Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

TC 11 Briefing Papers

# Security in microservice-based systems: A Multivocal literature review

*Anelis Pereira-Vale*[a,*], *Eduardo B. Fernandez*[b], *Raúl Monge*[a], *Hernán Astudillo*[a], *Gastón Márquez*[a]

[a] *Universidad Técnica Federico Santa María, Valparaíso, Chile*
[b] *Florida Atlantic University, Boca Raton, Florida, USA*

ARTICLE INFO

*Article history:*
Received 8 August 2020
Revised 10 November 2020
Accepted 11 January 2021
Available online 16 January 2021

*Keywords:*
Secure software development
Microservice-based systems
Security mechanisms
Security solutions
Multivocal literature review

ABSTRACT

Microservices define an architectural style that conceives systems as a suite of modular, independent and scalable services. While application design is now simpler, designing secure applications is in general harder than for monolithic applications and the current literature offers little orientation to architects and developers regarding solutions. This article describes the design and results of a multivocal literature review of the security solutions that have been proposed for microservice-based systems. The study yielded 370 academic articles and 620 grey literature; duplicates removal and the application of exclusion criteria left 36 from the academic literature and 34 from the grey literature. The security solution(s) proposed in each article were classified into variations of standard security mechanisms (e.g., Access Control) and scopes (Info Management, Threat Modeling, etc), and were associated to security contexts (detect, mitigate/stop, react, recover from attack). Our research questions addressed frequency of publications, research methodologies, security mechanisms, and security contexts. Key findings were that (1) both kinds of literature differ in their preferred empirical research strategies (examples, experiments and case studies); (2) The solutions proposed in the 70 selected articles correspond to 15 classifications of security mechanisms and analyses; (3) the most mentioned security mechanisms are Authentication and Authorization; (4) around 2/3 of solutions focused on Mitigate/Stop attacks, but none on reacting and recovering from them, and (5) the methodologies used are mostly block diagrams and code, with little use of models or analysis. These findings hold for both grey and academic literature. This study is a first step towards providing secure software researchers and practitioners a comprehensive catalog of security solutions and mechanisms, and where the clear identification of the most used security solutions will simplify their reuse to address security problems while designing microservice-based systems.



* Corresponding author.
*E-mail addresses:* anelis.pereira@sansano.usm.cl (A. Pereira-Vale), fernande@fau.edu (E.B. Fernandez), rmonge@inf.utfsm.cl (R. Monge), hernan@inf.utfsm.cl (H. Astudillo), gaston.marquez@usm.cl (G. Márquez).
https://doi.org/10.1016/j.cose.2021.102200
0167-4048/

## 1. Introduction

Microservice-based systems are an emerging variety of service-oriented architectures, composed of several small independent services Sill (2016), Dong et al. (2015). Each of these services is executed in its own process and communicates with the others through lightweight protocols Yu et al., 2018. These services are implemented independently, often through fully automated implementation tools Pathania (2017).

Although this emerging architectural style brings several advantages to develop complex systems, it also brings along new challenges, of which the main one is security. For every 100 KLOC, a monolithic application will have an average of 39 vulnerabilities, while a microservice application will have an average of 180 vulnerabilities [1]. The industrial report by Hinkley et al. (2018) show that in recent years some companies have suffered massive attacks on their microservice-based systems, ranging from specific services to the whole system. For this reason, companies are constantly creating variations and combinations of standard security mechanisms with the aim of protecting themselves from such attacks. Surveys have shown that security is an important concern of microservice-based applications. At the same time, other surveys, e.g. Dragoni et al. (2017), have highlighted a general lack of research in the area of microservice security.

In previous work Pereira-Vale et al., 2019, the authors of this article reported a *Systematic Literature Mapping* (SLM) of security mechanisms for microservice-based systems, and pointed out the vast unexplored knowledge and experience existing the practitioners' professional literature. Although there are some previous reviews of security-related literature for microservice-based systems Yu et al., 2018, Hannousse and Yahiouche, 2020, we did not find reviews that included grey literature. To cover this gap, this article describes a *Multivocal Literature Review* (MLR) of security solutions in microservice-based systems, surveying primary studies in academic and grey literature. According to Garousi et al. (2019) the most widely used and accepted definition of grey literature is literature produced on all levels of government, academics, business and industry, in print and electronic formats, but which is not controlled by commercial publishers, i.e., where publishing is not the primary activity of the producing body. *Multivocal Literature Reviews* (MLRs) are a type of *Systematic Literature Reviews* (SLRs) that combine data from several kinds of sources, e.g. scientific literature (i.e., academic "papers") and professional "grey" literature (i.e., blog posts, white papers, videos, presentations, and similar).

This study found 370 articles in the academic literature and selected 36 primary studies. In the grey literature 34 articles were selected from a total of 620. Its main contributions are (i) reporting the state of the art on security solutions and analyses for microservice-based systems, (ii) identifying the most used security solutions for these systems, and (iii) enumerating promising directions for research. Our audience includes researchers and practitioners, from both security and software engineering areas. We have included papers that consider microservices security including the effect of their implementation using containers but not papers that discuss only container security.

The remainder of this article is structured as follows: Section 2 provides an overview of security problems in microservices; Section 3 discusses related work; Section 4 describes the study design; Section 5 presents the results of the study; Section 6 indicates directions for research; Section 7 indicates use cases for our multivocal literature review; Section 8 addresses threats to validity; and Section 9 presents conclusions and future work.

## 2. Security in microservices architectures

A microservice architecture is a special case of the service-oriented architecture (SOA); it decomposes applications as collections of loosely-coupled services. More specifically: "Microservices architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies" Lewis and Fowler, 2020.

Typically, a microservice is responsible for a single task (implements one single business function), it is self-contained (it contains presentation, business and data layers making it a complete computational unit), light-weighted (typically a few hundred lines of code), autonomous (can be deployed individually), and loosely coupled (communicates over point-to-point networks). Microservices are typically deployed inside containers, which is another reason for their popularity. This structure provides for early release, localized failures, and bug fixing independence. It also enables service-specific scaling so that popular services can be scaled on demand. Microservices are built around capabilities and are implemented using agile software development techniques. Containers facilitate the work of DevOps teams by allowing the use of code to automatically perform infrastructure management tasks. A desired state is specified by changing configurations and this state is then automatically applied to the environment.

In spite of these advantages, security now becomes a larger problem compared to monolithic applications. The decomposition of functions increases the attack surface and application security becomes harder due to the fact that security is a global property, not the sum of local security defenses. This has brought a large number of discussions of microservices security, the most important of which are considered here and put in perspective.

In this environment, communications between microservices are exposed through the network, which creates a potential attack surface. When a microservice is compromised, it can send malicious requests to other microservices. In addition to using TLS to protect the communications among microservices, microservice infrastructures such as Kubernetes also provide inter-service authorization mechanisms to specify valid accesses. The problem is now how to coordinate dis-

[1] https://www.whitehatsec.com/blog/microservices-security/.

tributed authorization rules and how to update these rules when a microservice or a user comes or leaves. Further, inter-service authorization should follow the conceptual models of applications and apply the principle of least privilege He and Yang (2017b).

Authentication brings similar problems, one of which is where to keep the authentication information; for example, if kept in an authentication server, this server must be updated every time a new microservice or new users are added; if this information is kept by each microservice, each microservice must be updated when there is a new microservice or new users. Decomposing an application into microservices is a complex task and it can affect scalability and security. Some solutions define processes to perform this decomposition respecting security and scalability constraints, e.g Ahmadvand, Ibrahim, 2016.

Microservices architectures may have vulnerabilities because of their use of images from public or private repositories that could be potentially infected. Several defenses against these threats have been proposed, including Moving Target Defenses (MTD). MTDs transform specific system components to create uncertainty for attackers, thus reducing the probability of successful attacks Torkura et al., 2018.

An attacker can exploit not only the application vulnerabilities but also the vulnerabilities and misconfigurations of the microservices platform. A compromised credential, service, network, or host may all lead to a fully compromised cluster of services.

All the cases discussed above refer to IT applications. However, microservices are now being used also in cyber-physical systems (CPSs). CPSs interact intensively with their environments and can be attacked from the cyber and the physical domains. Most CPS include IoT portions. An IoT system can be viewed as a system made up of smart things (devices) that are interconnected together to perform some functions, but some design methodologies view the IoT just as a set of microservices. Any IoT node can be abstracted as a smart object providing certain services over the network, and the focus of developers can be raised to the level of data and services rather than on the devices and end-to-end communication Lu et al. (2017). This allows to applying conceptual security models.

## 3. Related work

A few studies have surveyed work related to security aspects of microservice-based systems.

Yu et al., 2018 surveyed work about security risks for microservices-based applications running on fog platforms, and found that security issues arose in four system aspects: containers (used as deployment and operational environment for microservices); data (communicated among services), permissions (as guarantee of services security); and network security (the foundation for secure communication). They proposed security solutions regarding services communication. These results apply only to fog systems and not to microservice applications running on other platforms. Our survey considers any platform.

Hannousse and Yahiouche, 2020 conducted a systematic mapping study to uncover the main threats to the security of microservice-based systems. They also pointed out security mechanisms used to detect, mitigate and prevent those threats, and considered techniques and tools used to examine and validate the proposed solutions. They examined 47 papers published since 2011, and found that unauthorized access, sensitive data exposure and compromising individual microservices were the most addressed threats in the literature. That paper has some common topics with ours but emphasizes threats and their classification is different from ours. It is not a multivocal survey either.

de Aguiar Monteiro et al. (2018) carried out a survey on security, privacy and standardization on cloud computing environments for microservices architectures. Among their main findings for security they include four fundamental aspects: containers, data, permissions, and network. In addition, they present a set of mechanisms to prevent security threats, such as: mutual transport layer security, host-authenticated TLS with in-band authentication and principal propagation via Security Tokens. The issues addressed by Aguiar et al. do not coincide with the research questions we conducted in our study.

In a previous study we presented a systematic mapping of security mechanisms in microservice-based systems, Pereira-Vale et al. (2019). We analyzed 26 primary studies in the academic literature published between 2015 and early 2019, and found that the most reported security mechanisms were authorization, authentication, and credentials. This paper is a significant expansion of that work in that it includes a wider scope (not just mechanisms) and considers grey literature.

## 4. Research protocol

A MLR Garousi et al. (2019) is a form of a SLR that includes grey literature in addition to the works published in the formal literature. Grey literature includes those publications made, for example, in blog, videos, white papers, audios, company websites; that is, literature that has not undergone a rigorous peer review process, and control of the process and where the publication channels are limited. Formal or academic literature is one that is published, for example in a journal or conference, and has undergone a thorough peer review process. MLRs benefit both researchers and professionals, as they summarize the state of the art of both literatures on a chosen topic. Although MLRs in Software Engineering are quite young Garousi et al. (2019), at least ten have been published since 2013 by Calderón et al. (2018), Garousi et al. (2017), Garousi and Mäntylä (2016) and Myrbakken and Colomo-Palacios (2017).

This study followed the guidelines proposed by Petersen et al. (2008) and Kitchenham and Charters (2007), complemented by the strategies presented by Garousi et al. (2019) for multivocal literature reviews.

The following subsections describe the study design and the outcomes of their execution.

### 4.1. *Goal and research questions*

The goal of this MLR is to *identify and organize security solutions and analyses proposed in the academic and grey literature for microservice-based systems*.

To achieve this goal, we propose several research questions (RQ's). The bulk of the study is the gathering, identifying and classifying of security solutions (RQ1 to RQ3). To organize this classification, we used dimensions borrowed from the security literature:

- *Security solution*: a secure development methodology must include artifacts to express conceptual entities used in the design. A security solution is a conceptual artifact that addresses a threat and describes how to satisfy a security requirement by realizing a policy Uzunov et al. (2015). Security patterns and security reference architectures are security solutions;
- *Security mechanism*: recognizable and recurring standard software structures that often (yet not necessarily) result from the use of systematic design; we used Uzunov et al. (2015) classification: *Authorization, Identity Management/Authentication, Access Control, Secure Communication, Filtering, Storage Security, Logging and Monitoring*, and *Execution Control*;
- *Security scope*: the focus of an analysis study, which can be Information Management, Application, Implementation, Evaluation, Threat Modeling, or General Architecture;
- *Security context*: describes at what stage of an attack the solution applies: before, during, or after; we used the taxonomy of Fernandez et al. (2015) for classifying the solutions reported by the academic and grey literatures. The four contexts are: *detect, mitigate (or stop), react to, and recover from* attacks.

Because this is the first opportunity to compare systematically academic and industrial literature on this topic, we also formulated some RQ's about the literature itself (RQ0 to RQ3). The RQ's of this study are:

- **RQ0**: *How has the **frequency of publications** on security in microservice-based systems varied along time? And how have the selected publication **publishers** changed?*
  Rationale: We want to understand the publications: determine characterization and evolution along years and chosen venues, and whether they differ in academic and grey literature.
- **RQ1**: *What **research methodologies** have been used to study security of microservice-based systems?*
  Rationale: We want to understand the research reported in those publications: what research and validation methodologies are used to conduct them, and whether academic and grey literatures differ. We want to see the formalizations used to develop the research, e.g. UML, ontologies, or formal methods.
- **RQ2**: Security Solutions Classification
  - **RQ2.1**:*What **security mechanisms** have been proposed or studied in microservice-based systems?*
  - **RQ2.2**: *What is the **security scope** of studies in microservices-based systems?*

  Rationale: We want to understand the security solutions reported in those publications: explore (1) if some security mechanisms are mature, (2) the security scope from the solutions, (3) the relative depth of the academic/grey literature on them, and (4) eventual research opportunities in under-studied areas.
- **RQ3**: *What **security contexts** have been addressed by research?*
  Rationale: We want to know (1) if research on some security contexts is already mature, (2) the relative depth of academic vs industrial research on them, and (3) eventual research opportunities in under- studied areas.

### 4.2. *Review process*

Fig. 1 illustrates the main steps for searching and selecting sources in the multivocal review process executed in this study. The following subsections describe its steps.

#### 4.2.1. *Search process*

To find articles in the academic literature, we queried seven major digital databases: **IEEE Xplore**[2], **ACM Digital Library**[3], **Wiley**[4], **Springer Link**[5], **Web of Science**[6], **ScienceDirect**[7], and **Scopus**[8].

To find reports in the grey literature we used generic web search engines. The common grey literature publication types include reports (annual, research, technical, project, etc.), working papers, government documents, white papers, videos and evaluations.

As recommended by Garousi et al. (2019), grey studies can be identified by exploring search strings on search engines.

To explore the digital databases and the web search engines, we built a structured search string guided by the research questions using Petersen et al. (2008) guidelines:

("secure" OR "security") AND ("microservi"* OR "micro servi"* OR "micro-servi"*)

The multivocal study was conducted from November 2018 to December 2019.

#### 4.2.2. *Source selection*

We defined the following *inclusion* and *exclusion criteria* aiming to collect primary studies and reports from grey literature.

- *Inclusion criteria*
  - Studies related to microservice-based systems;
  - Studies whose primary focus is security aspects of microservice-based systems;
  - Studies that provide solutions, methodologies, security mechanisms or other procedures to handle security scope;
  - Studies written in English.
- *Exclusion criteria*
  - Short studies (less than 3 pages for academic literature);
  - Secondary or tertiary studies (such as literature reviews, surveys, and others);

[2] IEEE Xplore: https://ieeexplore.ieee.org/Xplore/home.jsp.
[3] ACM Digital Library: https://dl.acm.org/.
[4] Wiley: https://onlinelibrary.wiley.com/.
[5] Springer Link: https://link.springer.com/.
[6] Web of Science: http://apps.webofknowledge.com.usm.idm.oclc.org/.
[7] ScienceDirect: https://www.sciencedirect.com/.
[8] Scopus: https://www.scopus.com/.

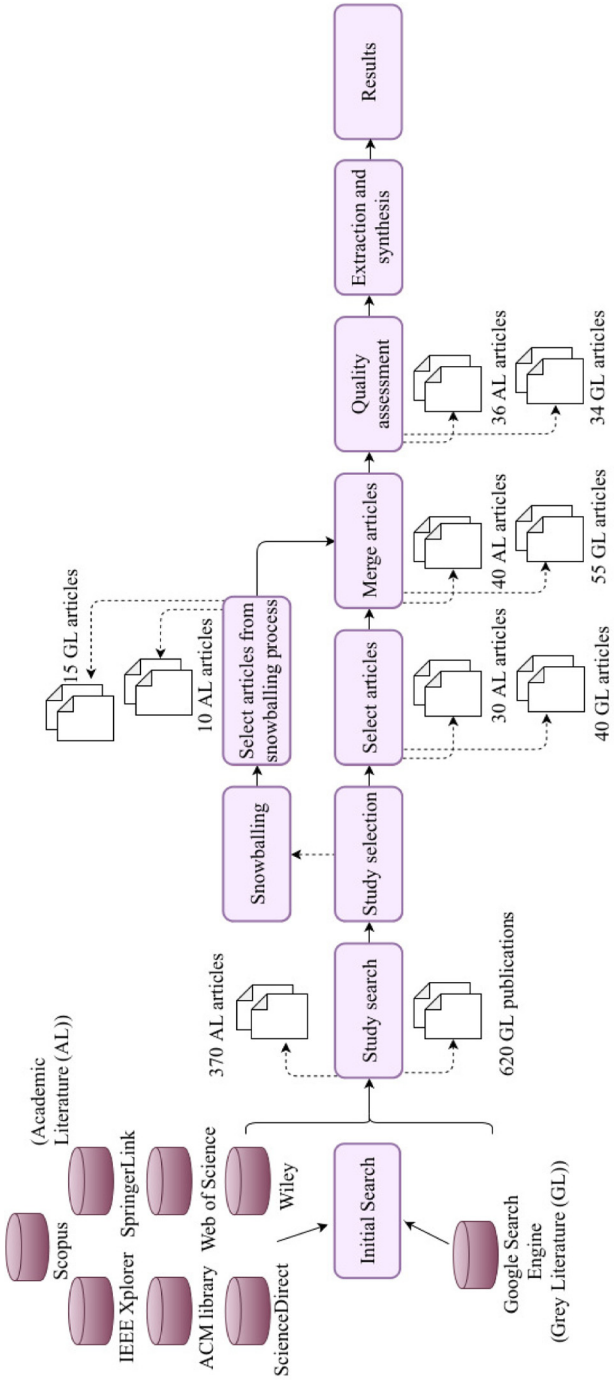



**Fig. 1 – Overview of the MLR process.**

- Studies without full text available;
- Studies structured as tutorial, editorials, and others (only for academic literature);
- Studies which do not offer details on security in microservice-based systems.

#### 4.2.3. *Snowballing process*

We applied "snowballing" Wohlin (2014) in the academic literature to avoid missing relevant studies. This closure-like technique reviews the references of each study in the paper set, adds them to the papers set, and repeats iteratively until there are no further changes.

We performed snowballing both backward and forward. For snowballing backward, we reviewed the references in each selected study. For snowballing forward, we serched for citations of each selected study. This process allowed identifying additional primary studies not found in the initial search (see Fig. 1), 10 in academic literature and 15 in grey literature.

Once the list of articles was expanded, the inclusion/exclusion and criteria were applied, and the final paper set had 36 primary studies in academic literature and 34 publications in grey literature.

#### 4.2.4. *Data extraction and synthesis*

After selecting the primary studies from the academic literature and works from the grey literature, the research team proceeded to extract and encode the necessary data from each of them. This was carried out collaboratively and in common agreement in the terminologies and classifications that we would use for each study. Initially we extracted the metadata, represented, among other fields, by the name, year of publication, venue type for academic literature or source type for grey literature, among other metadata. Table 1 shows the metadata defined for each kind of literature and summarizes the data extraction scheme used for this study.

Once encoding was complete, the team carried out a peer review to validate the results obtained.

#### 4.2.5. *Online repository*

The final set of articles and sources for both academic and grey literature are available online on One Drive spreadsheets [9]. The online repository is divided into two spreadsheets, each of them with the final list of articles for each kind of literature, as well as the encoding carried out to enable demographics and RQ's processing.

## 5. Results

The search was concluded in December 2019. The earliest publications found date back to 2015, so the study covers from 2015 to the December 2019.

To assess the quality of each study, the team defined the quality criteria in Table 2.

Each article was assesed using a 5-level Likert scale Robinson (2014), answering the question "is this quality criterion achieved?:" *strongly agree, agree, neither agree nor disagree,*

[9] http://bit.ly/MLR_MicroserviceSecurity.

**Table 1 – Data fields to extract information of publication (GL = *grey literature*, AL = *academic literature*).**

| RQ | Data field | Description | Source |
|---|---|---|---|
| | Study ID | Code ("D" for academic studies and "A" for grey literature), followed by sequential numbers starting at 1 | GL/AL |
| | Name | Title of article | GL/AL |
| | Type | Publication type (journal, conference) | AL |
| | Year | Year of publication | GL/AL |
| | URL | URL of publication | GL |
| | Source | Source of publication (company, government, communication, code repository, community, blog) | GL |
| | Institution | Institution name | GL |
| | Contribution type | Report, blog post, promotion, white paper, presentation, video, audio, release, other | GL |
| RQ0 | Papers frequency | Publication type, number of publications per year | GL/AL |
| RQ1 | Research Type | Evaluation research, solution proposal, validation research, philosophical research, opinion paper, personal experience paper Wieringa et al. (2006) | GL/AL |
| RQ1 | Methodologies | Methods used to describe the research (UML models, ontology, block diagram, code, formal analysis, text only) | GL/AL |
| RQ1 | Validation Type | Case study Wohlin et al. (2012), experiments Wohlin et al. (2012), illustrative examples, qualitative analysis, performance analysis, proof of concept | GL/AL |
| RQ2 | Security Mechanisms | Security mechanisms reported in the studies analyzed | GL/AL |
| RQ2 | Security Scope | Focus of an analysis study | GL/AL |
| RQ3 | Security contexts | Mitigate/stop attack, detect attack, react to attack, recover from attack Fernandez et al. (2015) | AL/GL |

**Table 2 – Quality criteria for publications.**

| QC id | Question |
|---|---|
| – | **For Academic Literature: does the study...** |
| QC1 | have a clear description of research objectives? |
| QC2 | describe the problem addressed by the security solution? |
| QC3 | describe the security solution? |
| – | **For Grey Literature: does the study...** |
| QC4 | come from a reputable publishing organization? |
| QC5 | seem to be balanced in presentation? |
| QC6 | have clearly stated date? |
| QC7 | have conclusions supported by the data? |
| QC8 | describe the background where the security solution is being used? |
| QC9 | describe the problem addressed by the security solution? |

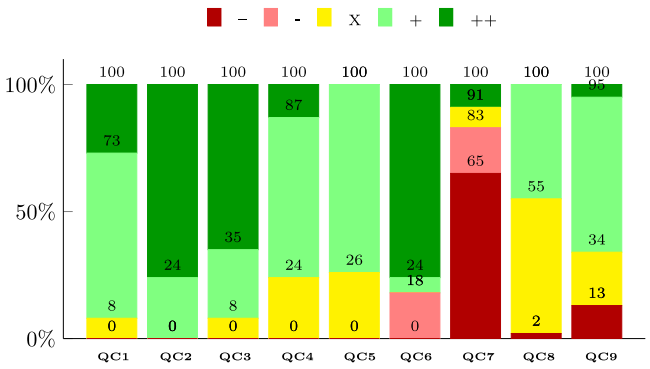



**Fig. 2 – Quality scores as percent of articles that meet each of them (academic literature: QC1 through QC3, grey literature: QC4 through QC9).**

*disagree*, and *strongly disagree*. In case of disagreement about some quality assessment, meetings were used to agree on a decision.

The quality assessment of the dataset yields some insights on the differences between academic and industrial (grey) literature, both on the aggregate and along time.

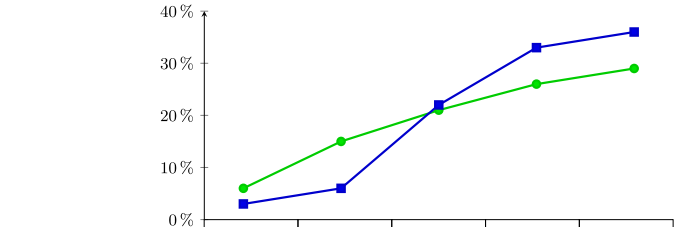
Fig. 2 summarizes the quality assessments results, for both academic and grey literature for each criterion, as percentage of the applicable total number of studies.

For academic literature studies, almost all (92%) described clearly their research objectives; all of them describe explicitly the addressed security problems; and most (92%) described explicitly the security solution provided.

For grey literature studies, regarding formality, 63% are published by professional press sites (like DZone[10]) but half of remaining (21%) are not clearly dated; and regarding argumental strength, less than half (45%) provide background, two thirds (66%) describe the problem addressed by their security solution, but only a few (5%) were deemed as having conclusions supported by the data.

The key findings of the study by Garousi et al. (2019) indicated that adding grey literature to the systematic literature review can provide benefits and certain challenges, since the evidence is based on experience and opinion. Some of the drawbacks are that we can find reports of lower quality, espe-

[10] DZone: dzone.com.




| | 2015 | 2016 | 2017 | 2018 | 2019 |
|---|---|---|---|---|---|
| AL | 3% | 6% | 22% | 33% | 36% |
| GL | 6% | 15% | 21% | 26% | 29% |

**Fig. 3 – Relevant publications per year for academic (AL) and grey (GL) literature, from 2015 to 2019.**

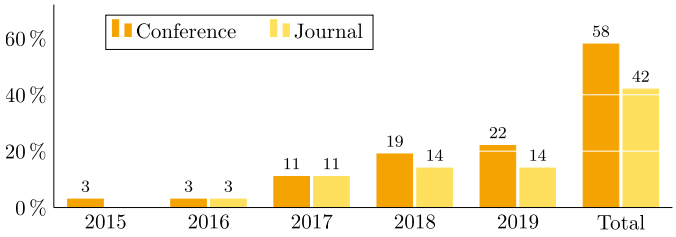



**Fig. 4 – Publication types for academic literature.**

cially when they describe the research methodology, so they must be taken into account and be careful in the different steps of the study.

### 5.1. RQ0: *Frequency of publications*

The final set of articles has 36 academic studies, labeled D1 through D36 (see Table 3), and 34 studies from the grey literature search, labeled A1 through A34 (see Table 4).

A clear trend (see Fig. 3) is that the number of publications, of both kinds of sources, has increased markedly each year. This increase can be safely attributed to the interest and maturity that the subject is reaching in both communities.

In academic literature, conference publications tie or prevail over journal articles every single year in the sample (see Fig. 4); indeed, the earliest reference found for secure microservice-based systems was published at a conference in 2015 (namely, CloudCom'2015). The number of publications in journals has grown steadily, but those in conferences have grown even faster. In the studied time interval, publications in journals added up to 42% and in conferences to the remaining 58%.

In grey literature, sources have varied widely throughout the years in the sample (see Fig. 5), but publications on professional communities have led every single year, with company publications a (sometimes close) second place. Indeed, the first publications on secure microservice-based systems (2015) are A1 and A2 in company blogs, and professional communities InfoQ[11] and DZone[12] still have the most publications on the topic. Surprisingly, the number of publications in professional communities peaked in 2018 and dropped in 2019, but the slack was more than compensated by publications in company blogs and government sites.

In the final count, over half of publications (53%) appeared in professional community publications, almost one third (26%) in company sites, and the reminder split among blogs (9%), government sites (6%), and communications (3% each). Clearly, professional communities remain a safe bet to stay updated on the topic from a professional yet rigorous standpoint.

### 5.2. RQ1: *Methodological approaches*

The selected studies from academic and grey literature where classified as *research type*, *validation type* or *methodologies used*.

#### 5.2.1. *Research types*

The selected studies from academic and grey literature where classified by *research type* using Wieringa et al. (2006) criteria:

- *Evaluation research:* Addresses investigation of a problem in practice or implementation of a technique in practice.
- *Proposal of solution:* Proposes a solution technique and argues for its relevance, without a full-blown validation.
- *Validation research:* Investigates the properties of a solution proposal that has not yet been implemented in practice.
- *Philosophical papers:* Sketches a new way of looking at things, a new conceptual framework, etc.
- *Opinion papers:* Contains the author opinion about what is wrong or good about something, how we should do something, etc.
- *Personal experience papers:* The emphasis is on what and not on why. The experience may consider one project or more, but it must be the authors personal experience.

[11] InfoQ: www.infoq.com/.
[12] DZone: www.dzone.com/.

**Table 3 – Primary studies in academic literature.**

| ID | Title | Authors | Venue | Year | Ref. |
|---|---|---|---|---|---|
| D1 | Security-as-a-Service for Microservice -Based Cloud Applications | Y. Sun et al. | Conf. | 2015 | Sun et al., 2015 |
| D2 | Secure Cloud Micro Services using Intel SGX | Brenner et al. | Conf. | 2017 | Brenner et al., 2017 |
| D3 | Integrating Continuous Security Assessments in Microservices and Cloud Native Applications | .K. A. Torkura et al. | Conf. | 2017 | Torkura et al. (2017) |
| D4 | Securing IoT Microservices with Certificates | M.O. Pahl et al. | Conf. | 2018 | Pahl and Donini, 2018 |
| D5 | Overcoming Security Challenges in Microservice Architectures | T. Yarygina et al. | Conf. | 2018 | Yarygina and Bagge, 2018 |
| D6 | Defense-in-depth and Role Authentication for Microservice Systems | K. Jander et al. | Jour. | 2018 | Jander et al. (2018) |
| D7 | A Game of Microservices: Automated Intrusion Response | T. Yarygina et al. | Conf. | 2018 | Yarygina and Otterstad (2018) |
| D8 | Authentication and Authorization orchestrator for microservice-based software architectures | A. Bánáti et al. | Conf. | 2018 | Bánáti et al., 2018 |
| D9 | Integrity Protection Against Insiders in Microservice-Based Infrastructures: From Threats to a Security Framework | M. Ahmadvand et al. | Jour. | 2018 | Ahmadvand et al. (2018) |
| D10 | Graph-Based IoT Microservice Security | M.O. Pahl et al. | Conf. | 2018 | Pahl et al., 2018 |
| D11 | Requirements Reconciliation for Scalable and Secure Microservice (De)composition | M. Ahmadvand et al. | Conf. | 2016 | Ahmadvand, Ibrahim, 2016 |
| D12 | A Cyber Risk Based Moving Target Defense Mechanism for Microservice Architectures | K.A. Torkura et al. | Conf. | 2018 | Torkura et al., 2018 |
| D13 | Implementing secure applications in smart city clouds using microservices | M. Krämer et al. | Jour. | 2019 | Krämer et al. (2019) |
| D14 | Securing Microservices | A. Nehme. Pahl et al. | Jour. | 2019 | Nehme et al. (2019a) |
| D15 | Component-Based Refinement and Verification of Information-Flow Security Policies for Cyber-Physical Microservice Architectures | C. Gerking et al. | Conf. | 2019 | Gerking and Schubert, 2019 |
| D16 | eZTrust: Network-Independent Zero-Trust Perimeterization for Microservices | Z. Zaheer et al. | Conf. | 2019 | Zaheer et al., 2019 |
| D17 | Building Critical Applications Using Microservices | C. Fetzer | Jour. | 2016 | Fetzer (2016) |
| D18 | Security and Privacy for Cloud-Based Data Management in the Health Network Service Chain: A Microservice Approach | C. Esposito | Jour. | 2017 | Esposito et al. (2017) |
| D19 | Claimsware: A Claims-based Middleware for Securing IoT Services | V.M. George et al. | Conf. | 2017 | M. George and Mahmoud (2017) |
| D20 | Access Control with Delegated Authorization Policy Evaluation for Data-Driven Microservice Workflows | D. Preuveneers et al. | Jour. | 2017 | Preuveneers and Joosen (2017) |
| D21 | Low-Level Exploitation Mitigation by Diverse Microservices | C. Otterstad et al. | Conf. | 2017 | Otterstad and Yarygina, 2017 |
| D22 | Authentication and Authorization of End User in Microservice Architecture | X. He et al. | Jour. | 2017 | He and Yang (2017a) |
| D23 | A Secure Microservice Framework for IoT | D. Lu et al. | Conf. | 2017 | Lu et al. (2017) |
| D24 | All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection | M. O. Pahl et al. | Conf. | 2018 | Pahl and Aubet (2018) |
| D25 | BlendSM-DDM: BLockchain-ENabled Secure Microservices for Decentralized Data Marketplaces | R. Xua et al. | Conf. | 2019 | Xu et al. (2019b) |
| D26 | Squad: A Secure, Simple Storage Service for SGX-based Microservices | M.S. Leite da Silva et al. | Conf. | 2019 | da Silva et al., 2019 |
| D27 | Microservice Security Agent Based On API Gateway in Edge Computing | R. Xu et al. | Jour. | 2019 | Xu et al. (2019a) |
| D28 | Identity and Access Control for micro-services based 5G NFV platform | D. Guija et al. | Conf. | 2018 | Guija and Siddiqui, 2018 |
| D29 | With great abstraction comes great responsibility: sealing the microservices attack surface | C.A. Chen | Conf. | 2019 | Chen (2019) |
| D30 | Towards multi-party policy-based access control in federations of cloud and edge microservices | D. Preuveneers et al. | Conf. | 2019 | Preuveneers and Joosen, 2019 |
| D31 | A Web Service Security Governance Approach Based on Dedicated Micro-services | S. Abidi et. al | Conf. | 2019 | Abidi et al. (2019) |
| D32 | Fine-Grained Access Control for Microservices | A. Nehme et al. | Conf. | 2018 | Nehme et al., 2019 |
| D33 | Towards Automated Inter-Service Authorization for Microservice Applications | X. Li et al. | Conf. | 2019 | Li et al., 2019 |
| D34 | Practical Defense-in-depth Solution for Microservice Systems | K. Jander et al. | Jour. | 2019 | Jander et al. (2019) |
| D35 | Applying Spring Security Framework and OAuth2 To Protect Microservice Architecture API | Q. Nguyen et al. | Jour. | 2019 | Nguyen and Baker (2019) |
| D36 | Microservices API Security | J. Salibindla et al. | Jour. | 2018 | Salibindla, 2018 |

**Table 4 – Studies in grey literature. Table A.7 shows the texts of these hyperlinks.**

| ID | Year | Title | Type | Company |
|---|---|---|---|---|
| [A1] | 2015 | API Gateway. An Introduction to Microservices | blog post | Auth0 |
| [A2] | 2015 | How To Control User Identity Within Microservices | blog post | Nordic APIs |
| [A3] | 2016 | Tutorial: Establish Trust Between Microservices with JWT and Spring Boot | blog post | Stormpath |
| [A4] | 2016 | Rethinking Application Security With Microservices Architectures | blog post | Dark Reading |
| [A5] | 2017 | Data-Centric Microservices Security | community | DZone |
| [A6] | 2017 | Security Enforcement of Microservices | community | DZone |
| [A7] | 2017 | Securing Microservices: A Brief Look at Different Technologies | community | DZone |
| [A8] | 2017 | 7 Best Practices of Microservices Security | blog post | Newizze |
| [A9] | 2017 | Security Standard-Microservices Architecture (SS-028) | government | Department for Work & Pensions UK |
| [A10] | 2017 | Microservice Architecture: API Gateway Considerations | company | Global Logic |
| [A11] | 2017 | Microservices security with OAuth 2 | ad-hoc blog | Piotr's Techblog |
| [A12] | 2017 | Authorization and Authentication with Microservices | company | LeanIX |
| [A13] | 2017 | How Do You Secure Microservices? | community | DZone |
| [A14] | 2017 | Advanced Microservices Security with Spring and OAuth 2 | community | DZone |
| [A15] | 2018 | Security Approaches for Microservice Architectures | community | Linux Foundation |
| [A16] | 2018 | Microservices Authentication & Authorization Best Practice | community | Codeburst.io |
| [A17] | 2018 | Security | community | Keitaro Inc. |
| [A18] | 2018 | Microservices Authentication and Authorization Solutions | community | Medium |
| [A19] | 2018 | Microservice and Container Security: 10 Best Practices | company | Apriorit |
| [A20] | 2018 | Microservice and Securing microservice environments in a hostile world | company | IDG Communications, Inc. |
| [A21] | 2018 | Scale Security While Innovating Microservices Fast | community | DZone |
| [A22] | 2018 | Implement Secure Microservices With Spring Security and OAuth 2.0 | community | DZone |
| [A23] | 2018 | How a Service Mesh Can Help With Microservices Security | community | DZone |
| [A24] | 2019 | Security Strategies for Microservices-based Application Systems | government US | NIST |
| [A25] | 2019 | Pattern: Access token | ad-hoc blog | Microservices.io |
| [A26] | 2019 | Improving Security in Your Microservices Architecture | company | Sumo Logic |
| [A27] | 2019 | 8 best practices for microservices app sec | community | Techbeacon |
| [A28] | 2019 | Securing modern API- and microservices-based apps by design | company | IBM |
| [A29] | 2019 | Microservices Security \| Best Practices To Secure Microservices | company | Edureka |
| [A30] | 2019 | Service-to-Service Authentication for Microservice APIs | ad-hoc blog | Hackernoon |
| [A31] | 2017 | Learn How to Secure Service-to-Service Microservices | community | DZone |
| [A32] | 2017 | Authentication and Authorization in Microservices | community | DZone |
| [A33] | 2019 | Microservices Authentication and Authorization Using API Gateway | community | DZone |
| [A34] | 2019 | Security | community | Istio |

For all publications in the selected set, we obtained their objectives and addressed problems, and the team classified them according to the list above. Fig. 6 describes the distribution of studies in academic and grey literature by research classification throughout the years 2015 to 2019.

Almost half of studies in the academic literature (42%) propose solutions to security problems of microservice-based systems (see Fig. 7). As we indicated in Section 2, security is hard in microservice systems but we need to protect their stored or transmitted data, among other aspects related to the development of microservice-based systems. 25% of the primary studies of academic literature investigated security aspects that have not been implemented (see Fig. 7). This group of studies aims to propose novel techniques to expand the range of security solutions that can be used by microservice-based systems, such as a model of access control based on semantics, decentralized authentication models, and an access control model for 5G infrastructures, among others. Another

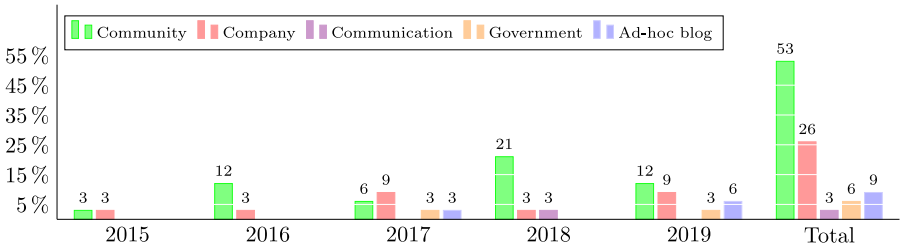



**Fig. 5 – Publication types for grey literature.**

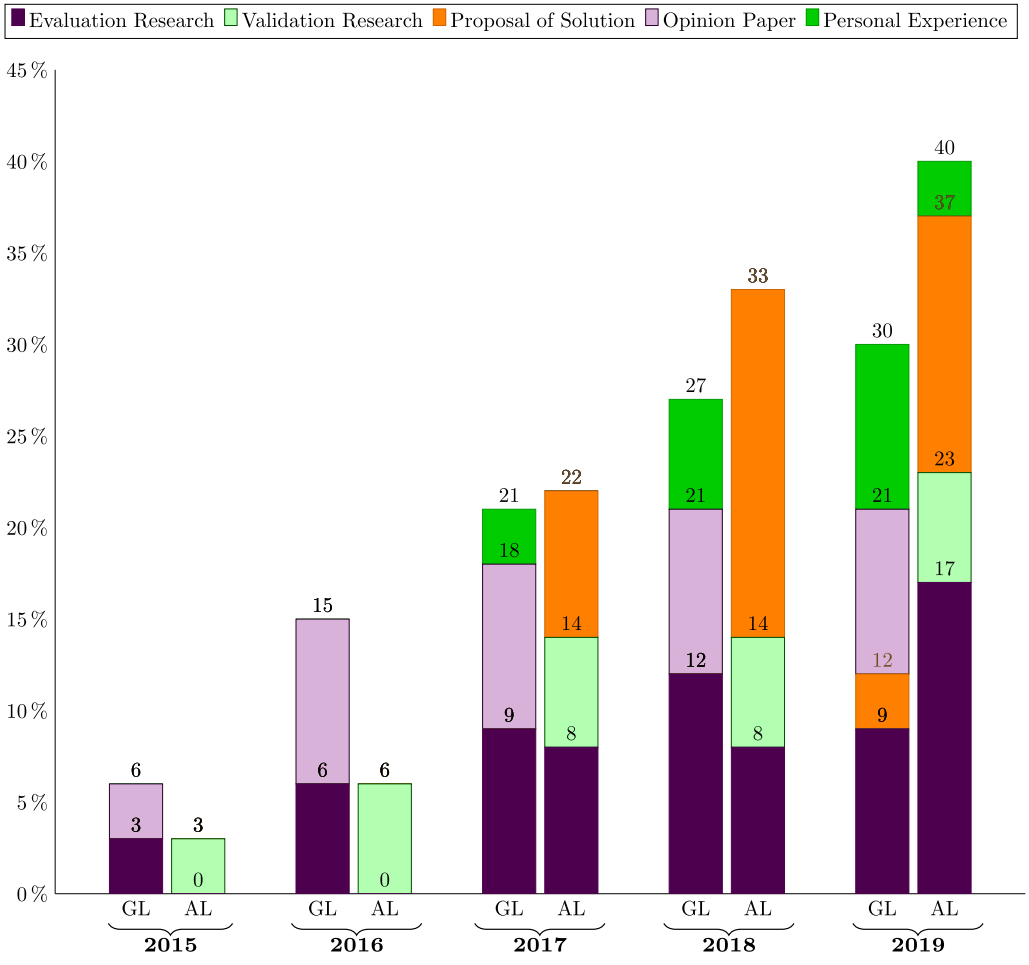



**Fig. 6 – Research type of the academic (AL) and grey (GL) literature studies.**

group of primary studies of academic literature (33%) evaluates security aspects that are already implemented (see Fig. 7). These primary studies conducted empirical studies to analyze the advantages and disadvantages of specific techniques and methods for the security for microservice-based systems. In the academic literature we did not identify primary studies that correspond to philosophical articles, opinion articles or articles from personal experience.

In the grey literature we did not find any validation studies. Almost half of them (38%, see Fig. 8) were opinion papers, where their authors explained the implementation of certain security aspects and expressed their opinion on them. In this group we found opinions on access control models, authorization, and cryptography. Another group with high prevalence included studies based on evaluation research (41%, see Fig. 8), where the authors evaluated approaches that had been previously proposed and implemented them, with authentication and authorization being the most frequently considered topics. To a lesser extent, we found studies where the authors present their personal experiences (18%, see Fig. 8) and proposed solutions to address security issues in microservice-based systems.

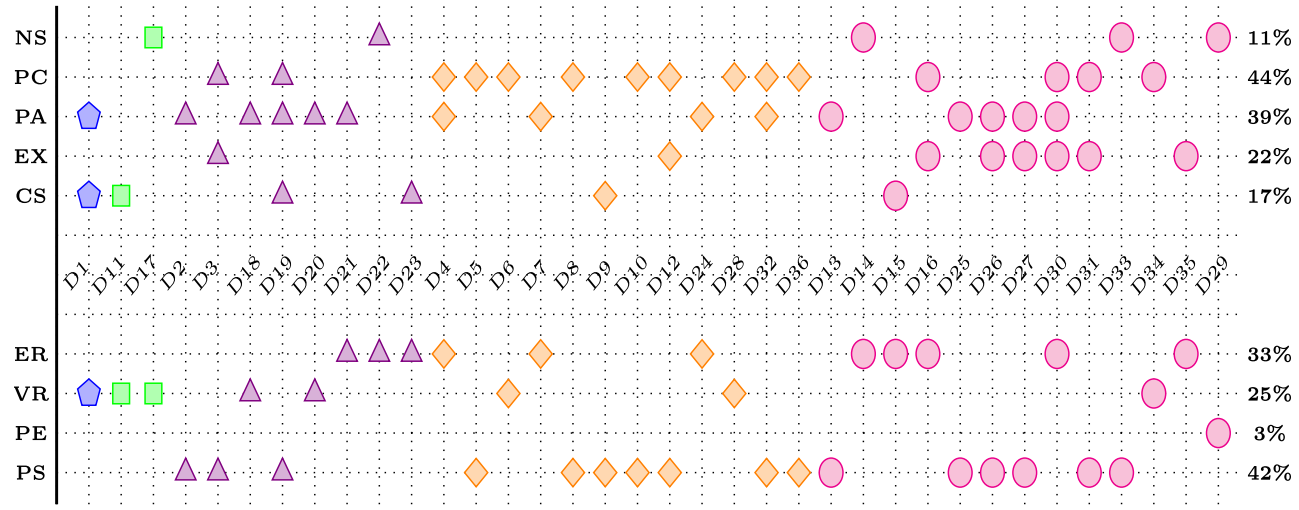



**Fig. 7 – Summary of methodological approaches to academic literature. At the top of the Y axis, we find validation types and the abbreviations used were: CS (case study); EX (experiment); PA (performance analysis); PC (proof of concept); NS (not specified). In the lower part of the Y axis, we find the research types and the abbreviations used were: ER (evaluation research); VR (validation research); PS (proposal of solution); PE (personal experience paper). The pentagon represents the papers from 2015, the square from 2016; the triangle from 2017, the diamond from 2018, and the circle the papers from 2019.**

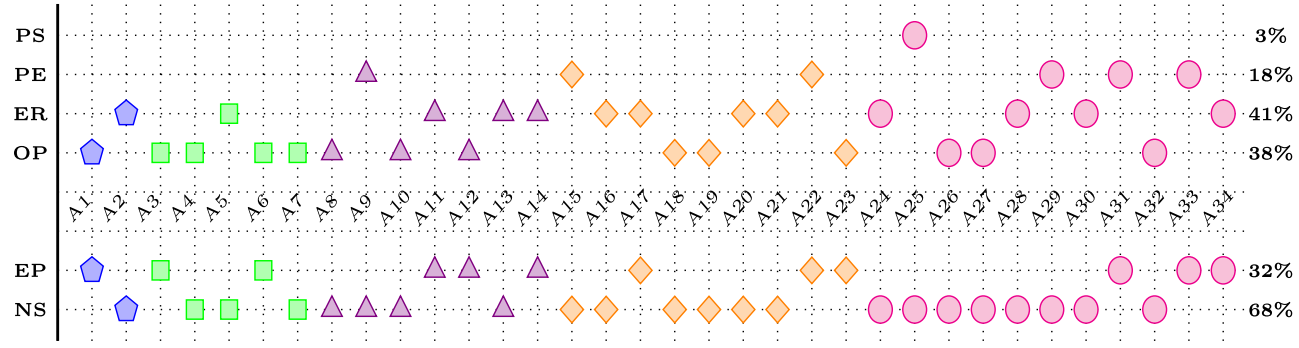



**Fig. 8 – Summary of methodological approaches to grey literature. At the top of the Y axis, we find research types and the abbreviations used were: ER (evaluation research); PS (proposal of solution); OP (opinion paper); PE (personal experience paper). In the lower part of the Y axis, we find the validation types and the abbreviations used were: EP (example) and NS (not specified). The pentagon represents the papers from 2015, the square from 2016, the triangle from 2017, the diamond from 2018, and the circle the papers from 2019.**

#### 5.2.2. *Validation types*

Each publication was classified for *validation type* using the validation taxonomy proposed by Wohlin (2014), which includes case studies, experiments, replication, and surveys. We added: proof of concepts, performance analysis, qualitative study, and illustrative example.

Almost all publications in the academic literature (89%) offer some kind of validation (see Fig. 9). The majority of them uses proof of concept (44%), and many of them carry out performance analysis. In some cases, several validation types are combined: D1 uses case studies and performance analysis, and D3 uses experiments and proof of concept. Curiously, in 2018 the favored approach was proof of concept (25%), but in 2019 the favored approach was experiments (17%). Figs. 7 and 8 show a summary of the methodological approaches, such as validation and research type for each of the studies in the academic and grey literature, respectively.

Most publication in the grey literature do not cover validation at all, and those that do (32%) do it using illustrative examples.

#### 5.2.3. *Methodologies*

Academic papers by far use block diagrams (89%) to describe a problem or solution graphically. A few (23%) of them use UML diagrams: class, sequence, deployment, use case. Only 8% use ontologies. 11% use some type of formal analysis.

Grey papers use almost exclusively block diagrams (32%) and code (38%) or just text descriptions (41%). Only one paper had UML sequence diagrams (3%). The lack of use of models by practitioners is a serious problem because the complexity of

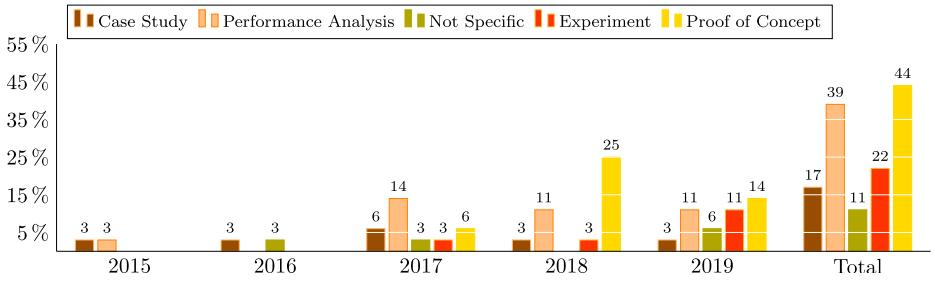



**Fig. 9 – Validation types of research in academic literature in security microservice-based systems.**

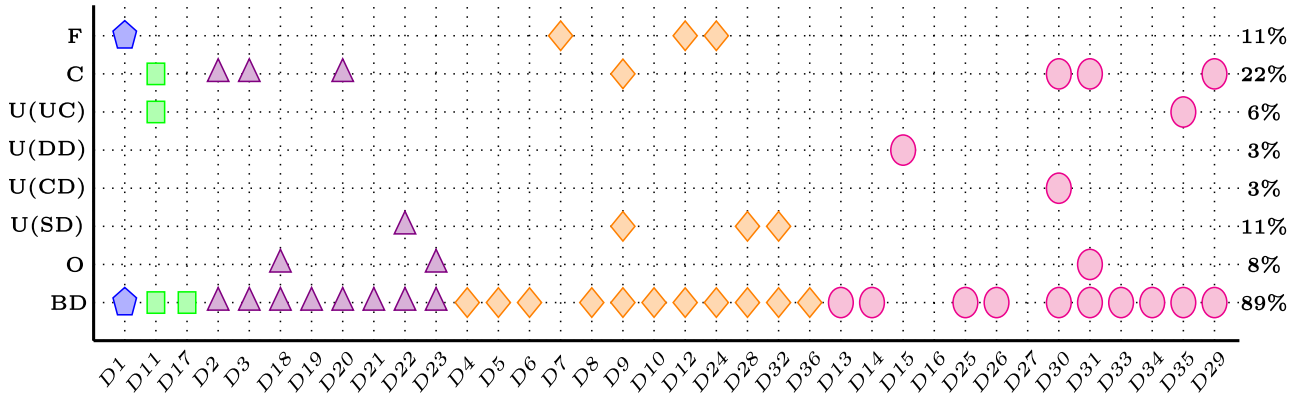



**Fig. 10 – Summary of methodologies used to grey literature. In the Y axis, we find** methodologies **and the abbreviations used were:** BD **(block diagram);** O **(ontology);** U (SD) **(UML sequence diagram);** U (CD) **(UML class diagram);** U (DD) **(UML deployment diagram);** U (UC) **(UML use case model);** T **(text only);** C **(code);** F **(formal analysis). In the X axis the grey studies ID. The pentagon represents the papers from 2015, the square from 2016, the triangle from 2017, the diamond from 2018, and the circle the papers from 2019.**

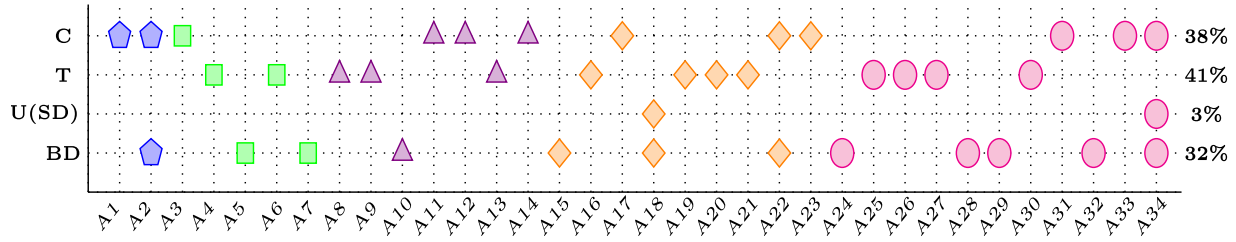



**Fig. 11 – Summary of methodologies used to grey literature. In the Y axis, we find** methodologies **and the abbreviations used were:** BD **(block diagram);** U(SD) **(UML sequence diagram);** T **(text only);** C **(code). In the X axis the grey studies ID. The pentagon represents the papers from 2015, the square from 2016, the triangle from 2017, the diamond from 2018, and the circle the papers from 2019.**

security solutions requires a higher level of abstraction. This weakness may allow to see more possible attacks in the future.

Figs. 10 and 11 represent the methodologies used by each study in the grey and academic literature, respectively.

Fig. 12 shows the use of methodologies by academic and grey literature along the years.

### 5.3. *RQ2: Security solutions classification*

In order to identify the security mechanisms associated with each proposed security solution, each team member assigned these security solutions in a shared spreedsheet, and any disagreements were discussed to yield unanimous assessments. We used the classification of Uzunov et al. (2015), which includes: **Authorization, Identity Management/Authentication, Access Control, Secure Communication, Filtering, Storage Security, Logging and Monitoring, Security Information Management** and **Execution Control**. We also added other classifications to include aspects of security that do not correspond to mechanisms such as **Application Security, Implementation Security, Security Evaluation, Threat Modeling**, and **General Security Architecture**. We call them security scopes. Fig. 13

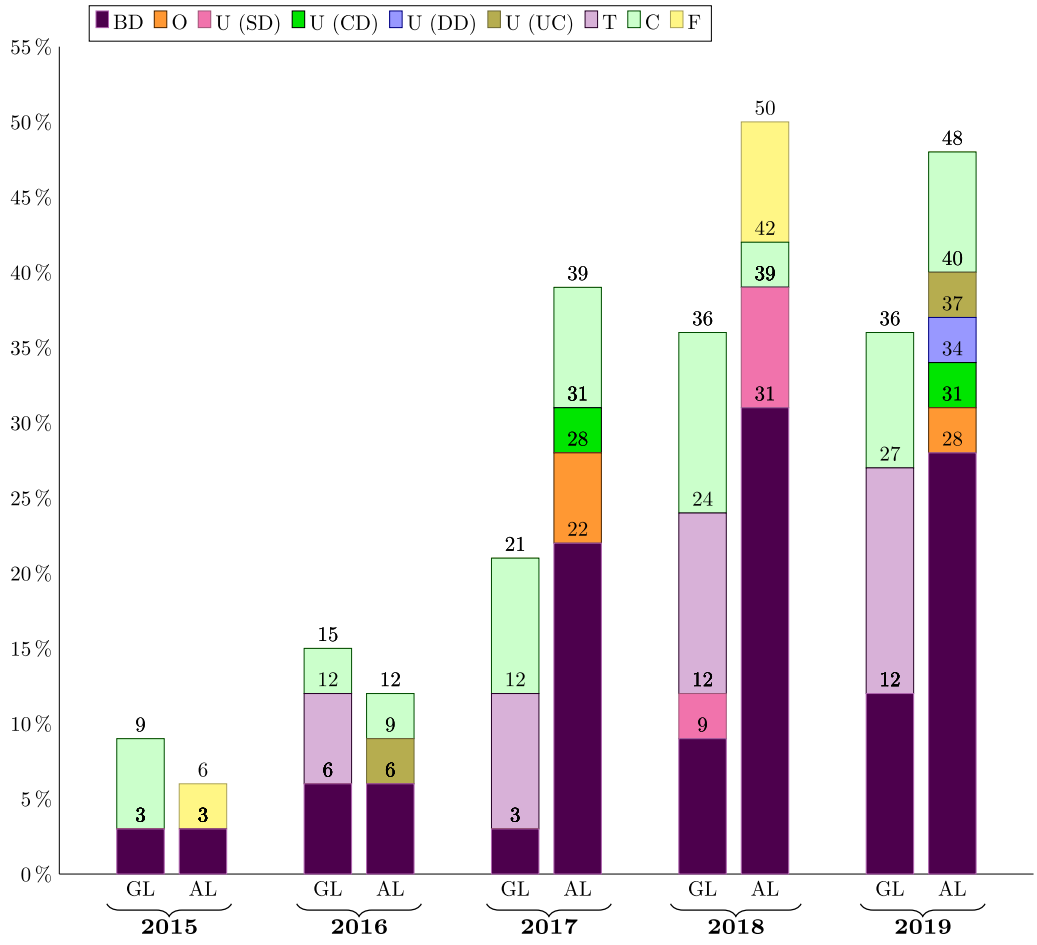



**Fig. 12 – Methodologies used of the academic (AL) and grey (GL) literature studies. The abbreviations used were: BD (block diagram); O (ontology); U (SD) (UML sequence diagram); U (CD) (UML class diagram); U (DD) (UML deployment diagram); U (UC) (UML use case model); T (text only); C (code); F (formal analysis).**

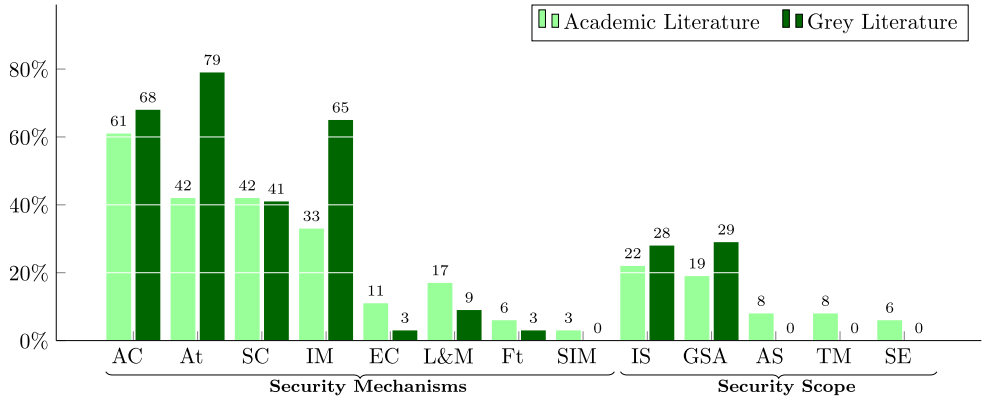



**Fig. 13 – Security mechanisms and security scope identified in studies of academic and grey literature on microservice-based systems. The initials used were: At (Authorization); IM (Identity Management/Authentication); AC (Access Control), SC (Secure Communication); Ft (Filtering); L&M (Logging & Monitoring); EC (Execution Control); SIM (Security Information Management); IS (Implementation Security); SE (Security Evaluation); TM (Threat Modeling); GSA (General Security Architecture); AS (Application Security).**

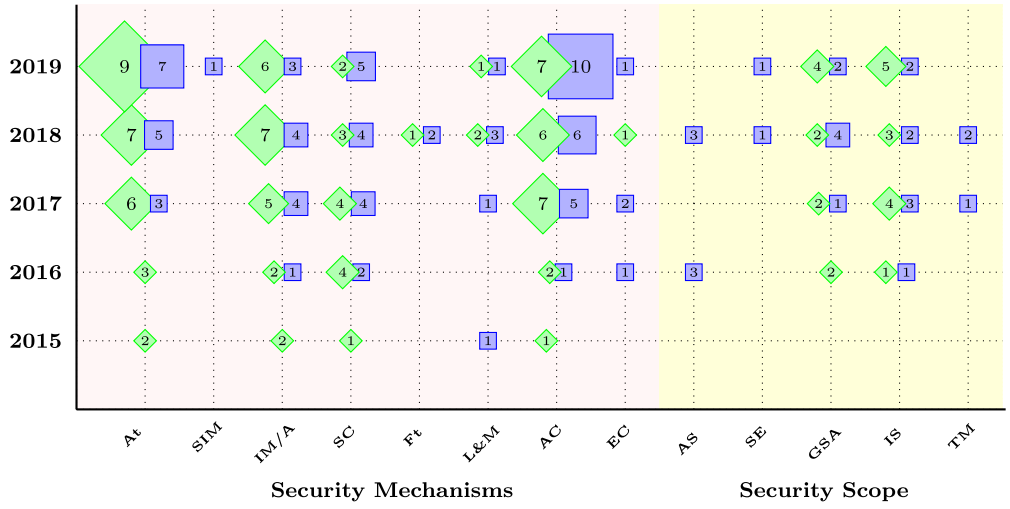



**Fig. 14 – Security mechanisms and security scope of the academic and grey studies throughout the years studied.** Diamonds **represent grey literature studies, whereas** squares **represent academic literature studies. The numbers inside are quantities of papers. The acronyms are as follows:** At **(Authorization);** IM/A **(Identity Management/Authentication);** AC **(Access Control),** SC **(Secure Communication);** Ft **(Filtering);** L&M **(Logging & Monitoring);** EC **(Execution Control);** SIM **(Security Information Management);** IS **(Implementation Security);** SE **(Security Evaluation);** TM **(Threat Modeling);** GSA **(General Security Architecture);** AS **(Application Security).**

shows the distribution of security mechanisms and security scope of the proposed solutions for microservice-based system security.

Fig. 14 shows the behavior of the security scopes and security mechanisms reported by the academic and grey literature throughout the years studied.

The following sections explore the results for each the security mechanisms and security scopes of the studies selected. Table 5 shows the reported solutions, grouped according to security mechanisms and security scope and for each, the identifiers of the papers that reported the security solution.

### 5.3.1. *Security mechanisms*

***Authorization***

Authorization describes the rights of subjects (users, processes) with respect to the system resources, i.e., who can access specific resources in the system and in what way. 79% of the grey literature and 42% of the academic literature report authorization-related solutions. Some of the solutions proposed in the academic literature to manage authorization are flow-oriented authorization frameworks that apply authorization policies in a decentralized manner, e.g., D20;; most solutions for authorization problems use standards and frameworks such as OAuth 2.0. D30 presents delegated authorization to multiple stakeholders (service providers and consumers, content owners, and others). D33 proposes authorization between services, which must follow the business logic of applications and comply with the principle of least privilege; for this purpose they use Jarvis, an automated inter-service authorization mechanism for microservice applications.. As expected, the grey literature shows more practical solutions and implementations; for example, a demonstration of the operation of Spring-based authorization (A3, A6, A14, A22, A31).

In general, both kinds of literature show solutions to implement authorization of the type RBAC (Role-Based Access Control), OAuth, PBAC (Policy-Based Access Control), XACML (eXtensible Access Control Markup Language), and Multilevel Security.

***Identity Management***

It is the organizational process to identify individuals and/or groups of people or active software units. Identities are fundamental for authentication, authorization, and logging/auditing. In our study, we found that 65% of the grey literature and 33% of academic literature reported solutions that use identity management to add security to their microservice-based systems. Protocols such as TLS (Transport Layer Security), MTSL (Mutual Transport Layer Security), SASL (Simple Authentication and Security Layer), X509 Certificates, and SAML (Security Assertion Markup Language) make use of some type of identity management for their authentication functions.

Both grey and academic literature agree (D8, D13, D14, D20, D22, D27, D28, D30, D32, D30, D35, A1, A2, A6, A7, A8, A10, A11, A13, A13, A14, A15, A17, A18, A19, A22, A24, A26, A27, A28, A31, A33, A34) that the most appropriate models and standards for authentication in microservices architectures are: (i) OAuth, (ii) OpenID Connect, (iii) Single Sign-On (SSO) solutions, (iv) JSON Web Token (JWT) (see Fig. 15).

***Access Control***

Implies the systematic verification that an entity requesting access to a resource is a legitimate subject and has the necessary rights to do so. Access control has two essential parts: authentication and authorization. In our study, 68% of grey literature studies and 61% of academic literature reported access

**Table 5 – Security scope and security mechanisms reported by academic and grey literature studies.**

| Security scope & mechanisms | AL | GL | ID Studies |
|---|---|---|---|
| *Mechanisms* | | | |
| **Authorization** | AL | GL | – |
| RBAC | 6% | 6% | D4, D5, A13, A30 |
| OAuth | 17% | 65% | D13, D14, D28, D30, D32, D34, A1, A2, A6, A7, A8, A10, A11, A12, A13, A14, A15, A17, A18, A19, A22, A24, A26, A27, A28, A31, A33, A34 |
| PBAC | 3% | - | D4 |
| ABAC | 17% | - | D23, D4 |
| XACML | 11% | - | D20, D30, D31, D32 |
| Multilevel Security | 3% | - | D13 |
| **Identity Management** | **AL** | **GL** | – |
| SAML | 3% | 6% | D19, A10, A12, A17, A24, A28, A30 |
| TLS | 14% | 9% | D2, D6, D17, D19„D26, A1, A6, A7, A9, A10,A19, A28 |
| MTLS | 3% | - | D5, A6, A23, A34 |
| SASL | 3% | - | D19 |
| Credential | 22% | 13% | D5, D8, D9, D13, D14, D18, D20, D22, A7, A8, A10, A11, A15, A16, A25, A28, A30 |
| Certificates | 8% | | D4, D5, D9 |
| Authentication | 11% | 14% | D4, D8, D22,D27, A1, A4, A6, A31 |
| Identity Provider | - | 9% | A2, A10, A17, A20, A27, A28, A34 |
| **Access Control** | **AL** | **GL** | – |
| Authentication & Authorization | 44% | 68% | D3, D5, D6, A10, D11, D12, D13, D14, D16, D18, D19, D20, D23, D25,D26, D28, D31, D33, A10, A11, A12, A13, A14, A15, A16, A19, A2, A3, A5, A8, A9, A22, A24, A26, A27, A28, A32 |
| Graph-Based Access Control | 3% | - | D10, A17, A18, A33 |
| Information Flow | 3% | - | D15 |
| **Secure Communication** | **AL** | **GL** | – |
| Cryptography | 3% | 38% | D6, A3, A4, A5, A6, A10, A12, A13, A15, A19, A20, A24, A27 |
| Secure Data Exchange | 3% | - | D25 |
| 5G & NFV Security | 3% | - | D28 |
| Encryption | 33% | - | D2, D11, D13, D14, D16, D17, D18, D2, D20, D23, D34, D5, D9, D38 |
| Health Data Exchange | 3% | - | D18 |
| Segmentation | - | 3% | A8 |
| **Logging & Monitoring** | **AL** | **GL** | – |
| IDS | 11% | 3% | D1, D7, D21, D33, A15, A20 |
| **Filtering** | 6% | 3% | D10, D23, A15 |
| **Execution Control** | 11% | 3% | D2, D17, D21, D26, A15 |
| **Security Information Management** | **AL** | **GL** | – |
| Web Service Security Governance | 3% | - | D31 |
| *Scope* | | | |
| **Application Security** | **AL** | **GL** | – |
| Security Development Methodology | 5% | - | D11, D36 |
| Secure Application Development | 3% | - | D13 |
| **Implementation Security** | **AL** | **GL** | – |
| Edge Computing | 3% | - | D27 |
| API Security | 3% | 9% | A14, A31, A33 |
| Security Gateway | 19% | 32% | D11, D18, D22, D23, D27, D28, D30, A6, A10, A11, A13, A18, A19, A25, A26, A28, A33 |
| Chain of Trust | - | 3% | A21 |
| **Security Evaluation** | **AL** | **GL** | – |
| Metrics | 3% | - | D12 |
| Security Evaluation | 3% | - | D35 |
| **Threats Modeling** | **AL** | **GL** | – |
| Vulnerability Analysis | 3% | - | D3 |
| Defense | 3% | - | D12 |
| Threats | 6% | - | D9, D12 |
| Moving Target Defenses | 3% | - | D12 |
| **General Security Architecture** | **AL** | **GL** | – |
| Defense in Depth approach | 11% | 15% | D34, D5, D6, D7, A9, A15, A19, A27, A29 |
| SRA | 3% | - | D14 |
| General Security | 6% | 21% | D5, D36, A4, A13, A19, A26, A27, A29, A34 |
| Secure Lifecycle | 3% | - | D14 |
| Security Patterns | 3% | 6% | D23, A6, A25 |
| Security Requirements | - | 3% | A9 |

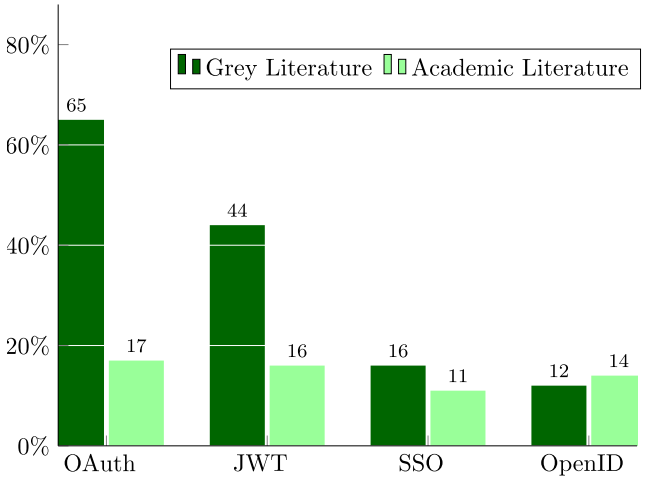



**Fig. 15 – Different standards and patterns used to implement authentication reported by the academic and grey literature.**

control in their security solutions. One of them (D10) shows a graph-based access control mechanism that runs as a unit on IoT nodes or in the network.

***Secure Communication***

Secure communication is normally obtained through encryption and mutual authentication. Messages are encrypted at the source and decoded at the destination to prevent intruders from reading any transmission they may intercept. Mutual authentication permits to establish a secure connection. Network segmentation is used to isolate sensitive units. 42% of academic studies and 41% of grey literature report the use of some method to guarantee secure communication. The reported methods are some variety of Cryptography, Secure Data Exchange, 5G and NFV Security, and Network Segmentation (see Table 5).

***Filtering***

It is used for restricting communications between a site and possibly suspicious sites. It is normally performed using several types of firewalls. For example, D10 describes an extension to the architecture to filter interservice traffic, which automatically creates a model of legitimate communication relationships. We found few studies that reported the use of solutions aimed at filtering. Only 3% and 6% in grey and academic literature, respectively. This reflects the fact that filtering is not in general tailored to applications or to the use of microservices, although there are XML and application firewalls that filter application aspects.

***Logging & Monitoring***

Logging is the recording of events that access information in a system. Application monitoring is used to detect anomalous behavior that may indicate an attack. The most reported solutions were solutions for detection of attacks (IDS); for example D24 presents a machine-learning-based approach for modeling IoT service behavior by observing inter-service communication. This solution makes possible retrofitting access control to existing non-secure IoT systems. Another paper, D1, considers network monitoring to detect attacks, while A10 considers health monitoring of services. Only 9% of grey literature and 17% in academic literature report solutions of this type.

***Execution Control***

An executing process needs to be protected from interference from other processes; at the same time, its execution should be restricted to a specific execution domain (a sandbox), where it can only use authorized resources. As an example, D17 shows the use of hardware-supported sandboxes (SGX), where critical processes can be protected from interference from other processes. Very few studies reported its use, only 3% in the grey literature and 11% in the academic literature

***Security Information Management***

Authentication and authorization require the storage of authentication information and authorization rules, respectively. This information must be protected or these fundamental functions would be disrupted. In the microservice-based systems security studies we found a minimum amount that allowed us to mention them in our research, since we only found one reference, which represents 3% in the academic literature.

### 5.3.2. *Security scope*

***Application security***

Application security should be based on its semantics and the lower levels should just enforce the restrictions defined at this level Pardon et al. (2018). The lifecycle steps to build applications should keep the application semantics defined in the requirements. In our review, we found that barely 8% of the academic literature reported solutions aimed at application security. One of the solutions prescribes a conceptual methodology that should reflect the application semantics in the service decomposition (D11). Another proposal, presents the design of a software architecture that can be used as a template for the implementation of Smart City applications (D13).

***Implementation Security***

These articles described the effect of platforms and other architectural aspects such as edge computing, gateways, APIs, and chain of trust (bootstrap). The academic literature reported 22% of these studies, and of them 16% consider API Gateways. D27 proposes microservice security agents to integrate the edge computing platform with the API gateway technology for assembling an authentication mechanism. On its part, the grey literature reports 28% of studies related to implementation security and within them, 34% use API Gateway. A21 uses a Trusted Platform Module (TPM) on the host to store the measurements resulting from the security model to measure and verify each component, to create a trust limit and build a chain of trust from its root.

***Security Evaluation***

There are a number of metrics to determine the degree of security of a system, although none of them is broadly accepted. Very few publications address security evaluation. D12 introduces the notion of diversification index as a security metric to express the depth of diversification of microservices; software diversification aims to thwart attackers by randomizing attack surfaces so that attackers are confused, thereby reducing motivation to attack and overall attack effectiveness.

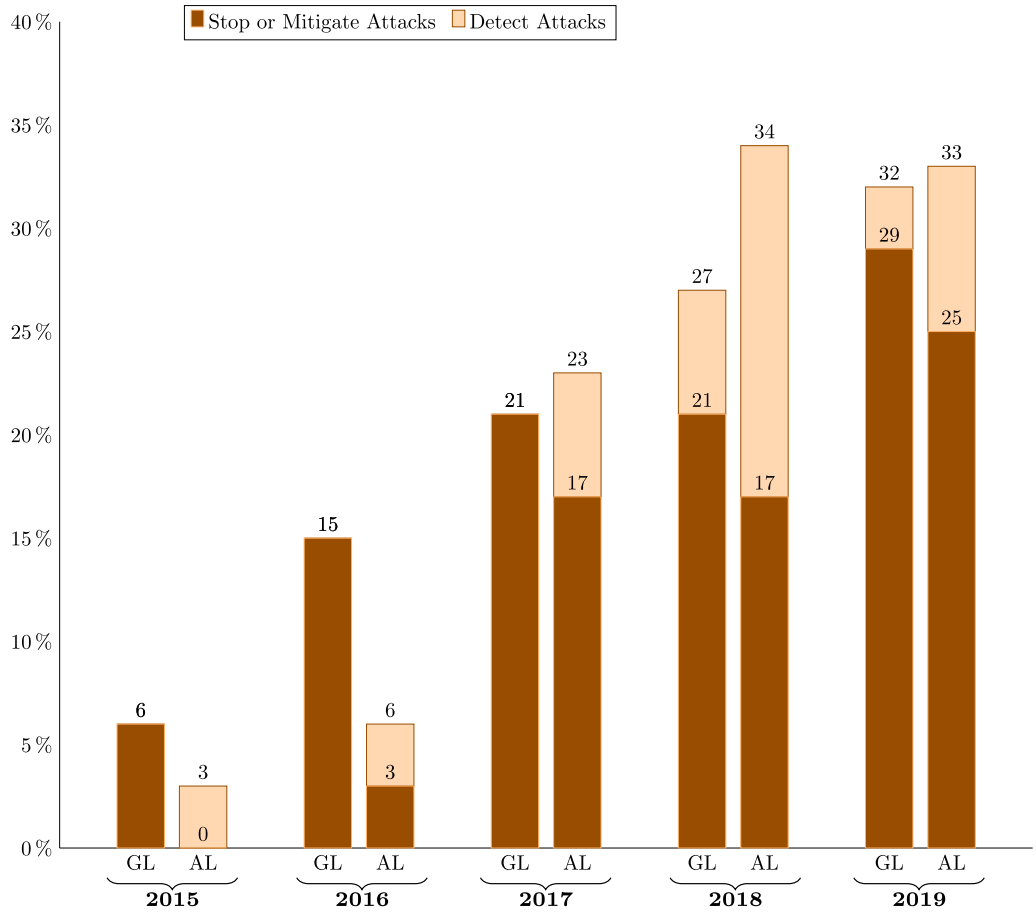



**Fig. 16 – Security contexts addressed by the security solutions. AL (Academic literature) and GL (Grey Literature).**

*Threat Modeling*

Threat modeling implies defining a threat model and being able to enumerate them for a given system. Threats exploit vulnerabilities in systems and some studies emphasize discovering vulnerabilities instead of threats. Only 8% of the primary studies in the academic literature address threat modeling. D9 identifies representative integrity threats to microservice-based systems by analyzing real-world systems. They also propose a framework that serves as a model for the protection of the integrity of sensitive information in microservices. D12 proposes a mechanism called Moving Target Defense, which after performing a risk analysis to detect and prioritize vulnerabilities in microservice-based systems shows a way to confuse attackers. D13 introduces a methodology to integrate continuous security assessment into microservices, allowing to effectively detect various vulnerabilities.

*General Security Architecture*

There are some articles on general aspects of security for microservices such as security principles (only defense in depth), such as D5, D6, D34, secure development lifecycles (D34), a few patterns (A6,A25), and a Security Reference Architecture (D14).

### 5.4. RQ3: Security contexts

Figure 16 shows the classification of security solutions from academic and grey literatures, according to the security contexts defined in Section 4.1.

Most of the mentioned solutions in the academic and grey literature describe security solutions that aim to mitigate/stop attacks (61% and 91% respectively); and indeed, Access Control and Identity Management mechanisms have the largest presence among studies. Almost all other identified mechanisms aim to detect attacks (36% in the academic literature and only 9% in the grey literature).

Figs. 17 and 18 shows a summary for each of the primary studies in the academic and grey literature respectively.

Each figure shows the security scope and/or security mechanisms reported by every academic and grey literature study. The circle represents the solutions directed to stop or mitigate attacks and the diamond the solutions that detect attacks. We did not find solutions that could be related to react to attacks and recover from attacks in either kind of literature.

### 5.5. Emerging challenges

The search for academic and grey literature ended in May 2020. In the final results they are not included because they were only for four months from 2020. The grey and academic

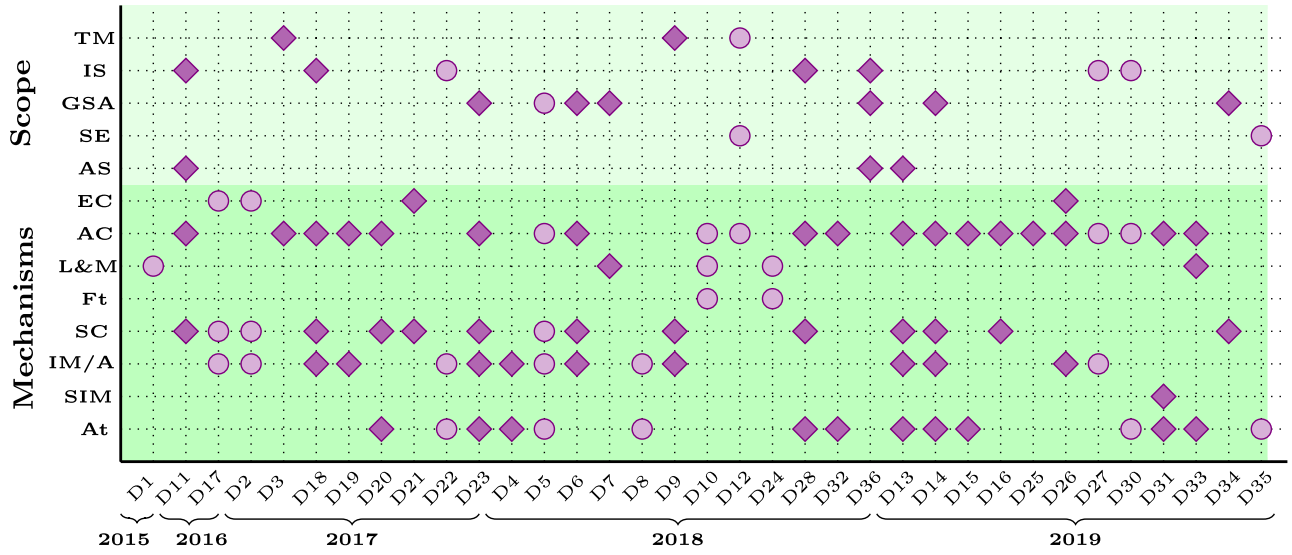



**Fig. 17 – Summary of the security scope and security mechanisms reported by each study in the academic literature.** Diamonds **represents the solutions directed to stop and mitigate attacks and the** circles **represents the solutions directed to detect attacks. The acronyms are follows:** At **(Authorization);** IM/A **(Identity Management/Authentication);** AC **(Access Control),** SC **(Secure Communication);** Ft **(Filtering);** L&M **(Logging & Monitoring);** EC **(Execution Control);** SIM **(Security Information Management);** IS **(Implementation Security);** SE **(Security Evaluation);** TM **(Threat Modeling);** GSA **(General Security Architecture);** AS **(Application Security).**

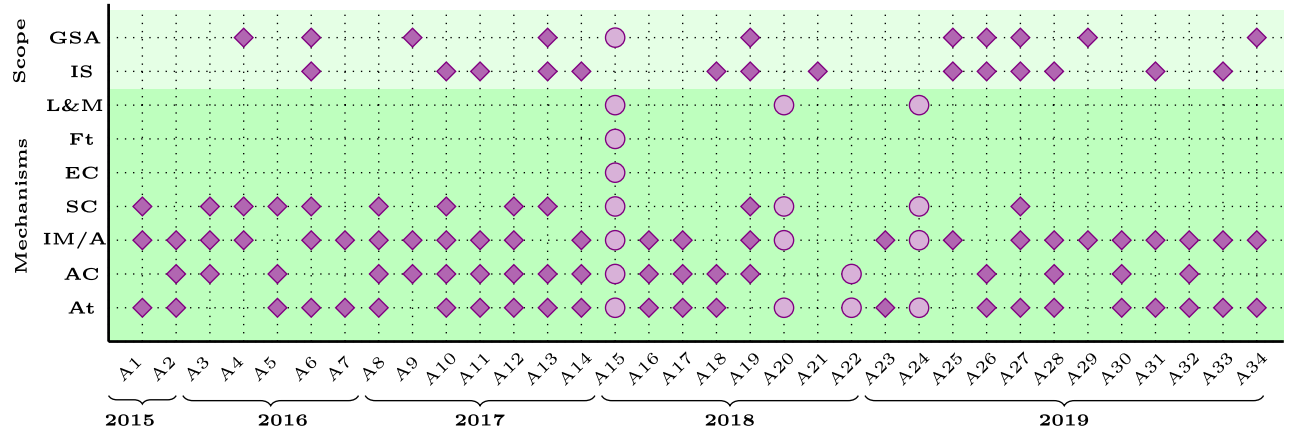



**Fig. 18 – Summary of the security scope and security mechanisms reported by each study in the grey literature.** Diamonds **represents the solutions directed to stop and mitigate attacks and the** circles **represents the solutions directed to detect attacks. The acronyms are follows:** At **(Authorization);** IM/A **(Identity Management/Authentication);** AC **(Access Control),** SC **(Secure Communication);** Ft **(Filtering);** L&M **(Logging & Monitoring);** EC **(Execution Control);** IS **(Implementation Security);** GSA **(General Security Architecture).**

literature selected was subjected to the same rigorous selection process as the final studies that form part of the MLR. In this section, we will talk briefly about the trends perceived in the few studies found in the first four months of 2020.

Regarding academic literature, we only found one paper published in the IEEE Transactions on Parallel and Distributed Systems Journal. This paper by Wen et al. (2020) describes dependable microservice orchestration framework GA-Par to effectively select and deploy microservices whilst reducing the discrepancy between user security requirements and actual service provision. They develop a parallel genetic algorithm framework based on Spark to accelerate the operations for calculating the optimal or near-optimal solution.

On the other hand, in the grey literature, we find a special NIST [13] publication that provides an implementation guide for the proxy-based Service Mesh components that together form the basis for secure communications between microservices.

[13] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-204A.pdf.

Two other articles [14][15] of 2020, address security mechanisms such as Authentication, Identity Management/Authentication, Logging & Monitoring and Secure Communication. This last mechanism is implemented using cryptography and segmentation. For authentication, they also use the OAuth, JWT, SSO, and OpenID patterns. It also mentions the use of TLS, identity provider and credentials for identity management. On the other hand, at [16] addresses issues related to implementation security, specifically with API security and security gateway. Raible [17] recommends a set of patterns and techniques to secure microservices architectures.

### 5.6. *Summary of results*

Fig. 19 shows an integrated summary of the results of *RQ1, RQ2* and *RQ3*. On the central vertical axis, we have the list of *security mechanisms and security scope* identified in the articles analyzed. To the right of the vertical axis, on the X axis are the *methodologies* used in the studies. To the left of the vertical axis, we have the *validation type* used by the authors in their proposals. The intersection of the vertical axis and the X axis is represented by a geometric figure. Five different figures were defined for the *research type* identified in the studied literature. In addition, two colors were used that represent the *security contexts* identified in the review. The number within each figure represents the number of articles from both literatures with the following meaning. Towards the left side of the vertical axis, each number is the number of articles identified according to the security mechanism or security scope, the validation type used, the security context, and the type of investigation. Similarly, towards the right side of the vertical axis, the number represents the number of articles according to the security mechanism or security scope, the methodology used, the security context, and the research type.

## 6. Directions for research

From our results we can find several directions which can prove fruitful for future work.

### 6.1. *Lack of attention to "react to"/"recover from" attack security contexts*

In the academic and industrial sources (see Table 3 and 4) there is a lack of attack recovery proposals, which can be explained because the existing methods to recover from security attacks do not depend on the specific approach used to write the applications. The only work we found about microservices recovery is an article of Pardon et al. (2018), which addresses the problem of backing up a complete microservice architecture where a running application broken down into multiple microservices needs a backup that can be recovered after a serious disruption. Their main objective is the total availability of the application. They state that, depending on the type of technique chosen for support, it will be necessary to suspend normal operation and only allow reading operations to ensure that the underlying state of each microservice is consistent. This is a topic that deserves more attention.

### 6.2. *Little work on intrusion detection and monitoring systems*

Intrusion detection (monitoring) systems are not generally concerned with specific types of applications but they usually analyze network traffic to detect the signature or behavior of attacks. Some monitoring systems specialized to detect anomalies in microservice application execution have appeared, e.g. D1, but more work is needed. It could be interesting to enumerate systematically attacks to microservices so their signatures or behavior can be catalogued and used by a specialized monitoring system.

### 6.3. *Lack of security patterns*

Security patterns are an effective way to build secure applications Uzunov et al. (2015); however, almost no security patterns for microservices have been published; in fact, no article dedicated to them exists in the academic literature. These patterns would be specialized versions of more general security patterns but considering the specific environments of microservices applications.

### 6.4. *Secure microservice-based application development.*

The development of secure applications is very important from a practical point of view. While there are many works on how to secure specific parts of a system, there is little work on a systematic methodology to build secure applications. There is a development methodology for smart city clouds (D13 Krämer et al. (2019)), a methodology idea for critical applications (D17 Fetzer (2016)), an IoT framework (D23 Lu et al. (2017)), a Defense in Depth strategy (D34 Jander et al. (2019)), and a design method (A28). We need a general and complete development methodology for this purpose, with three variants, for IT, CPS, and IoT systems.

### 6.5. *Use of trusted hardware.*

A Trusted Platform Module (TPM) provides assured software execution by verifying that the hardware and software are legitimate and can be trusted before execution takes place. Trusted zones in hardware processors such as Intels SGX provide secure execution of kernels and storage of cryptographic keys. The use of SGX has been explored in (D2 Brenner et al., 2017) and (D26 da Silva et al., 2019), but we did not find any use of the TPM. This is an important direction for places that process sensitive documents.

[14] https://docs.microsoft.com/en-us/dotnet/architecture/microservices/secure-net-microservices-web-applications/.
[15] https://www.mulesoft.com/resources/api/microservices-security.
[16] https://www.mulesoft.com/resources/api/microservices-security.
[17] https://developer.okta.com/blog/2020/03/23/microservice-security-patterns.

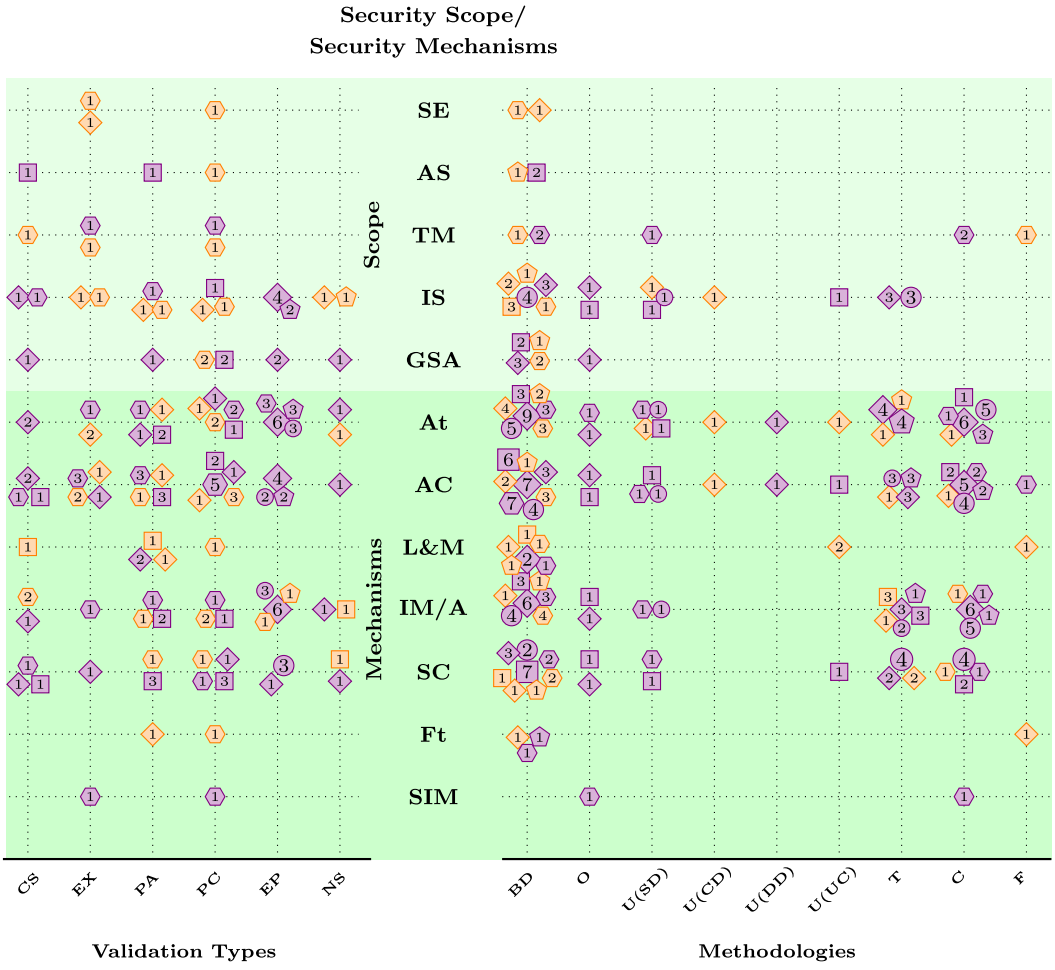



**Fig. 19 – The acronyms in the vertical axis are as follows:** At **(Authorization);** IM/A **(Identity Management/Authentication);** AC **(Access Control),** SC **(Secure Communication);** Ft **(Filtering);** L&M **(Logging & Monitoring);** SIM **(Security Information Management);** IS **(Implementation Security);** SE **(Security Evaluation);** TM **(Threat Modeling);** GSA **(General Security Architecture);** AS **(Application Security). The acronyms in the right X axis are as follows:** BD **(Block Diagram);** O **(Ontology);** U(SD) **(UML Sequence Diagram);** U(CD) UML Class Diagram; U(DD) **(UML deployment Diagram);** U(UC) **(UML Use Case Model);** T**; (Text Only);** C **(Code);** F **(Formal Analysis). The acronyms in the left X axis are as follows:** CS **(Case Study);** EX **(Experiment);** PA **(Performance Analysis);** PC **(Proof of Concept);** EP **(Example);** NS **(Not Specific). The geometric figures are:** Diamond **(Evaluation Research);** Pentagon **(Personal Experience);** Circle **(Opinion Paper);** Hexagon **(Proposal of Solution);** Square **(Validation Research). The colors are:** Violet **(Stop or Mitigate Attacks);** Orange **(Detect Attacks).**

## 7. Use cases (UC)

The results of our MLR should be useful to guide practitioners and researchers in the following possible use cases:

### *7.1. UC1: To publish new mechanisms or security analysis of microservice-based systems.*

When researchers want to write and publish their "new" mechanisms or security analysis, they can be aware of existing related work so they can avoid rehashing old research. Deficiencies or limitations of existing solutions can also lead to new ideas. Finally, our list of directions for research can indicate areas where there is no much work done that can be fruitful for new work. The methodologies used in the identified works can also support writing new research by suggesting appropriate abstraction levels, modeling approaches, domain specificity and quality attributes to be addressed. Validation is often a difficult problem for researchers, the enumeration of the validation approaches used by other researchers can provide a catalog of possibilities to validate new research.

### *7.2. UC2: To solve microservices security design problems.*

When practitioners and researchers want to solve design problems in microservices-based systems, our classification results can help them to compare existing mechanisms or analysis, and then select and reuse the appropriate one according to their objectives. Developers can utilize our classification scheme and results in the different life cycle development phases: 1) to consider the appropriate security constraints in the high-level system architectures of the analysis

**Table 6 – Use of these results.**

| Role | UC | Use of these results |
|---|---|---|
| Researcher | UC1 | Publish new research |
| | UC2 | Solve design problems |
| | UC3 | Search new ideas |
| | UC4 | Transition between areas |
| | UC5 | State of the art view |
| Practitioner | UC2 | Solve design problems |
| | UC3 | Search new ideas |
| | UC4 | Transition between areas |
| Student | UC3 | Search new ideas |
| | UC5 | State of the art view |

phase as well as the early architecting phase, 2) developers can first review existing designs to get guidance on how to proceed.

### 7.3. *UC3: To communicate and search new ideas.*

Our classification results can serve as a reference for the microservice engineering community, including practitioners and researchers. Our results can be extended by peers, providing the community with an important body of knowledge to guide future communications and research on this subject. Abstracting this knowledge in the form of a pattern catalog would provide a useful tool for future developers.

### 7.4. *UC4: To transition from software engineering to security or from security to software engineering.*

A software engineering expert (researcher or practitioner) wants to add security to her microservice applications; in this survey she can find knowledge about the needed security levels, security mechanisms, or security evaluation. A security expert wants to see how the security mechanisms he knows can be applied in the design of microservices applications.

### 7.5. *UC5: A view of the state of the art.*

Somebody preparing a presentation or publication of the current state of the art on microservices security would have a summary and repository of the relevant research.

Table 6 shows how different roles can use the results of this paper.

## 8. Threats to validity

Several threats to validity Wohlin et al. (2012) of this study had to be mitigated.

### 8.1. *Internal validity*

The threats of internal validity describe factors that could affect the results obtained from the study. To mitigate these threats, we created a strategy in which the related publications search was performed on defined keywords, and then followed with backward and forward snowballing processes on the selected studies, as described in Section 4.2.

To ensure that this review was repeatable, we defined the search terms and the inclusion/exclusion criteria carefully. To search the academic literature, we used recognized databases, in which the published articles went through a rigorous peer review process, which is a solid requirement for high-quality publications. For the search of grey literature, we used search engines, such as Google, and then applied rigorous inclusion/exclusion criteria approved by the authors after a deep brainstorming and voting with the intention of eliminating the possible personal bias that could arise, product of the judgment and experience of the researchers involved, and only the articles with high scores were selected for this study.

### 8.2. *External validity*

Threats to external validity are conditions that limit our ability to generalize our multivocal literature review. We include in our study academic and grey literature to get to know the state of the art in both literatures. The potential threat related to external validity is related to whether the selected studies describe or analyze security mechanisms or security scope. We covered this threat by selecting studies from both literatures that contained necessary information that covered the knowledge reported by researchers and professionals respectively. In the data analysis process, we performed validity tests of the extracted information, through a cross analysis process. Our main results are related to the security mechanisms and security scope reported in microservice-based systems, so it is not our objective to generalize to knowledge that is outside our object of study.

### 8.3. *Validity of the conclusions*

The threats to the conclusions validity are concerned with issues that affect the ability to draw the correct conclusions. In order to mitigate potential threats to the conclusions validity, we followed the best practices proposed by Petersen et al. (2008) and Wohlin et al. (2012). Also, we used the security tactics taxonomy by Bass et al. (1999) and Fernandez et al. (2015) to identify security strategies. Furthermore, to ensure the reliability of our study, we selected an acceptable size for primary studies in academic literature and grey literature, and the terminology was reviewed by all authors to avoid ambiguity. At least two of the authors reviewed all selected primary sources to mitigate bias in data extraction. Following the multivocal review approach allows us to replicate the study and similar future studies should yield similar results.

## 9. Conclusions

We conducted a multivocal literature review in order to get a perspective of what security ideas, including principles, analyses, mechanisms, and designs have been used to protect microservice-based systems, where we selected 36 primary studies in the academic literature and 34 grey literature

studies. The results of this study reveal that (i) the high frequency of articles on authentication, authorization, and access control is not surprising because they are the most basic security mechanisms and any secure system must have them; (ii) detecting attacks and stopping/mitigating attacks are the most frequent architectural security strategies used in microservice-based systems research; (iii) the majority of security mechanisms are validated through case studies and experiments; (iv) the most used research strategy is evaluation research; and (v) we found very few patterns for microservice-based systems security. Based on these results we recommended several directions for research. We believe that this is the first multivocal study of the security of microservice-based systems and should be of value to researchers and practitioners.

## Declaration of Competing Interest

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

## CRediT authorship contribution statement

**Anelis Pereira-Vale:** Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization. **Eduardo B. Fernandez:** Conceptualization, Formal analysis, Writing - original draft, Writing - review & editing, Visualization, Supervision. **Raúl Monge:** Conceptualization, Supervision. **Hernán Astudillo:** Conceptualization, Writing - original draft, Supervision. **Gastón Márquez:** Conceptualization, Writing - original draft.

## Acknowledgments

This work was supported by DPP scholarship, Universidad Técnica Federico Santa María (UTFSM), CONICYT (Chile) with grants PCHA/Doctorado Nacional/2016-21161005 and by PIA (Basal FB0821 CCTVal). The two reviewers provided useful constructive criticism that made this a significantly better paper.

## Appendix A. Grey Literature

**Table A1 – Grey Literature URLs.**

| ID | URL |
|---|---|
| [A1] | https://auth0.com/blog/an-introduction-to-microservices-part-2-API-gateway/ |
| [A2] | https://nordicapis.com/how-to-control-user-identity-within-microservices/ |
| [A3] | https://stormpath.com/blog/microservices-jwt-spring-boot |

(*continued on next page*)

**Table A1 (*continued*)**

| ID | URL |
|---|---|
| [A4] | https://www.darkreading.com/endpoint/rethinking-application-security-with-microservices-architectures/a/d-id/1325155 |
| [A5] | https://dzone.com/articles/data-centric-microservices-security |
| [A6] | https://dzone.com/articles/security-enforcement-of-the-microservices |
| [A7] | https://dzone.com/articles/securing-microservices-a-brief-look-at-different-m |
| [A8] | https://newizze.com/7-best-practices-of-microservices-security/ |
| [A9] | https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/694889/DWP-ss028-security-standard-microservices-architecture.pdf |
| [A10] | https://www.globallogic.com/wp-content/uploads/2017/08/Microservice-Architecture-API-Gateway-Considerations.pdf |
| [A11] | https://piotrminkowski.com/2017/02/22/microservices-security-with-oauth2/ |
| [A12] | https://www.leanix.net/en/blog/authorization-authentication-with-microservices |
| [A13] | https://dzone.com/articles/how-do-you-secure-microservices |
| [A14] | https://dzone.com/articles/advanced-microservices-security-with-spring-and-oa |
| [A15] | https://events19.linuxfoundation.org/wp-content/uploads/2017/12/Security-Approaches-for-Microservices-Architectures-Kameshwara-Rao-Marthy-Thomson-Reuters.pdf |
| [A16] | https://codeburst.io/i-believe-it-really-depends-on-your-environment-and-how-well-protected-the-different-pieces-are-7919bfa6bc86 |
| [A17] | https://www.microkubes.com/docs/security.html |
| [A18] | https://medium.com/tech-tajawal/microservice-authentication-and-authorization-solutions-e0e5e74b248a |
| [A19] | https://www.apriorit.com/dev-blog/558-microservice-container-security-best-practices |
| [A20] | https://www.networkworld.com/article/3299021/securing-microservice-environments-in-a-hostile-world.html |
| [A21] | https://dzone.com/articles/scale-security-while-innovating-microservices-fast |
| [A22] | https://dzone.com/articles/implement-secure-microservices-with-spring-security-and-oauth-20 |
| [A23] | https://dzone.com/articles/how-a-service-mesh-can-help-with-microservices-sec |
| [A24] | https://csrc.nist.gov/publications/detail/sp/800-204/final |
| [A25] | https://microservices.io/patterns/security/access-token.html |
| [A26] | https://www.sumologic.com/insight/microservices-architecture-security/ |
| [A27] | https://techbeacon.com/app-dev-testing/8-best-practices-microservices-app-sec |
| [A28] | https://developer.ibm.com/technologies/api/articles/securing-modern-api-and-microservices-apps-1/ |
| [A29] | https://www.youtube.com/watch?v=wpA0N7kHaDo |
| [A30] | https://hackernoon.com/service-to-service-authentication-for-microservice-apis-ccf4ab8073e6 |
| [A31] | https://dzone.com/articles/learn-how-to-secure-service-to-service-microservic |
| [A32] | https://dzone.com/articles/authentication-and-authorization-in-microservices |
| [A33] | https://dzone.com/articles/security-in-microservices |
| [A34] | https://istio.io/docs/concepts/security/ |

## REFERENCES


Abidi S, Essafi M, Guegan CG, Fakhri M, Witti H, Ghezala HHB. A web service security governance approach based on dedicated micro-services 2019;159:372–86. doi:10.1016/j.procs.2019.09.192.

de Aguiar Monteiro L, Almeida WHC, Hazin RR, de Lima AC, Silva SKGe, Ferraz FS. Survey on microservice architecture-security-trends in architecture, privacy and standardization on cloud computing environment. International Journal on Advances in Security 2018;11:201.

Ahmadvand M, Ibrahim A. Requirements Reconciliation for Scalable and Secure Microservice (De)composition. In: IEEE 24th International Requirements Engineering Conference Workshops (REW); 2016. p. 68–73.

Ahmadvand M, Pretschner A, Ball K, Eyring D. Integrity protection against insiders in microservice-based infrastructures: From threats to a security framework. In: Software Technologies: Applications and Foundations. Springer International Publishing; 2018. p. 573–88. doi:10.1007/978-3-030-04771-9_43.

Bánáti A, Kail E, Karóczkai K, Kozlovszky M. Authentication and Authorization Orchestrator for Microservice-based Software Architectures. In: 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE; 2018. p. 1180–4. doi:10.23919/MIPRO.2018.8400214.

Bass LJ, Clements PC, Kazman R. In: Addison-Wesley-Longman. Software Architecture in Practice, SEI Series in Software Engineering; 1999.

Brenner S, Hundt T, Mazzeo G, Kapitza R. Secure Cloud Micro Services Using Intel Sgx. In: Distributed Applications and Interoperable Systems. Cham: Springer International Publishing; 2017. p. 177–91. doi:10.1007/978-3-319-59665-5_13.

Calderón A, Ruiz M, O'Connor RV. A multivocal literature review on serious games for software process standards education. Computer Standards & Interfaces 2018;57:36–48. doi:10.1016/j.csi.2017.11.003.

Chen C. In: IEEE Cybersecurity Development (SecDev). With great abstraction comes great responsibility: Sealing the microservices attack surface; 2019. 144–144

Dong M, Ota K, Liu A. Preserving source-location privacy through redundant fog loop for wireless sensor networks. In: IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing; 2015. p. 1835–42. Doi:10.1109/CIT/IUCC/DASC/PICOM.2015.274

Dragoni N, Giallorenzo S, Lafuente AL, Mazzara M, Montesi F, Mustafin R, Safina L. Microservices: Yesterday, Today, and Tomorrow. Springer International Publishing; 2017. Doi:10.1007/978-3-319-67425-4_12

Esposito C, Castiglione A, Tudorica C, Pop F. Security and privacy for cloud-based data management in the health network service chain: a microservice approach. IEEE Commun. Mag. 2017;55(9):102–8. doi:10.1109/MCOM.2017.1700089.

Fernandez EB, Astudillo H, Pedraza-García G. Revisiting architectural tactics for security. European Conference on Software Architecture 2015:55–69. doi:10.1007/978-3-319-23727-5_5.

Fetzer C. Building critical applications using microservices. IEEE Security Privacy 2016;14(6):86–9. doi:10.1109/MSP.2016.129.

Garousi V, Felderer M, Hacaloğlu T. Software test maturity assessment and test process improvement: a multivocal literature review. Inf Softw Technol 2017;85:16–42. doi:10.1016/j.infsof.2017.01.001.

Garousi V, Felderer M, Mäntylä MV. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf Softw Technol 2019;106:101–21. doi:10.1016/j.infsof.2018.09.006.

Garousi V, Mäntylä MV. When and what to automate in software testing? a multivocal literature review. Inf Softw Technol 2016;76:92–117. doi:10.1016/j.infsof.2016.04.015.

Gerking C, Schubert D. Component-based Refinement and Verification of Information-flow Security Policies for Cyber-physical Microservice Architectures. International Conference on Software Architecture (ICSA). IEEE; 2019. p. 61–70.

Guija D, Siddiqui MS. Identity and Access Control for Micro-services Based 5G NFV Platforms. 13th International Conference on Availability, Reliability and Security, Association for Computing Machinery. ACM, 2018. doi:10.1145/3230833.3233255.

He X, Yang X. Authentication and authorization of end user in microservice architecture. J. Phys. Conf. Ser. 2017a;910:012060. doi:10.1088/1742-6596/910/1/012060.

He X, Yang X. Authentication and Authorization of End User in Microservice Architecture, Vol. 910. IOP Publishing; 2017b. Doi:10.1088/1742-6596/910/1/012060

Hinkley C, Snyder A, McAndrew T. In: Application Security Statistics Report. The evolution of the secure software lifecycle; 2018. https://info.whitehatsec.com/rs/675-YBI-674/images/WhiteHatStatsReport2018.pdf

Jander K, Braubach L, Pokahr A. Defense-in-depth and role authentication for microservice systems. Procedia Comput Sci 2018;130:456–63. doi:10.1016/j.procs.2018.04.047.

Jander K, Braubach L, Pokahr A. Practical defense-in-depth solution for microservice systems. Journal of Ubiquitous Systems and Pervasive Networks 2019;11(1):17–25. doi:10.5383/juspn.11.01.003.

Kitchenham B, Charters S. In: Tech. Rep. EBSE Technical Report EBSE-2007-01. Keele University. Guidelines for Performing Systematic Literature Reviews in Software Engineering; 2007.

Krämer M, Frese S, Kuijper A. Implementing secure applications in smart city clouds using microservices. Future Generation Computer Systems 2019;99:308–20. doi:10.1016/j.future.2019.04.042.

Zaheer Z, Chang H, Mukherjee S, der JV. Merwe, Eztrust: Network-independent Zero-trust Perimeterization for Microservices. In: Proceedings of the Symposium on SDN Research, SOSR '19. ACM; 2019. p. 49–61. doi:10.1145/3314148.3314349.

Hannousse A., Yahiouche S. . 2020. Securing microservices and microservice architectures: A systematic mapping study. CoRR abs/2003.07262. http://arxiv.org/abs/2003.07262 arXiv:2003.07262.

Li X, Chen Y, Lin Z. Towards Automated Inter-service Authorization for Microservice Applications. Proceedings of the SIGCOMM Conference Posters and Demos. ACM; 2019. p. 3–5.

Lu D, Huang D, Walenstein A, Medhi D. A secure microservice framework for iot. In: IEEE Symposium on Service-Oriented System Engineering (SOSE); 2017. p. 9–18. Doi:10.1109/SOSE.2017.27

M George V, Mahmoud QH. Claimsware: a claims-based middleware for securing iot services. IEEE 41st Annual Computer Software and Applications Conference (COMPSAC) 2017;1:649–54. doi:10.1109/COMPSAC.2017.85.

Myrbakken H, Colomo-Palacios R. Devsecops: A multivocal literature review. In: Software Process Improvement and Capability Determination. Cham: Springer International Publishing; 2017. p. 17–29. Doi:10.1007/978-3-319-67383-7_2

Nehme A, Jesus V, Mahbub K, Abdallah A. Securing microservices. IT Prof 2019a;21(1):42–9. doi:10.1109/MITP.2018.2876987.

Lewis J., Fowler M.. 2020. Microservices. A definition of this new architectural term.

https://martinfowler.com/articles/microservices.html online; accessed April 5th.
Nehme A, Jesus V, Mahbub K, Abdallah A. In: Fine-grained access control for microservices. Foundations and Practice of Security. Cham: Springer International Publishing; 2019. p. 285–300.
Nguyen Q, Baker OF. Applying spring security framework and OAuth2 to protect microservice architecture API. JSW 2019;14(6):257–64. doi:10.17706/jsw.14.6.257-264.
Otterstad C, Yarygina T. Low-level Exploitation Mitigation by Diverse Microservices. In: Service-Oriented and Cloud Computing. Cham: Springer International Publishing; 2017. p. 49–56. doi:10.1007/978-3-319-67262-5_4.
Pahl M, Aubet F. All Eyes on You: Distributed Multi-dimensional IoT Microservice Anomaly Detection. In: 14th International Conference on Network and Service Management (CNSM); 2018. p. 72–80.
Pahl M, Aubet F, Liebald S. Graph-based IoT Microservice Security. Network Operations and Management Symposium. IEEE/IFIP; 2018. p. 1–3.
Pahl M, Donini L. In: Network Operations and Management Symposium (NOMS). Securing IoT Microservices with Certificates. IEEE/IFIP; 2018 doi:10.1109/NOMS.2018.8406189.
Pardon G, Pautasso C, Zimmermann O. Consistent disaster recovery for microservices: the bac theorem. IEEE Cloud Comput. 2018;5:49–59. doi:10.1109/MCC.2018.011791714.
Pathania N. Setting Up Jenkins on Docker and Cloud. Apress; 2017. Doi:10.1007/978-1-4842-2913-2_4
Pereira-Vale A, Márquez G, Astudillo H, Fernandez EB. Security Mechanisms Used in Microservices-based Systems: A Systematic Mapping. In: XLV Latin American Computing Conference (CLEI). IEEE; 2019. p. 01–10. doi:10.1109/CLEI47609.2019.235060.
Petersen K, Feldt R, Mujtaba S, Mattsson M. Systematic mapping studies in software engineering. in: International Conference on Evaluation and Assessment in Software Engineering (EASE) 2008;8. doi:10.14236/ewic/EASE2008.8.
Preuveneers D, Joosen W. Access control with delegated authorization policy evaluation for data-driven microservice workflows. Future Internet 2017;9(4). doi:10.3390/fi9040058.
Preuveneers D, Joosen W. Towards Multi-party Policy-based Access Control in Federations of Cloud and Edge Microservices. European Symposium on Security and Privacy Workshops (EuroS PW). IEEE; 2019. p. 29–38.
Robinson J. Likert scale. Netherlands: Springer; 2014. doi:10.1007/978-94-007-0753-5_1654.
Salibindla J. Microservices api security. International Journal of Engineering Research & Technology (IJERT) 2018. doi:10.17577/IJERTV7IS010137.
Sill A. The design and architecture of microservices. IEEE Cloud Comput. 2016;3(5):76–80. doi:10.1109/MCC.2016.111.
M. S. L. da Silva and F. F. de Oliveira Silva and A. Brito. Squad: A secure, simple storage service for sgx-based microservices, in: 9th LatinAmerican Symposium on Dependable Computing (LADC), 2019, pp.1-9. doi:10.1109/LADC48089.2019.8995723.
Torkura KA, Sukmana MIH, Kayem AVDM. A Cyber Risk Based Moving Target Defense Mechanism for Microservice Architectures. Intl Conference on Parallel Distributed Processing with Applications, Ubiquitous Computing Communications, Big Data Cloud Computing, Social Computing Networking, Sustainable Computing Communications. IEEE; 2018. p. 932–9.
Torkura KA, Sukmana MI, Meinel C. Integrating continuous security assessments in microservices and cloud native applications. In: 10th International Conference on Utility and Cloud Computing, UCC '17. ACM; 2017. p. 171–80. Doi:10.1145/3147213.3147229
Sun Y, Nanda S, Jaeger T. Security-as-a-service for Microservices-based Cloud Applications. In: 7th International Conference on Cloud Computing Technology and Science (CloudCom). IEEE; 2015. p. 50–7. doi:10.1109/CloudCom.2015.93.
Uzunov AV, Fernandez EB, Falkner K. ASE: a comprehensive pattern-driven security methodology for distributed systems. Computer Standards & Interfaces 2015;41:112–37. doi:10.1016/j.csi.2015.02.011.
Wen Z, Lin T, Yang R, Ji S, Ranjan R, Romanovsky A, Lin C, Xu J. Ga-par: dependable microservice orchestration framework for geo-distributed clouds. IEEE Trans. Parallel Distrib. Syst. 2020;31(1):129–43. doi:10.1109/TPDS.2019.2929389.
Wieringa R, Maiden N, Mead N, Rolland C. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering 2006;11(1):102–7. doi:10.1007/s00766-005-0021-6.
Wohlin C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th International conference on evaluation and assessment in software engineering; 2014. p. 38. Doi:10.1145/2601248.2601268
Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A. Experimentation in software engineering. Springer Science & Business Media; 2012. Doi:10.1007/978-1-4615-4625-2
Xu R, Jin W, Kim D. Microservice security agent based on API gateway in edge computing. Sensors 2019a;19(22). doi:10.3390/s19224905.
Xu R, Ramachandran GS, Chen Y, Krishnamachari B. Blendsm-DDM: BLockchain-ENabled secure microservices for decentralized data marketplaces. In: International Smart Cities Conference ISC2, IEEE; 2019b. p. 14–17. Doi:10.1109/ISC246665.2019.9071766
Yarygina T, Bagge AH. Overcoming Security Challenges in Microservice Architectures. Symposium on Service-Oriented System Engineering (SOSE). IEEE; 2018. p. 11–20.
Yarygina T, Otterstad C. A game of microservices: automated intrusion response. In: Distributed Applications and Interoperable Systems. Springer International Publishing; 2018. p. 169–77. Doi:10.1007/978-3-319-93767-0_12
Yu D, Jin Y, Zhang Y, Zheng X. A survey on security issues in services communication of microservices-enabled fog applications. Concurrency Computation Practice and Experience 2018;31(22). doi:10.1002/cpe.4436.

**Anelis Pereira-Vale** obtained her Informatics Engineering degree from the University of Matanzas, Cuba, 2009. She is currently a Ph.D. student in Informatics Engineering at Universidad Técnica Federico Santa María (UTFSM), Chile. Her research focuses on secure methodologies for software development, security patterns, ontologies, microservice architectures, and IoT. Her thesis topic is about methodologies to build secure microservices.

**Eduardo B. Fernandez**(Eduardo Fernandez Buglioni) is a professor in the Department of Computer Science and Engineering of Florida Atlantic University. He has published numerous papers as well as several books on computer security and software architecture. He holds a degree in Electrical Engineering from Universidad Técnica Federico Santa María (UTFSM), Chile, an MS in EE from Purdue University, Lafayette, Indiana, USA, and a Ph.D. in Computer Science from UCLA, USA. He has published numerous papers on authorization models, object-oriented analysis and design, cloud computing, and security patterns. He has written four books on these subjects, the most recent being a book on security patterns. He is an active consultant for industry, including assignments with IBM, Allied Signal, Panasonic, Motorola, Lucent, Huawei, and others.

**Raúl Monge** has been a professor since 1994 at the Computer Science Department of Universidad Técnica Federico Santa María (UTFSM), Chile, where he previously graduated in 1981 as Electronic Engineer. In 1992 he obtained the degree of Dr.-Ing. from the Friedrich Universität Erlangen Nürnberg in Germany with a

specialization in distributed computing systems. He is interested in information security issues and secure software development.

**Hernán Astudillo** is Professor of Informatics at Universidad Técnica Federico Santa María (UTFSM). He is Informatics Engineer (UTFSM, 1988) and Ph.D. Information and Computer Science (Georgia Tech, 1995). After several years as lead/senior applications architect for consulting companies in USA and Chile, joined UTFSM in 2003. His main R&D interest is identification, recovery and reuse of architectural decisions and architectural knowledge (including tactics, patterns and trade-offs) for complex software systems, with a recent focus on microservices and Edge/Cloud systems. He is Principal Investigator of the Toeska R&D Team, which conducts teaching, research and technology transfer in software architecture, semantic software systems and software process improvement, and their application in e-governance and heritage computing. At UTFSM, he is responsible for software engineering academic activities; chairs the Doctoral Program in Informatics Engineering; and co-chairs the BPM Center. He is also a member of IFIP TC2 (Software Engineering), IEEE, ACM, SCCC, IASA and INCOSE.

**Gastón Márquez** is Ph.D. in Informatics Engineering at Universidad Técnica Federico Santa María (UTFSM), Chile. He is working in the research fields of architectural tactics, patterns, microservice architectures, technical debt, and security in Telehealth systems. He has published in several international conferences and has participated in international software architecture schools. He participated as a research visitor at the Rochester Institute of Technology (RIT), Rochester, New York, USA and the Université de Technologie de Compiègne (UTC), Compiègne, France. Before becoming a Ph.D. student, he worked in financial companies for five years.