native and Multi-Tenant Log Database’, in *Proceedings of the 2021 International Conference on Management of Data*, Virtual Event China, Jun. 2021, pp. 2464–2476. doi: 10.1145/3448016.3457565.

[81] C.-H. Lee, Z. Li, X. Lu, T. Chen, S. Yang, and C. Wu, ‘Multi-Tenant Machine Learning Platform Based on Kubernetes’, in *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, Tianjin China, Apr. 2020, pp. 5–12. doi: 10.1145/3404555.3404565.

[82] M. Cavalcanti, P. Inacio, and M. Freire, ‘Performance Evaluation of Container-Level Anomaly-Based Intrusion Detection Systems for Multi-Tenant Applications Using Machine Learning Algorithms’, New York, NY, USA, 2021. doi: 10.1145/3465481.3470066.

[83] A. Zafeiropoulos, E. Fotopoulou, N. Filinis, and S. Papavassiliou, 'Reinforcement learning-assisted autoscaling mechanisms for serverless computing platforms', *Simul. Model. Pract. Theory*, vol. 116, p. 102461, Apr. 2022, doi: 10.1016/j.simpat.2021.102461.

[84] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, 'Requirements engineering paper classification and evaluation criteria: a proposal and a discussion', *Requir. Eng.*, vol. 11, no. 1, pp. 102–107, Mar. 2006, doi: 10.1007/s00766-005-0021-6.

[85] P. Bourque, R. E. Fairley, and IEEE Computer Society, *Guide to the software engineering body of knowledge*. 2014.

[86] C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, 'Cloud Container Technologies: A State-of-the-Art Review', *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 677–692, Jul. 2019, doi: 10.1109/TCC.2017.2702586.

[87] J. Frazelle, 'Multi-Tenancy Design Space. Retrieved from', 2018. https://docs.google.com/document/d/1PjlsBmZw6Jb3XZeVyZ0781m6PV7-nSUvQrwObkvz7jg/edit# (accessed Nov. 22, 2021).

[88] B. Pittl, W. Mach, and E. Schikuta, 'Cloud Resellers on Bazaar-Based Cloud Markets', in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, Jul. 2018, pp. 564–571. doi: 10.1109/CLOUD.2018.00078.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.