The Journal of Systems and Software 213 (2024) 112055

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

An empirical investigation on the competences and roles of practitioners in Microservices-based Architectures[☆]

Hamdy Michael Ayas [a,*], Regina Hebig [b], Philipp Leitner [a]

[a] CSE Department, Chalmers | University of Gothenburg, Gothenburg, Sweden
[b] Institute of Computer Science, University of Rostock, Rostock, Germany

ARTICLE INFO

Keywords:
Microservices
Technical competences
Practitioners profiles
Developer roles
Stackoverflow

ABSTRACT

Microservices-based Architectures (MSAs) are gaining popularity since, among others, they enable rapid and independent delivery of software at scale, facilitating the delivery of business value. Additionally, there are attempts towards understanding practitioners' roles and technical knowledge. MSAs call for affinity in several technologies as well as business domains. This diversity makes it challenging to scope and describe the roles of practitioners. In addition, practitioners often do not receive training and contents of MSA training remain largely undefined, even though there are challenges in finding or developing relevant technical expertise. In this research, we determine the different technical roles that are required in MSAs, along with their detailed competences. We use public online forums (e.g., StackOverflow), where developers share technical knowledge. We analyze 13,517 public profiles of software engineers, deriving their technical competences. Our taxonomy of technical competences in MSAs, contains 11 competences clusters, organized in 3 collections of competences — Web Technologies, DevOps, and Data Technologies. In addition, we derive the roles of microservice practitioners and the characteristics of their roles. Our findings organize the technical competences of MSAs practitioners and determine the training topics and combination of topics that can prepare engineers for MSAs.

1. Introduction

Microservices-based Architectures (MSAs) are gaining popularity since, among others, they enable rapid and independent development and delivery of software features on a large scale, thus creating business value (Soldani et al., 2018). In addition, as software systems are increasingly taking a core role in companies' business value delivery, software engineers are becoming an invaluable asset to organizations. Hence, research and practice are turning towards (1) understanding what makes great software engineers in terms of technical knowledge (Liang et al., 2022) or personality (Smith et al., 2016) and (2) understanding the roles of software engineers in the development of software (Montandon et al., 2021a; Meesters et al., 2022). Adopting MSAs can fundamentally change how software is developed on a systemic level (Michael Ayas et al., 2023b), and requires working with different technical artifacts (Waseem et al., 2021; Michael Ayas et al., 2022). This new paradigm of software development predisposes a different mindset from engineers and teams, with different ways of thinking in developing software and making technical design decisions (Michael Ayas et al., 2021). In combination with the traction that transitions to microservices have, it is of value to start investigating the profiles of software engineers that specifically work with microservices and this study intends to serve this niche.

However, practitioners often do not receive professional training and contents of professional training on MSAs remains largely undefined. The tech stack of MSAs is quite dense and diverse, calling for highly skillful and capable engineers that have an affinity with a large variety of technologies for software infrastructure, development, and operation, as well as their respective business domains (Waseem et al., 2021). Even though the current state-of-the art covers technical and process related aspects of microservices, there is a lack of empirical investigations and understanding of the current workforce's characteristics and the technical roles that the workforce has. Current literature indicates both technical and organizational aspects that need empirical coverage (Jamshidi et al., 2018). Even though several studies cover the technical aspects, fewer studies investigate practitioners-centered aspects of developing software using MSAs. In addition, organizations are reported to be challenged in finding or developing relevant technical skills and expertise for MSAs Fritzsch et al. (2019), Baškarada et al. (2018). Consequently, it is challenging to scope and describe the

[☆] Editor: Kelly Blincoe.
[*] Corresponding author.
E-mail addresses: ayas@chalmers.se (H. Michael Ayas), regina.hebig@uni-rostock.de (R. Hebig), philipp.leitner@chalmers.se (P. Leitner).

https://doi.org/10.1016/j.jss.2024.112055
Received 5 May 2023; Received in revised form 25 March 2024; Accepted 3 April 2024
Available online 6 April 2024

roles of practitioners effectively, as well as the learning curve of microservices engineers which is reported to be high, time-intensive and costly (Buchgeher et al., 2017). Moreover, there is evidence that engineers often fall to anti-patterns and architectural technical debt (Cerny et al., 2023) especially since it is challenging to learn all the involved technologies effectively and in a systematic manner (Zhou et al., 2023; Soldani et al., 2018). Therefore, it is important to understand the technical profiles of engineers that work with MSAs.

The objective of this research is to provide a detailed view of the different technical roles that are required to develop and operate MSAs, along with their characteristics in the form of competences. To do so, we analyze public platforms where developers share technical knowledge, in order to derive the technical competences they possess. In addition, we organize the diverse technical expertise that appears in practitioners of MSAs. The aim is to organize technical competences, with the potential to determine contents of microservices training as well as understanding the needs of microservices teams in terms of personnel and required skills. Moreover, we aim to empirically understand the technical competences of practitioners working with MSAs that are confident enough on their competences and the community finds their discussions insightful. An increasingly valuable source for such an understanding is public platforms for social coding, like GitHub and online forums (e.g., StackOverflow), according to a number of studies (Marlow et al., 2013; Capiluppi and Izquierdo-Cortázar, 2013; Sarma et al., 2016; Singer, 2013). Especially for microservices, there is evidence that companies heavily rely some of their technical decisions (e.g., tool selection) on the availability of an active community in expert exchange sites (i.e., StackOverflow), tech blogs, and documentation (Buchgeher et al., 2017).

Consequently, to achieve the objectives of this study, we derive the technical competences and affinity of 21,189 practitioners that participate in StackOverflow discussions about microservices. We define technical competence as the basic ability or experience to understand, use, or develop the functioning mechanisms of a specific technical artifact or concept that we can derive from the semantics of 508 tags. Specifically, we use the tags that are associated with the answers of practitioners. Then, we cluster these tags based on how they appear together and organize them into collections of technical competences. Subsequently, we analyze in depth the associations of profiles with collections of competences to derive an understanding of practitioners working with MSAs. Concretely, we address the following research questions:

RQ1: What are the technical competences of practitioners involved in the Microservices community on StackOverflow?

RQ2: How are these technical competencies distributed across practitioners' profiles?

RQ3: What roles do practitioners working on microservices have?

RQ4: What are the main characteristics of the identified roles of microservice practitioners?

Answering the RQs lead us to a taxonomy for the technical competences that practitioners have in MSAs, the roles of practitioners, and the characteristics of these roles. We identify 3 core collections of competences in the taxonomy, *Web Technologies*, *DevOps* and *Data Technologies* competences, along with 3 stand-alone collections of competences. We then investigate how 13,517 profiles involved with microservices discussions associate with the identified collections of competences. Hence, we derive the 3 main roles in MSAs, namely *Web-based software engineers*, *DevOps engineers*, as well as *Data engineers*. Finally, we derive detailed competences profiles, based on the association of practitioners with 11 competences clusters.

This study makes the following main contributions:

- We provide an understanding of the technical competences and roles that are needed in MSAs.
- Our study is a first step towards describing the technical knowledge that engineers need to learn to be successful in the microservices domain.
- We identify and discuss combinations of competences that many engineers working with MSAs have, which can be used for example for future training and development purposes.
- We organize and determine the technical topics that can become part of microservices training, preparing engineers for working with MSAs.

2. Background and related work

Naturally, organizations and researchers make several attempts to understand what are suitable engineering profiles in terms of technical capabilities (Brooks, 1995), both for recruitment, training and talent retention. In addition, research is making attempts to understand what organizations need from engineers in recruitment (Montandon et al., 2021a). The technical profile of engineers is the combination of competences, abilities and skills that engineers have on a technical topic (Brooks, 1995). Even though there is a wave of published research on the different aspects of MSAs (Di Francesco et al., 2019; Hassan et al., 2020; Soldani et al., 2018), little research exists that investigates the profiles of engineers that work on such architectures.

2.1. Microservices based architectures

MSAs enable the fast development and delivery of software applications or new features, enriching the digital services that organizations provide and capturing the competitive advantage of launching new features rapidly (Jamshidi et al., 2018). Software engineers working in MSAs have varied and diverse concerns, from front-end development, to databases expertise, to deep knowledge of infrastructure and cloud technologies (Waseem et al., 2021; Soldani et al., 2018). Therefore, there is a plethora of studies investigating the individual technologies involved in MSAs (Hassan et al., 2020; Di Francesco et al., 2019). The identified multifaceted complexity of MSAs (Michael Ayas et al., 2021) adds to the already recognized importance of the quality and expertise of software engineering workforce for the success in software projects (DeMarco, 1999).

Montandon et al. (2021a) reports that there is great value in understanding in depth the characteristics and roles of software engineers in developing different technologies. Moreover, the value of reporting the perspective of practitioners in MSAs is also reported (Waseem et al., 2021). Consequently, it is of value to start identifying and understanding the characteristics and roles of software engineers specifically in developing microservices-based software systems.

2.2. Designing and developing microservices-based software

MSAs are a type of Service-Oriented Architectures (SOA) that focuses on having individual and independent design, development and maintenance of services (Dragoni et al., 2017). Two key unique properties of microservices in comparison to SOA is the Domain Driven Design and Development of services, utilization of REST communication protocols for the endpoints of services and deployment of pieces of software in lightweight containers (Newman, 2015; Zimmermann, 2017). In addition, microservices are characterized by low coupling that enables the rapid service delivery, and independence between services in different aspects like programming languages or datasourses (Newman, 2015). These properties enable many benefits, including allowing to scale applications to a greater extent than other types of software architectures (Dragoni et al., 2018).

Furthermore, MSAs newly introduce to practitioners different concerns, such as monitoring and testing in different levels of abstraction (Waseem et al., 2021; Michael Ayas et al., 2022) However, there is a gap in observing patterns on the specific topics that practitioners work with and inductively generating a theoretical understanding of the necessary competences for MSAs.

Current literature investigates the problems that practitioners come across on topics specific to MSAs, calling for more research towards that direction (Wu et al., 2022). There are different ways to develop a system based on MSAs, that can result to different challenges and advantages (Soldani et al., 2018). Microservices entail a variety of different technologies (Waseem et al., 2021) and often engineers have to make several decisions on how to tackle trade-offs between different concerns that they might face (Michael Ayas et al., 2021; Waseem et al., 2022). Current research indicates that organizational and cultural changes are among the key future challenges that need to be investigated (Jamshidi et al., 2018). Even though literature discusses extensively the development of MSAs based on the technological requirements of existing solutions, we lack understanding and empirical evidence of the current workforce's characteristics and the different technical roles that the workforce has.

2.3. Understanding the profiles and evolution of software engineers

Over time, a plethora of research has been trying to understand what characterizes the profiles of software engineers in terms of, for example, programming languages affinity (Miranda et al., 2022) and personality traits (Brooks, 1995). Software engineers contribute in many diverse ways, across different roles and thus, it is important to investigate in detail the skills that help them conduct their work (Liang et al., 2022).

Recent research investigates and derives different technical roles of practitioners, for example from their contributions in GitHub (Montandon et al., 2021b). In addition, it is possible to analyze the skills that organizations are looking for in engineers that are to be recruited (Montandon et al., 2021a). Furthermore, research investigates also how different technologies (e.g., programming languages) are related with each other in public networks (Miranda et al., 2022). Such research is enlightening in understanding software engineers and more related findings are required to start empirically drawing the landscape of software engineering skills and profiles. Furthermore, such research is beneficial not only on the broader scope of software engineering, but also on specific sub-topics of software technologies.

One of these sub-topics is microservices. Zimmermann (2017) calls out the increased cognitive load put on microservices architects as well as the increased effort of designing, testing, and maintaining an MSA. In addition, new fields that predispose a paradigm shift in development are re-defining the key roles and responsibilities of their engineers' roles and responsibilities (Kim et al., 2016). Similarly, MSAs are leading a wave of change in developers' mindsets, and thus, starting to define the specific roles and competences of engineers in them can be beneficial.

Recent work deriving the technical roles and competences of engineers by Montandon et al. (2021b) uses three different machine learning models on GitHub users. Miranda et al. (2022) use the Louvain method for StackOverflow profiles to make sense of existing technical communities. There is value in further applying the Louvain method on StackOverflow data. Specifically, to identify the communities that exist in StackOverflow profiles, from profiles that work on a focused subject, such as microservices.

Research contributions can make use of the knowledge gained from Miranda et al. and evolve towards not only comparing the Louvain method with other clustering techniques but also making use of the algorithm in order to extract a detailed view of how competences appear in microservices practitioners (which is part of this study's contribution). In addition, using different data specified to sub-domains of Software Engineering (in this case MSAs) can strengthen our understanding of practitioners as well as the validation of results from previous works.

2.4. Understanding the profiles and evolution of microservices practitioners

With microservices, engineers are often faced with a dense and diverse technology stack calling for affinity in a variety of technologies. MSAs are bringing changes to many technologies and infrastructure of software, such as testing (Michael Ayas et al., 2022), monitoring, and other supporting artifacts (Waseem et al., 2021). There is a gap in explicitly clarifying the competences that go beyond what is currently documented for programming in general (Miranda et al., 2022) and the most essential technical competences for microservices practitioners.

In addition, there is potential to build on earlier works indicating that the technologies introduced with microservices are by nature more multi-faceted and complex (Zimmermann, 2017), naturally requiring developers to wear many different "hats" as well. The diverse technology stack that engineers come across in MSAs makes it challenging to scope and describe the roles of practitioners effectively.

Hence, engineers fall into anti-patterns and technical debt due to the difficulty of learning all these technologies (Cerny et al., 2023). This makes it essential to evolve the works of analyzing and understanding practitioners' technical roles (Montandon et al., 2021b) with the derived competences taxonomy specializing on microservices. Research on the decision-making in MSAs (Michael Ayas et al., 2021; Waseem et al., 2022) or on anti-patterns of MSAs (Cerny et al., 2023) can benefit from the detailed understanding of main stakeholders' technical competences.

3. Methodology

We conduct our study by mining StackOverflow, using the StackExchange API. An overview of our methodology is presented in Fig. 1. During data gathering (Phase 1), we collect all available StackOverflow profiles that were related to posts about microservices. Specifically, we gather the tags of the profiles, since our analysis is based only on the associated tags of microservices-related profiles. We apply a set of inclusion criteria to gather only profiles that have confidence in the topics they discuss, as well as tags that appear multiple times in our dataset. In addition, we process the data by calculating the co-occurrence of tags and applying the Louvain method (Blondel et al., 2008; Lambiotte et al., 2008) for community detection. Thereafter, we perform manual analysis to define the identified competences clusters. Next, we calculate a custom association score between profiles and competences. Then, we analyze the association scores of profiles and we identify the primary roles of engineers. Finally, we provide the detailed characteristics of the identified roles of engineers working with MSAs. The replication package of the data gathering and analysis can be found in Michael Ayas et al. (2023a).

3.1. Data gathering

In this paper, we gather the public profiles of StackOverflow users that posted microservices-related posts in the Q&A forum. Specifically, we use the StackExchange API[1] to gather the data used in the study. Our study includes the tags associated with 21,189 different profiles of users. We gather the users along with all their associated tags (i.e. the tags that all of their posts had), since tags have a consistent presence and content. We include the tags that are both associated with microservices as well as other topics, since we want to derive the overall competences of engineers working with MSAs.

The posts that exist in StackOverflow discussing Microservices comprise the data that this study is based on, using the StackExchange API for data gathering. The data gathering source code and raw datasets can be found in the folder `source_code-raw_data` in the replication package (Michael Ayas et al., 2023a). Our data gathering makes use

[1] https://api.stackexchange.com/docs

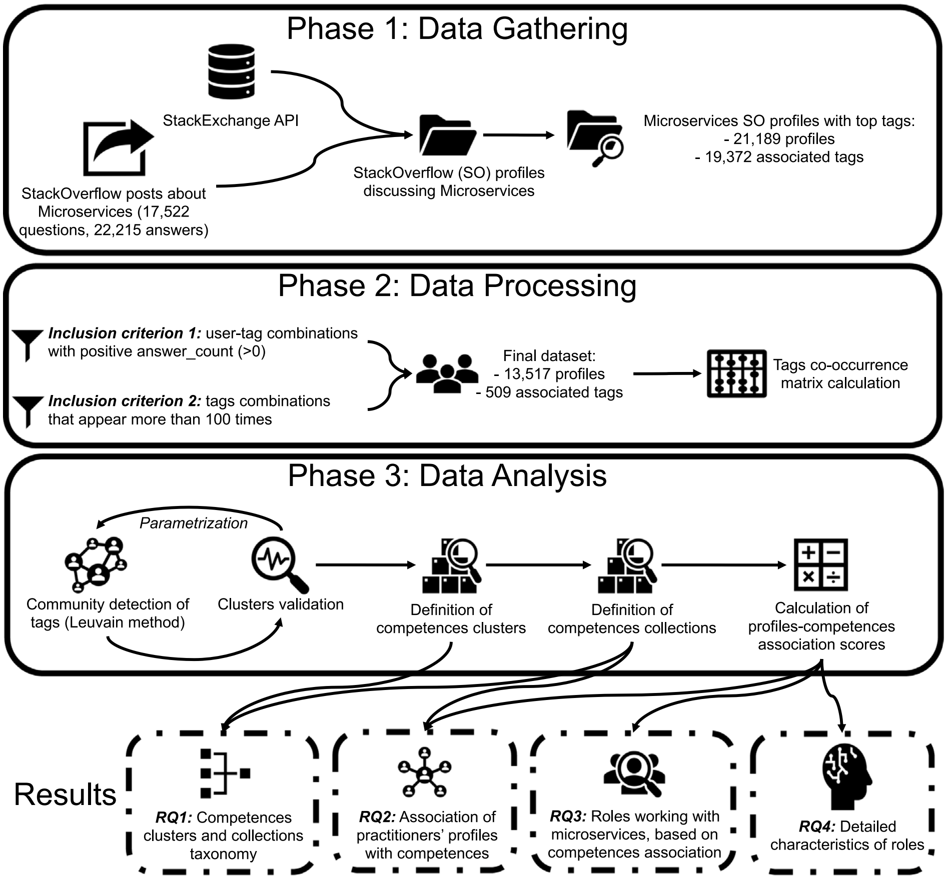

Phase 1: Data Gathering
StackExchange API
StackOverflow posts about Microservices (17,522 questions, 22,215 answers)
StackOverflow (SO) profiles discussing Microservices
Microservices SO profiles with top tags:
- 21,189 profiles
- 19,372 associated tags
Phase 2: Data Processing
Inclusion criterion 1: user-tag combinations with positive answer_count (>0)
Inclusion criterion 2: tags combinations that appear more than 100 times
Final dataset:
- 13,517 profiles
- 509 associated tags
Tags co-occurrence matrix calculation
Phase 3: Data Analysis
Parametrization
Community detection of tags (Leuvain method)
Clusters validation
Definition of competences clusters
Definition of competences collections
Calculation of profiles-competences association scores
Results
RQ1: Competences clusters and collections taxonomy
RQ2: Association of practitioners' profiles with competences
RQ3: Roles working with microservices, based on competences association
RQ4: Detailed characteristics of roles

Fig. 1. Overview of the research methodology.

of the raw data of microservices posts that are gathered in Wu et al. (2022), containing 17,522 questions and 22,215 answers related to MSAs. We use the posts dataset gathered by Wu et al., and enrich it by gathering the profiles of the users that authored these discussions on MSAs. An example of a user profile can be seen in Fig. 2. To gather the profiles, we developed a Python script that connects with the StackExchange API V2.3[2] and gathers for every post from the initial dataset, the author profiles of associated tags. The Python script accesses the csv file (`Wu_et_al_2021-dataset.csv` in the replication package) containing the raw data by Wu et al., to facilitate further data gathering of user profiles and associated tags. In our data gathering script, all the information of a given post from the dataset of Wu et al. are gathered using the `GET posts` function[3] and the tags are gathered using the `GET users tags`.[4] We anonymize the data of users' tags and store them in `users-toptags-fulldataset-21189.json` in the replication package (Michael Ayas et al., 2023a).

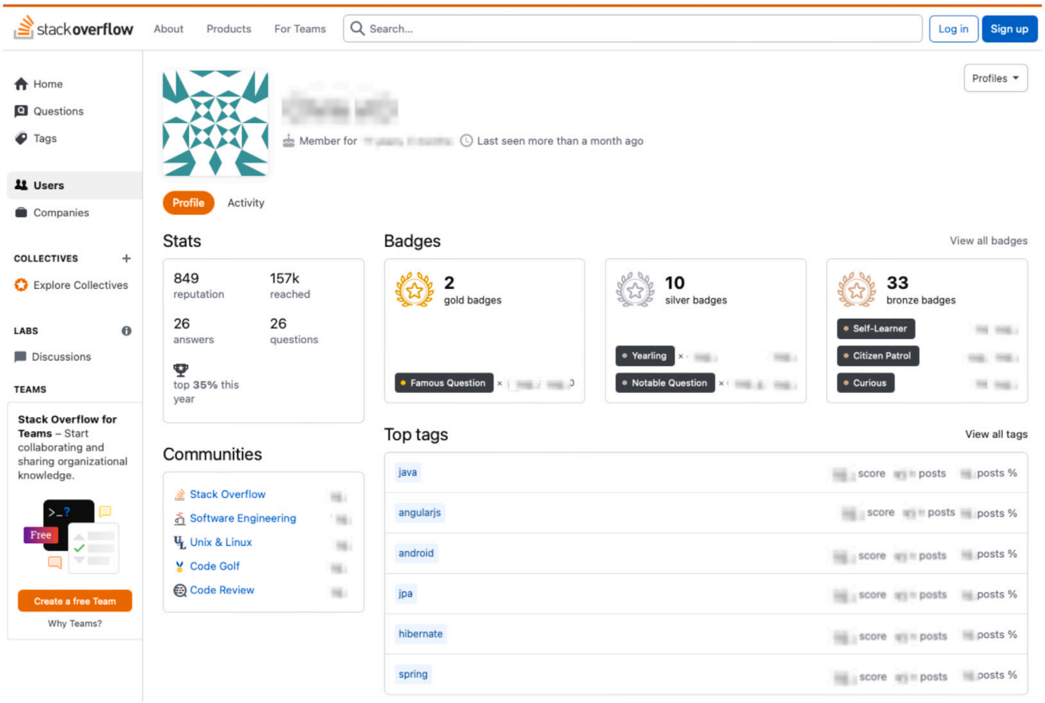
For each StackOverflow user associated with a post discussing microservices (one of the aforementioned posts), we fetch the complete set of tags that are associated with their profile, either through posting a question or posting an answer. Then, we filter the dataset to include only the set of top-tags for each profile, as it is defined by StackOverflow. The top tags are the 30 tags that have the highest positive tag score for the user, i.e., the up-votes of a tag's associated posts minus the down-votes of a tag's associated posts. This results in a total of 19,372 different tags (file tags.csv in the replication package (Michael Ayas et al., 2023a)) for the 21,189 different profiles. For each tag associated with a StackOverflow profile, we gather the answer count, answer score, question count, and question score.

3.2. Data processing

In this study, we first take into account the answer count, which is the amount of answers that a user gave, associated with a particular tag. Specifically, for each user, we included only the tags that had a positive (greater than zero) amount of answers, associated to a tag, filtering out the rest. That is, for each user, we only include the tags associated with an answer provided by this user. We reason that providing a public answer to a question about a topic (rather than simply asking a question) demonstrates that the user is somewhat confident in their expertise on this topic. Hence, we are able to capture not only the self-perception, but also the self-confidence related to a particular topic. This allows us to obtain an understanding of the technical competences on different topics that people have sufficient confidence in.

Subsequently, we filter out the tags that appear in less than 100 user profiles of the entire dataset. The reasoning for this decision is twofold. On the one hand, we make the assumption that tags appearing in very few user-profiles do not represent widely used technologies, tools and skills. Specifically, our rationale is that isolated concerns might be specific to cases of individual engineers, a particular version of a tool that raises individual issues, or case-specific challenges of tools that are not widely used. On the other hand, keeping all tags would pollute the analysis with too much noise. Hence, it was necessary to make a cut on the tags we select to analyze. We tested the next steps of the analysis with lower thresholds (i.e., 10 and 50 instead of 100), but the resulting users-tags matrix was *(1)* too sparse and

[2] https://api.stackexchange.com
[3] https://api.stackexchange.com/docs/posts-by-ids
[4] https://api.stackexchange.com/docs/tags-on-users



Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.

[redacted]

Member for [redacted] Last seen more than a month ago

Stats

849 reputation
157k reached
26 answers
26 questions
top 35% this year

Badges

2 gold badges
Famous Question × [redacted]

10 silver badges
Yearling × [redacted]
Notable Question × [redacted]

33 bronze badges
Self-Learner [redacted]
Citizen Patrol [redacted]
Curious [redacted]

Communities

Stack Overflow [redacted]
Software Engineering [redacted]
Unix & Linux [redacted]
Code Golf [redacted]
Code Review [redacted]

Top tags

java	[redacted] score	[redacted] posts	[redacted] posts %
angularjs	[redacted] score	[redacted] posts	[redacted] posts %
android	[redacted] score	[redacted] posts	[redacted] posts %
jpa	[redacted] score	[redacted] posts	[redacted] posts %
hibernate	[redacted] score	[redacted] posts	[redacted] posts %
spring	[redacted] score	[redacted] posts	[redacted] posts %

Fig. 2. Example profile with the associated top tags.

(2) computationally too demanding to analyze effectively. Furthermore, after analyzing a subset of the less aggressively filtered data, we observed that the results were not altered significantly and we did not have reasons to believe that significant improvements can be achieved by analyzing a larger dataset. For example, including more tags resulted in including different versions of used frameworks without altering the main findings (e.g., `google-cloud-endpoints-v2` co-exists with `google-cloud-endpoints` 94% of the time). Hence, to probe whether a less aggressive filtering would lead to a change in the outcomes, we bypassed the problem of the demanding computation by analyzing a subset of the data. Importantly, we do not claim completeness of the data in relation to every possible case of microservices practitioner, but rather we aim to provide a representative overview to some microservices practitioners, on the fundamental competences that microservices require. Consequently, even without a complete representation, our results are generalizable to a representative sample of microservices practitioners.

The filtering allows us to analyze only sufficiently prominent tags across the dataset. Then, we drop the (7672) profiles of users that remained without any tags after filtering. This filtering resulted in a considerable, yet not too dense, dataset of 13,517 StackOverflow user profiles, associated with a total of 508 tags, as shown in Table 1. The implementation of filtering took place in an automated manner, using a script that ensured the inclusion of only the user profiles with the tags that appear more than 100 times. The first author wrote the script for filtering and the second and third authors reviewed the script and made a quality check to make sure that the filtering code works as expected. In addition, all authors met in a set of discussion sessions where (among others) the filtering was discussed and finalized.

Table 1 provides an overview of the final dataset that is analyzed in this study. Besides the number of profiles and tags, some characteristics of the final filtered dataset are presented. The average and median number of answers per profile is shown to give an idea of how active the analyzed profiles were in terms of posting answers. In addition, the average and median scores of the answers per profile is a proxy of the quality of the answers that the profiles post, as perceived by other StackOverflow users. Moreover, tags per profile showcase how many tags are associated with profiles. Furthermore, the overview includes the average occurrence of each tag, across the 13,517 profiles and the number of profiles associated with only one tag. This information was derived from running and analyzing descriptive statistics on the final dataset.

Table 1
Overview of the final dataset showcasing the number of profiles and tags analyzed, with their characteristics. The average and median: (1) number of answers per profile, (2) scores of the answers per profile, and (3) tags per profile. Furthermore, the overview includes the average occurrence of each tag, across the 13,517 profiles and the number of profiles associated with only one tag.

Data attributes	*Values*
Profiles	13 517
Tags	508
Answers/profile (avg, median)	(296, 37)
Score/profile (avg, median)	(1130, 78)
Tags/profile (avg, median)	(21.6, 16)
Avg tags occurrences	379
Profiles with only one tag	360

Table 2
An example subset of the data gathered for user profiles and associated tags. The presented subset contains 5 user profiles out of 13,517 and 6 tags out of 508. User_ids are replaced with random numbers for demonstration purposes.

User_id	*Android*	*Ruby*	*Java*	*Apache-kafka*	*Thymeleaf*	...
0	74	0	13	0	0	...
1	0	1	0	0	0	...
2	1	0	54	17	6	...
3	2	0	40	0	0	...
4	0	0	9	0	0	...
...	...	...	...	...	...	...

An excerpt of the final dataset is presented in Table 2, where a subset of user profiles are shown. Each row is a user profile, each column it is an associated tag and the number of their relationship is the amount of answers that each user profile had on the specific tag. For example the user with the id 0 answered 74 questions tagged with ‘android’ and 13 questions tagged with ‘java’.

In the next data processing step, we analyze the dataset and extrapolate the tags co-occurrence matrix of our dataset. The tags co-occurrence matrix calculates the amount of times that every pair of tags appears in the same user. For example, users 0 and 3 from Table 2 increases by 1 the co-occurrence of the tags `android` and `java` and user 4 increases by 1 the co-occurrence of the tags `java` and `spring-boot`. Essentially, the resulting tags co-occurrence matrix is a symmetric table with all tags in rows and columns, showing how each tag connects to each other tag in our dataset, as illustrated in Table 3. A first observation in this step is that every tag co-occurred at least once with every other tag in our dataset.

Table 3
A subset of the tag co-occurrence matrix. The matrix is symmetric by construction, hence only the top half is shown. The full matrix is 508 × 508.

	Android	*Ruby*	*Java*	*Apache-kafka*	*Thymeleaf*	…
android		607	4174	624	237	
ruby			1230	189	50	
java				1593	544	
apache-kafka					93	
thymeleaf						
…						…

3.3. Data analysis

In the first step of the analysis, we create the full network of how tags connect to each other, based on the co-occurrence matrix. The outcome is a large and very dense, fully connected graph. Each node of the graph is a tag and there are connections among them for every time a tag occurred with another. The connections can be seen in the co-occurrence matrix (Table 3), showing how many times a pair of tags (source tag and target tag) occurs in the dataset.

Subsequently, we perform the Louvain method for community detection, as defined in the algorithm of Blondel et al. (2008) and the application approach of Lambiotte et al. (2008). We selected to use the Louvain method because we want to identify the topical communities from the tags associated to the analyzed StackOverflow user profiles. It is worth specifying that we base our analysis only on the associated tags of users.[5] Tags are predefined key words and phrases that are used to categorize and sort posts, as defined by StackOverflow.[6] In the case of our analysis, we investigate the tags that are associated with user profiles. Tags are structured because every profile has a finite amount of associated tags, from a sub-set of all tags that are predefined in StackOverflow.[7] The sub-set of tags is determined by our filtering. Therefore, we do not analyze any of the open text data that users posted (i.e., questions, answers or comments), neither any unstructured textual data that are provided to describe the profiles (i.e., about the user), but rather, only the connected tags. Since we want to investigate the technical competences of practitioners working with MSAs the information available in the tags is sufficient, scrutinized by the StackOverflow community and structured, enabling the efficient analysis of many user profiles. Tags are inherently structured in nature and therefore, representing them in a graph-based data structure, using the Louvain method is preferable in contrast to, for instance, Latent Dirichlet allocation (LDA), or other clustering algorithms.

With the Louvain algorithm, we modularize the network based on automatically identified communities. The first author run the algorithm multiple times with different parameters. Then, the first and third authors manually evaluated the resulting clusters, altering the parameters until a sensible clustering was reached. Specifically, we held a meeting and skimmed through every cluster, checking that it had enough tags to contain a coherent topical area. In addition, the first and third authors made the observation that when the clusters contained more than 100 tags, many tags in the same cluster were unrelated to each other content-wise, making it impossible to derive the topical area of the cluster. Therefore, the criteria for evaluating the clusters was that the clusters should have more than 10 tags and less than 100 tags belonging to them. In this way, we avoid very fine-grained clusters that are too specific, as well as very coarse-grained clusters that are too generic. Consequently, we selected the run of the algorithm with the most clusters adhering to the criteria of having 10 to 100 tags. The outcome set of parameters results in 15 clusters, as shown in Table 4. The four clusters that fail to meet these criteria are excluded from the subsequent analysis steps, meaning that we selected approximately 75% of the resulting clusters.

Table 4
The identified clusters with the number of tags in each and if they are included or excluded.

Cluster	*Size (tags #)*	*Included*	*Rationale*
1	87	✓	
2	12	✓	
3	6	X	Fewer than 10, ruby-specific tags
4	31	✓	
5	3	X	Fewer than 10, generic tags
6	19	✓	
7	24	✓	
8	61	✓	
9	3	X	Fewer than 10 tags, no clear scope
10	93	✓	
11	43	✓	
12	4	X	Fewer than 10 tags, no clear scope
13	78	✓	
14	14	✓	
15	31	✓	

Different parameters were tried for how the tags grouped together. In every run of the algorithm, we check manually the tags that cluster together, to see how cohesive each cluster is. Hence, we ensure that with the same parameters, we get the same results (which was the case for the final choice of parameters). The first parameter used in this analysis step is that the algorithm was set to start by randomly picking nodes and tag connections for identifying communities. This ensured that our communities were not dependent on the order of the dataset. Next, we set the algorithm to consider the weight of the connections between tags (i.e., how many times a pairing of tags occurs). The resolution of the algorithm that gave clusters as an output with 10 to 100 tags is 0.8.

The output of each run was checked manually by all authors. Specifically, our previous knowledge and expertise on the topics helped us determine how cohesive are the topics of the tags within a cluster. We held meetings in which all authors discussed the clusters of different runs and the final clustering was determined after reaching consensus among all authors. Furthermore, in these thematic analysis meetings, all authors discussed the collections of tags within each cluster and gave titles to the clusters, based on the context that the containing tags revealed. For example, a cluster that had tags like `css` and `browser`, was given the name of *Front-end development* by the authors. Therefore, the algorithm helped to cluster together related tags and the authors defined the themes of the clustered tags, based on their context.

Next, we aggregate the sum of associations between profiles and identified clusters. Specifically, for each profile, we count how many different associations they had with each cluster, by aggregating the number of tags that belong to each cluster. We do this association by iterating over the dataset of profiles with their associated tags (Table 2). Then, we calculate each profile's association score with each cluster, using Eq. (1).

$$\alpha = \frac{\sum \gamma \times 100}{\sum \beta} \tag{1}$$

In Eq. (1), α is the association score, γ is the associated tags of an identified cluster and β is all the associated tags of the profile. The

[5] https://stackoverflow.com/help/interesting-topics
[6] https://stackoverflow.com/help/tagging
[7] https://stackoverflow.com/tags

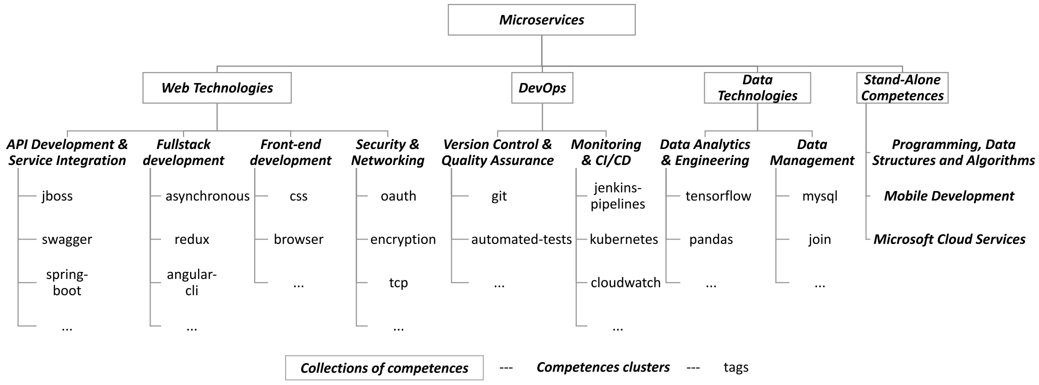


Fig. 3. Derived taxonomy of microservices competences organized into collections of competences, competences clusters, and tags that describe them.

association score is basically the percentage of associated tags with each cluster. Hence, we first count the number of tags per cluster that a profile is associated with. Then, we consider the total number of user-cluster associations as 100% and we calculate the percentage of every association pairing between a user and a cluster. This allows us to recognize the primary cluster of each user and the scores in all the other clusters. In addition, we stratify the data over the clusters, making sense of them at a higher level of abstraction.

4. Results

The analysis of this study leads to a taxonomy of competences in MSAs, the roles that practitioners have in MSAs as well as characteristics of the identified roles. In its lowest level of abstraction, the identified taxonomy of competences starts from tags that are associated with StackOverflow profiles that answered microservices posts. The tags are clustered into 11 identified technical competences clusters (from the algorithm). The contextual themes of the competences clusters are determined by the authors. We further group the clusters into collections of competences, describing the overarching themes that exist in the microservices-related profiles. We derive the roles of practitioners that work with microservices based on the competences collections, taking into account their primary competences as well as their complementary competences. Finally, we identify and describe different trends that exist in the identified roles, in terms of specific technical competences.

4.1. RQ1: Technical competences clusters

A technical competences cluster is the identified group of competences, from the inter-connected tags that are associated with user profiles. From the resulting network of how tags connect to each other, we derive 11 competences clusters. These competences clusters are the clusters we identified using the Louvain method, excluding four clusters as indicated in Table 4. Furthermore, we assigned clusters that are semantically interconnected with each other to distinct collections of competences, resulting in a taxonomy of competences in MSAs presented in Fig. 3. In the first collection of the taxonomy, four competences clusters are closely related to *Web Technologies*. In the second collection, two competences clusters are about *DevOps*. In addition, we find a third collection that is about *Data Technologies*. Finally, we have three clusters that are stand-alone, entailing competences that are either foundational or orthogonal to microservices development.

4.1.1. Web technologies competences collection

The first collection of competences clusters brings together competences clusters that have the common theme of *Web Technologies*. The theme of web technologies is indicating the predominant cloud-native nature of microservices. Microservice systems commonly make use of modern web technologies, such as APIs and asynchronous communication structures.

API development and service integration is the competences cluster that covers technical topics that are related to the development of the backbone of a MSA. Specifically, the topics that the tags of this competences cluster indicate, are about the design of core APIs and service integration structures. For example, `swagger`, `jboss` and `spring-boot` are among the tags that exist in this competences cluster, indicating the development of APIs in the backend. In addition, we observe that the cluster includes other aspects of API development, for example service integration, integration testing and authenticating access to APIs specifically (e.g., `spring-security-oauth2`). Service integration is indicated by frameworks mainly in the runtime of services (e.g., `jhipster`, or `kafka-consumer-api`, or `log4j`). Consequently, this competences cluster is about understanding, skills, and abilities related to service integration.

Moreover, a list of technical abilities to work with this competences cluster is presented in Table 5. The list is based on the tags that belong to the API development and service integration competences cluster.

Table 5
Examples of technical abilities to work in API development and service integration.

Ability to work with	Example tags
Swagger	swagger-ui, swagger
Spring	spring-mvc, spring-boot-actuator, spring-batch
Apache Kafka	apache-kafka, kafka-consumer-api

Fullstack Development is a competences cluster closely related to the previous one, and groups together understanding and skills about fullstack web development. However, whereas the "API Development and Service Integration" cluster focuses on integration, this competences cluster contains skills related to the actual development of services. Considering the cloud-based nature of MSAs, it is natural to have a cluster for the web technologies that are used. On the one hand, such technologies are about the design of the system's logic through managing cache, routing application flow and managing the stateless nature of microservices (e.g., `asynchronous` designs). On the other hand, such technologies include specific implementation frameworks and designs, identified through tags like `redux`, `socket.io`, and `angular-cli`. Table 6 presents the list of example technical abilities that are identified from tags belonging to the Fullstack Development competences cluster.

Table 6
Example technical abilities identified for Fullstack Development.

Ability to work with	Example tags
JS frameworks	node.js, express, nest.js, angular, react, vuejs2
Distributed comm.	callback, sequelize.js, socket.io, promise, axios
Database programming	mongodb, firebase, mongoose

Front-end Development is the competences cluster grouping together tags that direct to technical skills and abilities in delivering front-ends of web applications. For example, we observe tags that are about developing and delivering the frontend of web apps (e.g., browser, wordpress, redirec, css etc.). In addition, there is indication of competences about web applications development frameworks such as laravel and bootstrap. The emergence of this cluster from our automated analysis indicates that front-end development is a significant aspect in service development, and requires skills that are different to the competences contained in the other three clusters in this collection. Examples of these technical abilities are aggregated and presented in Table 7.

Table 7
Example technical abilities identified for Front-end Development.

Ability to work with	Example tags
Web design	css, html, twitter-bootstrap, checkbox, button
Browsers	session, redirect, google-chrome, firefox
Front-end Frameworks	jquery

Security and Networking competences cluster includes tags that indicate understanding, skills and ability in cybersecurity and network communication topics. On the one hand, cybersecurity-related competences are indicated through tags about authentication and authorization (e.g., oauth, single-sign-on). In addition, this competences cluster includes designs such as single-sign-on and specific frameworks such as jwt. Furthermore, there are advanced concepts such as encryption mechanisms. Moreover, networking competences in this cluster are indicated by tags including tcp, http-headers, and websocket, among others. Finally, an aggregated list of example technical abilities to work with Security and Networking competences cluster can be seen in Table 8.

Table 8
The technical abilities identified for Security and Networking.

Ability to work with	Example tags
Authentication	oauth-2.0, encryption, single-sign-on, cookies
Network communication	tcp, http, websocket, request, server

4.1.2. DevOps competences clusters

The second collection of competences is mainly concerned with the *DevOps* competences of engineers developing microservices. This competences collection indicates the needed competences to setup and configure the required infrastructure for seamlessly operating microservice-based systems.

Version Control and Quality Assurance cluster groups together competences about essential artifacts of MSAs, such as the general version control and testing architecture of systems. Specifically, competences that are grouped together include (1) testing, (2) testing stages in the CI pipeline, and (3) version control competences. Testing is represented through tags that are associated with artifacts and concepts such as mocking, debugging, unit-testing, and error handling. Moreover, this competences cluster presents a clear connection of testing with testing stages in the CI pipeline, through tags such as automation, automated-tests, command-line and build. Finally, competences related to version control complement the required knowledge with tags such as git, repository, and merge. This collection of competences showcases the diverse and interconnected understanding, abilities, and skills that are required to ensure the quality of microservices. The technical abilities based on the tags of this competences cluster are also illustrated in Table 9.

Table 9
Example technical abilities identified for Version Control and Quality Assurance.

Ability to work with	Example tags
Testing	mocking, debugging, unit-testing, error-handling
CI pipelines	automation, automated-testing, command-line, build
Version control	git, repository, merge

Monitoring and CI/CD is the cluster grouping together competences that are about the technical infrastructure needed for developing MSAs. These competences compose the DevOps skills and experiences that are needed to maintain and operate an MSA. Specifically, there are competences indicated by tags about configuring the continuous cloud integration, deployment, monitoring, and managing the system. Specifically, CI-related tags include jenkins, contiuous--integration, and jenkins--pipelines. Moreover, CD-related topics that we identify are about containerization (e.g., docker or istio) and orchestration (e.g., kubernetes-related tags such as kubectl, yaml, or kubernetes-helm). Importantly, this cluster also entails tags about monitoring tools such as cloudwatch. Finally, this competences cluster contains tags about cloud architecture topics, including load balancing and other artifacts essential for service delivery. A list of technical abilities is composed in Table 10, based on tags of this specific competences cluster.

Table 10
The technical technical abilities identified for Monitoring and CI/CD.

Ability to work with	Example tags
Continuous integration	jenkins, gitlab-ci, yaml, cron, scripting
Deployment	kubernetes, docker, docker-swarm, serverless
Monitoring	prometheus, cloudwatch

4.1.3. Data technologies competences clusters

The third collection of competences clusters is about *Data Technologies* competences of microservices practitioners. Specifically, data technologies are about the competences of practitioners in processing, analyzing, storing, and managing the data of their services and software applications.

Data analytics and data engineering brings together competences that are heavily related to the data science, data analytics, and data engineering nature of microservice developers' tasks. The competences cluster includes on one hand tags about machine-learning. On the other hand, the cluster contains specific implementation competences for frameworks such astensorflow. In addition, many of the included tags are related to data processing skills, such as pandas, dataframes, and data manipulation. Last but not least, there is a prominence of tags related to integrating the produced artifacts in the entire system or scaling the analysis capabilities (e.g., apache-spark or hadoop). Table 11 collects example technical abilities derived from the tags of this competences cluster.

Table 11
Example technical abilities identified for data analytics and data engineering.

Ability to work with	Example tags
Machine learning	machine-learning, tensorflow
Data processing	dataframe, numpy, pandas, python, sqlalchemy
Data analytics	Hadoop, apache-spark

Data Management is the competences cluster grouping together tags that are about databases. Such competences include tags about specific database technologies (e.g., mysql), database design skills, and database querying. Data management topics are expected to appear in profiles discussing microservices, since such topics are often essential parts of cloud-based software systems. Interestingly, join and indexing are among the more central tags of this cluster, indicating that joins and indexes are among the more intricate and popular data management tasks that microservice practitioners face. The abilities to work with data management are presented in Table 12.

Table 12
Example technical abilities identified for data management.

Ability to work with	Example tags
SQL databases	mysql, postgresql, mariadb, database-design
Querying	join, indexing, select, group-by

4.1.4. Stand-alone competences clusters

Finally, we collect together the competences clusters that are standalone and that do not relate distinctly with other competences clusters.

These clusters are more generic in the sense that they are generally not specific to microservices development. We interpret the cluster of *Programming, data structures and algorithms* as the basic competences that practitioners working in advanced architectures such as MSAs have. The competences cluster of *Mobile development* contains competences that are specifically about mobile development, which is an orthogonal set of skills to the Web Technologies collection, but not inherently about microservices. Finally, the cluster on *Microsoft Cloud Services* contains competences that can, in principle, relate to all the other clusters, but are grouped together by our automated analysis due to specifically being about Microsoft technology.

Programming, data structures and algorithms is the cluster that has to do with the programming skills and experience that is typically present in designing and developing MSAs. Specifically, this cluster includes tags that are generally about programming (e.g., `if-statements`, `perl`). In addition, in this cluster, there is a variety of competences about data structures, through tags such as `hash tables`. Finally, this cluster also has competences about `algorithms`. All these skills are fundamental and it is interesting to see their prominent position in clusters of competences about MSAs. Examplee tags found in this competences cluster are aggregated in Table 13.

Table 13
Example technical abilities identified for programming, data structures and algorithms.

Ability to work with	Example tags
Programming	if-statements, for-loop, parsing, design-patterns
Data structures	hash-tables, static, arraylist
Algorithms	sorting, recursion, parallel-processing

Mobile development is the competences cluster indicating that software development in mobile applications is a distinct collection of competences, found in MSAs. As expected, this collection of competences includes fundamental artifacts of mobile development, such as `android`, `ios`, and `objective-c`. In addition, there are tags about more specific aspects of mobile development, such as lightweight databases (`sqlite`), `user-interface` aspects, and usage of external services such as `google-maps`. Therefore, there is a set of topics about mobile development that are relevant to designing and engineering MSAs. technical abilities to work in mobile development are presented in Table 14.

Table 14
Example technical abilities identified for mobile development.

Ability to work with	Example tags
Apps development	android, ios, objective-c, swift
Tools and services	sqlite, google-maps, android-studio

Microsoft Cloud Services includes tags about any aspect of the end-to-end design of systems using MSAs, but using a specific cloud provider. Specifically, the tags are about tools and technical frameworks that a large cloud provider offers. The cluster includes tags for the full stack of technologies that are used in microservices. Competences that are about `azure` link to tags such as `.net`, `visual studio`, `c#` and their related tags. Furthermore, we observe tags about all aspects of the system such as development tools (e.g., `vs-code`) and architecture designs and frameworks that are prominent in the cloud providers implementations, such as `asp.net-mvc`, `mvvm` and `domain-driven-design`. It is worth noting that this competences cluster contains tags that could in principle fit in any of the other clusters. The emergence of this cluster from our automated analysis mostly indicates that Microsoft technologies, such as `azure` or `.net`, are frequently used together with other Microsoft technologies (more frequently than is the case for open source technologies or technologies from other vendors), and the related users in our dataset predominantly were less “Web Technologies” or “DevOps” experts than “Microsoft” experts. Technical abilities to work with Microsoft’s cloud services are depicted in Table 15.

Table 15
Example technical abilities identified for Microsoft’s cloud services.

Ability to work with	Example tags
Architecture	asp.net-mvc, mvvm
Tools and services	.net, visual-studio, nuget, excel

Answering RQ1:
The technical competences of practitioners involved in the Microservices community on StackOverflow are described in the competences taxonomy, developed based on the tags associated to practitioners involved with MSA topics. The taxonomy has eight competences clusters, organized in three overarching collections of competences clusters and three stand-alone competences clusters. We identified the following core competences clusters: (1) APIs and service integration, (2) Fullstack Development, (3) Front-end Development, (4) Security and Networking, (5) Version Control and Quality Assurance, (6)Monitoring and CI/CD, (7) Data Analytics and Engineering, and (8) Data Management. Additionally, we identify three stand-alone competences clusters that are orthogonal to the core competences clusters: (1) Programming, Data Structures, and Algorithms, (2) Mobile Development, and (3) Microsoft Cloud Services.

4.2. RQ2: Technical competences in practitioners profiles

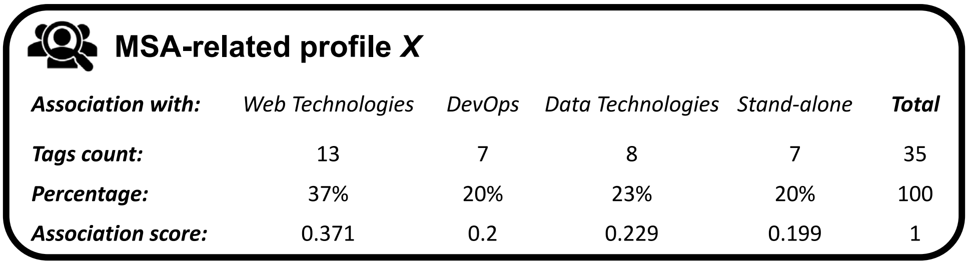
To investigate how competences appear in the profiles of practitioners, we first calculate how each profile is associated with the main collections of competences clusters we identified in RQ1 (Web Technologies, DevOps, and Data Technologies). The associations between competences collections and MSA-related profiles (StackOverflow profiles related to MSAs) is based on the association score of the profiles with the clusters. The association score of a competences cluster with a profile is effectively the percentage of the tags related to the specific competences cluster, from the aggregated amount of tags that a MSA-related profile associate with. Association scores for each profile (user) add up to 1 by construction. An example of the association scores of an MSA-related profile is shown in Fig. 4.

4.2.1. Association score distribution

Fig. 5 showcases the association scores distribution of the three main competences collections.

The *Web Technologies* competences cluster is forming a distribution with relatively low density (not exceeding 0,6) across the different association scores of associated profiles. Specifically, the amount of profiles that associate with Web Technologies with 20% is similar to those that associate with 70% and everything in between. Therefore, the standard deviation from the mean is high, indicating that there is no common pattern of how practitioners working with Web Technologies associate with the related competences. The diversity of association scores in Web Technologies, showcases the different levels of specialization that are possible from engineers working with this collection of competences. Web Technologies seem to have evolved over time and becoming an important aspect of how software is getting developed. Furthermore, almost all profiles that are associated with the *Web Technologies* competences collection score between 20% and 80%, meaning that it is equally possible to have practitioners with some Web Technologies-related concerns with being solely concerned in their work with the specific collection of competences. In addition, it is worth noting that there is a large number of profiles that are associated with the *Web Technologies* competences collection at a rate of 100%. Hence, they are associated only with tags that belong to the specific cluster.

Contrary to Web Technologies, the *DevOps* competences cluster is forming a distribution with a lower standard deviation from the mean,



MSA-related profile *X*

Association with:	*Web Technologies*	*DevOps*	*Data Technologies*	*Stand-alone*	***Total***
Tags count:	13	7	8	7	35
Percentage:	37%	20%	23%	20%	100
Association score:	0.371	0.2	0.229	0.199	1

Fig. 4. An example of the association scores of an MSA-related profile (StackOveflow profile related to MSAs).

indicating a clearer pattern on how practitioners associate with DevOps competences. Specifically, the profiles associated with the *DevOps* competences collection have a high density in scores between 10% and 30%. Hence, DevOps competences appear in profiles that also have other competences. This could be due to the fact that many practitioners have to perform DevOps tasks to accommodate their own needs. The association distribution of profiles with the *DevOps* competences is denser in lower percentages than the *Web Technologies* competences collection and higher percentages than the *Data Technologies* competences collection. In addition, the *DevOps* competences collection is associated with a lower number of profiles at 100%. This means that fewer practitioners are solely concerned with DevOps-related competences. In fact, more than 9000 profiles have some association with DevOps and thus, at least one DevOps competence appears in most microservices-related profiles in some shape or form.

Finally, the *Data Technologies* competences collection presents a unique pattern, in comparison to the rest of the competences clusters. Specifically, we observe that a large number of profiles (and at a high density) are associated with the specific competences collection with lower scores. This means that MSA-related profiles are rarely associated with a high score with the *Data Technologies* competences collection. Practitioners often require limited Data Technologies competences to conduct their work, leading to a low association score. The often low association score (i.e., many profiles have few tags about Data Technologies) indicates that Data Technologies competences are more likely to be complemented with another collection of competences, than the other collections of competences.

4.2.2. *Profiles associating with only one collection of competences*

As shown in Fig. 5, there is a peak in the density of profiles that associate at a 100% with the three main collections of competences. This high level of specialization could be an artifact of different reasons. Thus, we give an overview of them, by aggregating the profiles that belong to those peaks and extrapolate what those profiles are.

Specifically, we identify that 2061 profiles are exclusively associated (associate at a 100%) with one collection of competences. From those profiles, 796 are associated with the collection through five answers or more. Hence, these profiles belong to practitioners that have confidence in answering to discussions related to each other, under a specific competences collection. These profiles have focused concerns and thus, they can actually be specialists in their respective collection, since they give multiple answers.

On the other hand, 360 profiles associate only with one tag, 394 with two tags, and 309 with three tags. These could be occasional users

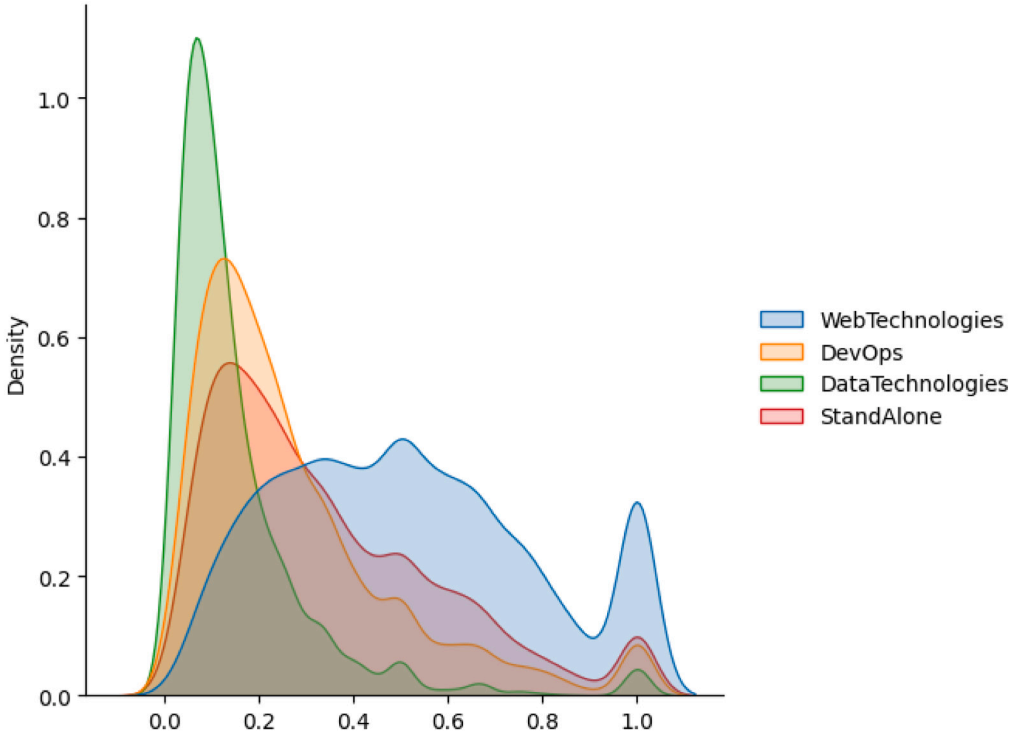


Fig. 5. The association distribution of profiles with competences collections. The X -axis is the associated percentage of profiles with competences collections and the Y -axis is the density, based on the amount of associated profiles. It is worth noting that the peak in association score around 1.0 is not the artifact of profiles with a very low number of tags, since only 1063 profiles have less than 3 tags. Instead, there are a substantial number of profiles in our dataset that are strongly focused on a specific topic.

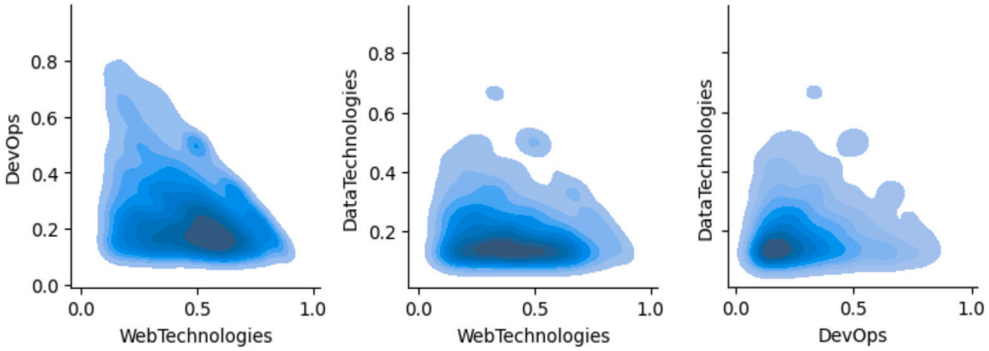


Fig. 6. The relationships between all groups of competences clusters at the range of association scores (in X and Y axes).

that only discussed about specific topics a few times. For these profiles, we cannot draw any conclusions about their specialty, since we only know that they gave few answers. Nevertheless, we include them in our analysis, since these occasionally active profiles account for a small percentage (≈10%) in comparison to the entire dataset.

In Section 4.3 we give a more detailed analysis of specialists, discussing the profiles with an association score higher than 50% in one particular collection of competences.

Moreover, we derive the relationships among the different competences collections, in terms of the association score of different profiles with each collection. Fig. 6 showcases how the competences collections relate to each other. It is worth noting that there are no strong correlations identified between the three groups of competences clusters, even though, we can still observe some peculiarities in every collection of competences. In fact, profiles that have an association score higher than 20% in 2 collections of competences are rare.

As shown in Fig. 6, the *Data Technologies* competences collection maintains consistently low scores when it relates to the rest of the clusters. Further, *Web Technologies* have a distributed association score when they relate to Data Technologies, but relatively high association scores when they relate to DevOps.

4.2.3. Relationships between collections of competences

Finally, the *DevOps* competences collection has very low association scores relating to Data Technologies and fairly distributed scores when the cluster relates to Web Technologies. Therefore, practitioners with DevOps competences can also have any association with Web Technologies, but typically they have lower association scores with Data Technologies.

Answering RQ2:
The vast majority of the microservices-related profiles associate with Web Technologies and DevOps competences and almost half of the profiles associate with Data Technologies. Most practitioners have a wide range of association scores with Web Technologies (between 20% and 80%. On the contrary, most practitioners have a tendency to associate with lower association scores to DevOps and Data Technologies competences. The identified collections of competences vary in how they relate to each other. It is worth noting that most profiles with high association to Web Technologies also have some competences in DevOps.

4.3. RQ3: Roles of engineers in developing MSAs

Next, we determine the primary competences of each profile to determine their role, from the competences collection with the highest association score. For example, for the profile in Fig. 4, we determine that their primary competence is Web Technologies, as the user has the highest association score for this competences collection. Fig. 7, shows the number of profiles in which every competences collection is the primary competence. Specifically, Web Technologies is the primary competence of 7250 profiles, which we describe as the *Web-based software engineers* role. DevOps is the primary competence of 1822 profiles, which we describe as the *DevOps engineers* role. Data Technologies is the primary competence of 407 profiles, which we describe as the *Data engineers* role. Finally, one of the three Stand Alone competences is the primary competence for 3103 profiles.

However, most profiles will also have positive association scores (larger than 0) to at least one other complementary competences collection (besides their primary one). In this section we describe the combinations of primary and complementary competences that the three main roles have, to understand the profiles of practitioners that have those roles. The intensity of the complementary competences is shown in Fig. 8. We include the standalone competences as complementary ones. They serve as a baseline to the other competences, emphasizing that both Web Technologies and DevOps are fairly typical in their distributions as secondary skills, while Data Technologies seem to be associated with rather low scores and few profiles. Even though there are many practitioners having as their primary competence one of the stand-alone competences clusters, we do not associate them with a specific role of engineers, since they can have competences from all the other three collections.

4.3.1. Web-based software engineers

The *Web Technologies* competences collection is the most populated cluster from the identified ones, in terms of the number of profiles having this specific competences collection as their primary one. In this section, we focus our analysis on the subset of profiles whose primary competences belong to the *Web Technologies* competences collection. An example of such a profile can be seen in Fig. 9, having 13 Web Technologies tags, 7 DevOps tags, 8 Data Technologies tags, and 7 stand-alone tags.

As shown in Fig. 8(a), most *Web-based software engineers*, also have a percentage of tags associated with the other two clusters, namely *DevOps* and *Data Technologies*. The association scores of the complementary competences collections are relatively low, with most complementary competences not exceeding 30%. On the one hand, Web-based software engineers that have complementary DevOps competences have distributed association scores, ranging from approximately 12% to 27%. Interestingly, there is a somewhat high density in Web-based software engineers that associate with DevOps competences between 20 − 30%. Therefore, Web-based software engineers are very often competent in DevOps, or they are required to have some DevOps competences. On the other hand, Web-based software engineers that are also represented in the Data Technologies competences collection seem to typically have low association scores, with very few exceeding 20%.

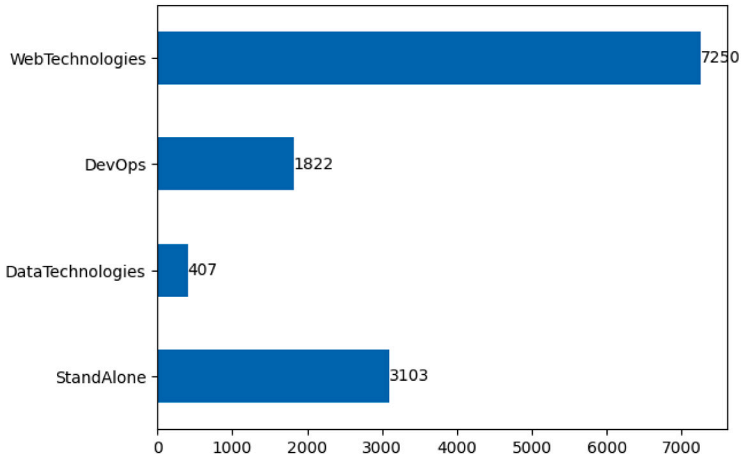


Fig. 7. The amount of MSA-related profiles that are primarily associated with each competences collection.

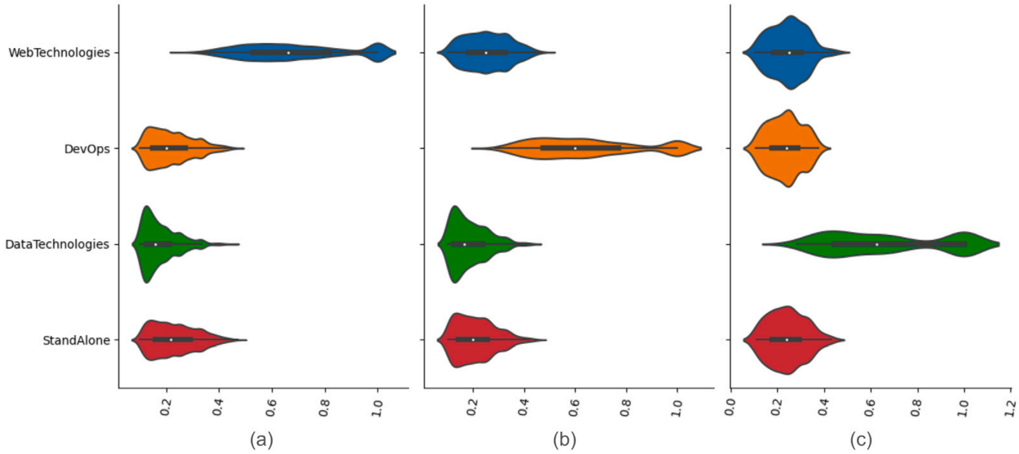
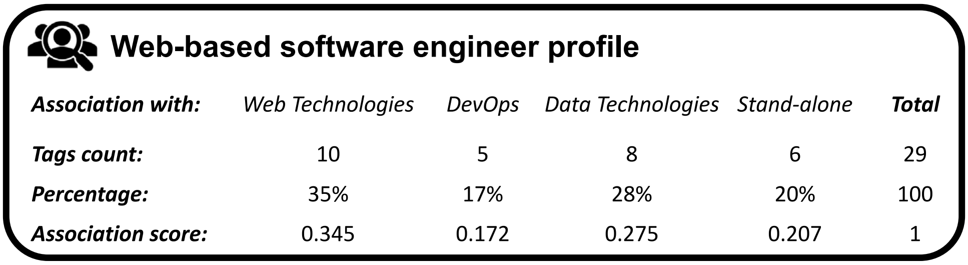


Fig. 8. The primary and complementary competences association scores (X-axis) of *Web-based software engineers* in *(a)*, *DevOps engineers* in *(b)*, and *Data engineers* in *(c)*.



Web-based software engineer profile

Association with:	*Web Technologies*	*DevOps*	*Data Technologies*	*Stand-alone*	*Total*
Tags count:	10	5	8	6	29
Percentage:	35%	17%	28%	20%	100
Association score:	0.345	0.172	0.275	0.207	1

Fig. 9. An example of a Web-based software engineer profile.

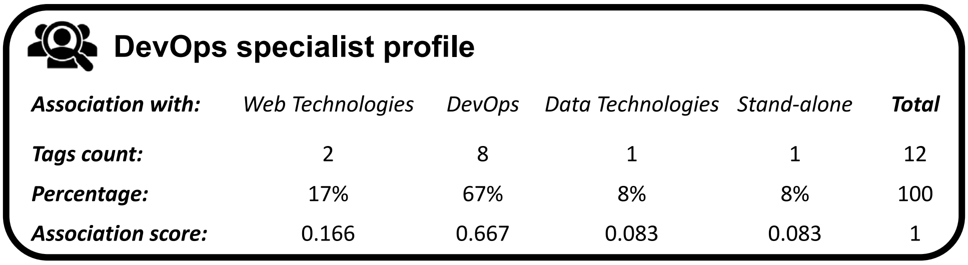
4.3.2. DevOps engineers

An important aspect of MSAs is DevOps, dealing with setting up engineers to seamlessly develop and operate their services. The *DevOps* competences collections are concerned with competences on setting up the infrastructure of systems. In addition, DevOps competences are those that implement the non-functional aspects of software systems, such as integrating, testing, and monitoring services. An example of a DevOps engineer can be seen in Fig. 10, associated with DevOps competences at 67%.

As shown in Fig. 8(b), DevOps engineers also have complementary competences from the other groups of competences clusters. Specifically, we observe that competences related to Web Technologies are quite popular, with relatively high association scores for being a secondary competences cluster. Most DevOps engineers with Web Technologies competences, are associated with the Web Technologies group of clusters with a score between 15% and 35%. On the other hand, there are DevOps engineers that relate to Data Technologies competences collection. Typically, these DevOps engineers have a relatively low association with Data Technologies, with scores between 10% and 25%.

4.3.3. Data engineers

Data Technologies mark their presence as the smallest competences collections, in terms of presenting the primary competences of practitioners. Data Technologies are dealing with handling the data of software systems, both in the sense of the data model of applications, as



DevOps specialist profile

Association with:	*Web Technologies*	*DevOps*	*Data Technologies*	*Stand-alone*	*Total*
Tags count:	2	8	1	1	12
Percentage:	17%	67%	8%	8%	100
Association score:	0.166	0.667	0.083	0.083	1

Fig. 10. An example of a DevOps specialist profile.

well as the logic of analyzing data assets. An example of a Data engineer can be seen in Fig. 11, associated with Data Technologies competences at 40%.

As shown in Fig. 8(c), Data engineers typically have complementary competences with a relatively high association score (in comparison to the other roles' complementary competences). In particular, we can observe that both complementary competences of Web Technologies and DevOps, are very popular to make their appearance with association score of 20% - 25%. This indicates that Data engineers are rarely exclusively concerned with Data Technologies. On the contrary, usually they have other complementary competences that they are concerned with.

4.3.4. Practitioners that specialize in their primary collection of competences

Table 16 shows the percentage of profiles that are specialists in one collection of competences. As described in Section 4.2, specialists are considered the profiles that have an association score higher than 50% in one particular collection of competences. In this regard, a total of 9185 profiles out of the 13,517 profiles analyzed are specialists in one collection of competences, accounting for 68% of the analyzed profiles being considered as specialists.

Table 16
The percentage of specialist profiles.

	Generalists	*Specialists*
Microservices practitioners	32%	68%
Web-based software engineers	23%	77%
DevOps engineers	44%	66%
Data engineers	41%	59%

Furthermore, by breaking down the data further, we can see the percentage of specialists in each identified role. Specifically, 77% of Web-based software engineers (5583) are also specialists in web technologies. In addition, 66% (1202 DevOps engineers) are specialists in their primary collection of competences and 59% (241) of Data Technologies engineers are specialists. This means that most practitioners show a tendency to more confidently answer on one distinct collection of competences, even though they occasionally discuss about topics that indicate different competences from their primary one.

Answering RQ3:
The overarching competences clusters are based on the three overarching themes of the competences taxonomy, Hence, we identify (1) *Web-based software engineers,* (2) *DevOps engineers and* (3) *Data engineers*. All roles are primarily about one of the competences clusters but they might have complementary competences, associating also with other competences. Therefore, the profiles of practitioners in microservices can have a predominant specialization, but very often they are required to at least have a comprehensive technical skillset.

4.4. RQ4: Characteristics of roles in MSA

In this section, we describe in detail the characteristics of each identified role (based on the competences collections). Specifically, we discuss at what density each of the different competences clusters is apparent in each of the identified roles.

4.4.1. Characteristics of web-based software engineers

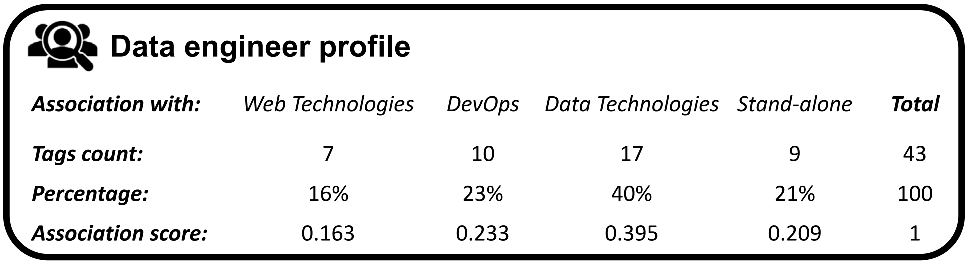
Web-based software engineers can be characterized in more detail, by observing the specific competences that they associate with, as shown in Fig. 12(a). Web-based software engineers associate with API development and service integration at a high distribution, reaching also high association scores. Hence, profiles that have API development and service integration competences have a lower remaining score for complementary competences. Consequently, APIs and service integration is a competences cluster in which practitioners have a high specialization. In addition, it is a competences cluster that practitioners often have exclusively, meaning that it is not complemented with other competences.

On the contrary, most Web-based software engineers that are associated with the competences collection through Security and networking, score below 40% (specifically, around 20%). Therefore, Web-based software engineers with security and networking competences are more likely to have other complementary competences. Web-based software engineers that have full stack development as their primary competences cluster have fairly distributed association scores, ranging from 15% to 60%. There are Web-based software engineers with primary competence in full-stack development that are likely to have other complementary competences, but there are many that have high scores, meaning that they are exclusively associated with the specific competence cluster. Finally, most front-end developers associate with the competences cluster with relatively low scores. Consequently, Web-based software engineers that have primary competence in front-end development are likely to have secondary competences.

However, it is worth noting that most profile answers are accounted for tags that concern APIs and service integration as well as Full-stack development, as shown in Fig. 12. Front-end development as well as security and networking seem to be less represented in the answers given, potentially due to fewer questions being asked about those topics.

4.4.2. Characteristics of DevOps engineers

Fig. 13(a) demonstrates how the scores of DevOps engineers are distributed across their primary competences cluster, namely *Version Control and Quality Assurance* and *Monitoring and CI/CD*. On the one hand, we observe a large number of DevOps engineers that associate with competences about version control and quality assurance with scores between 15% and 40%. Such scores indicate that DevOps engineers that are concerned with version control and quality assurance are more likely to be associated with other competences as well. On the other hand, we observe that there are DevOps engineers that are primarily associated to monitoring and CI/CD. The association scores of these DevOps engineers to their primary competences cluster is



Data engineer profile

Association with:	*Web Technologies*	*DevOps*	*Data Technologies*	*Stand-alone*	*Total*
Tags count:	7	10	17	9	43
Percentage:	16%	23%	40%	21%	100
Association score:	0.163	0.233	0.395	0.209	1

Fig. 11. An example of a Data Engineer.

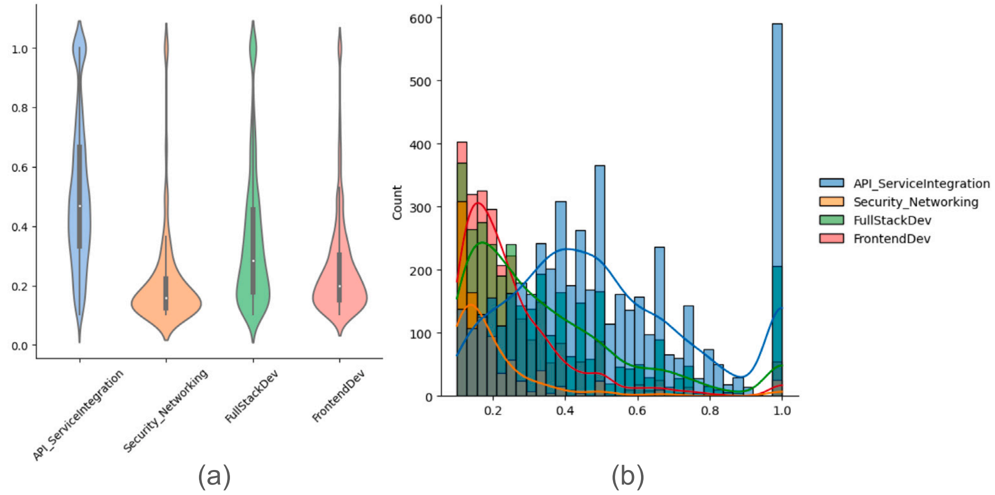


Fig. 12. Analysis results of the *Web Technologies* competences collection with *(a)* the association score (Y-axis) of each competence cluster in the collection and *(b)* the distribution of each competence in the cluster (Y-axis is the count of profiles and X-axis the association score).

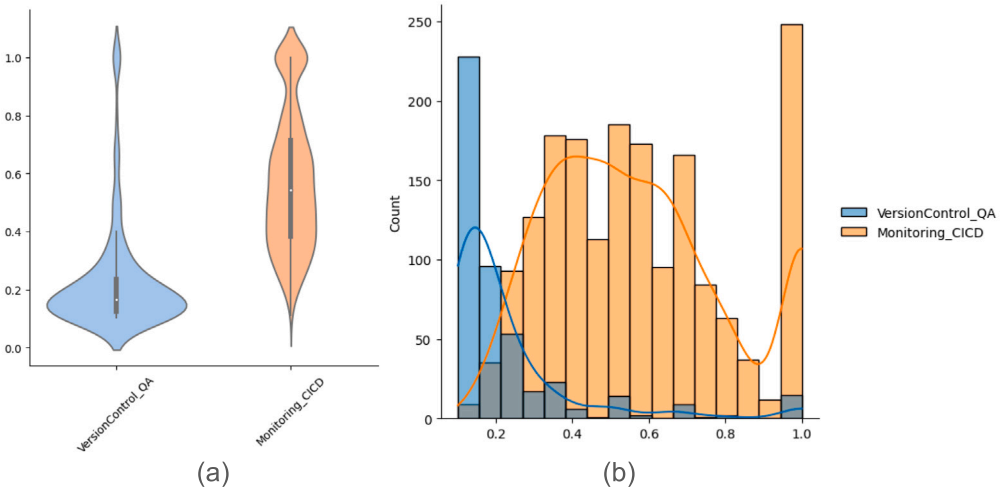


Fig. 13. Analysis results of the *DevOps* competences cluster with *(a)* the association score (Y-axis) of each competence cluster in the collection and *(b)* the distribution of each competence cluster in the collection (Y-axis is the count of profiles and X-axis the association score).

between 20% and 80%. This is a relatively large distribution of scores, which means that there are both DevOps engineers on monitoring and CI/CD that also have other competences, as well as DevOps engineers that are more exclusively concerned with monitoring and CI/CD.

On a similar note, we can observe in Fig. 13(b) that there is a large number (approximately 200 profiles) of DevOps engineers that are exclusively concerned with monitoring and CI/CD at a rate of 100%. This means that DevOps engineers on monitoring and CI/CD are not likely to have other complementary competences. On the contrary, there are very few DevOps engineers that are associated with a high score on version control and quality assurance. In fact, Fig. 13(b) shows that most DevOps engineers on version control and quality assurance score on the specific competences cluster around 10%. This implies that engineers associated with version control and quality assurance are likely to also have other complementary competences.

4.4.3. Characteristics of data engineers

Data engineers are due to either have primary competences in data analytics and data engineering or for having competences in data management. As shown in Fig. 14(a), the association scores in data analytics and data engineering, are fairly distributed, with more MSA-related profiles being associated between 20% and 45%. Interestingly, there are very few profiles that associate with the specific competences cluster with less than 20% and a moderate amount that associate with 50% or more. This showcases that most Data engineers who specialize in data analytics and data engineering are either exclusive on those competences, or at least highly associated with their primary competence. On the contrary, Data engineers with primary competence in Data Management, are typically low scoring in their primary competence. Most profiles that are primarily concerned with Data Management have an associated score between 15% and 25%. Such association scores mean that data engineers with data management competences are typically concerned with other complementary competences as well and it is rare that their primary competences are exclusive to other topics.

Fig. 14(b) supports further the exclusive specialization of the two competences clusters. Specifically, it can be observed that a significant amount of Data engineers (approximately 100), are exclusive on the data analytics and data engineering competences, where only 20 profiles are exclusive in the data management competences and 60 profiles have below 20% score.

Answering RQ4:
The roles of practitioners in MSAs are characterized in a variety of ways, through many different competences. On the one hand, some competences clusters indicate exclusive specialty, meaning that practitioners specializing in some competences are less likely to show prominence in complementary competences. On the other hand, some competences clusters indicate less specialty, showing a tendency of practitioners specialized in them to have more complementary competences.

5. Discussion

This study combines the two streams of research about (1) MSAs, and (2) understanding the developer profiles. Specifically, we enrich the research landscape that investigates the practitioners' perceptions of MSAs, with the technical competences that practitioners have. In this section we discuss the implications of this research (for both research and practice), along with future work that can further advance this area of research.

Web developers, DevOps engineers, and data engineers are the key roles in MSA projects. Research has been taking place to identify the general technical roles of practitioners (Montandon et al., 2021b). Our study specifies the roles of practitioners in MSAs specifically(namely Web developers, DevOps engineers, and data engineers), largely confirming the roles identified by Montandon et al. (2021b). In addition, we enrich those roles with more competences clusters and more details about the roles. For example, we provide detail to the description of the DevOps technical role identified by Montandon et al. (2021b) with the competences of version control and quality assurance as well as monitoring and CI/CD.

The angle of MSAs in researching software engineering practitioners provides non-trivial insights due to the challenges that organizations and practitioners face in understanding, staffing, and training practitioners, specifically for microservices-related roles. Therefore, it could not just be assumed that the competence clusters would be the same as the ones by Montandon et al. (2021b), even though there is some overlap. In addition, whereas current literature focuses largely on how different machine learning classifiers can be used to distinguish between roles (and at what accuracy), our contributions focus heavily on empirically understanding the roles of microservices practitioners specifically. Hence, this study contributes with detailed descriptions (both qualitative and quantitative) of the characteristics of the roles, how they relate to each other, and how they relate to the engineering principles of microservices (besides the development languages, tools, and frameworks).

Microservices practitioners should have competences in at least one, ideally multiple, of the core MSA roles (web developer, DevOps engineer, or data engineer). The three main collections of competences indicate the primary technical skills that practitioners who want to work with microservices should have. Furthermore, we organize the backend and frontend roles to a microservices-specific context and we extend them with competences such as security. Finally, we evolve the data science role with more detail and stronger connection to the microservices context.

Therefore, frameworks for recruiting and training microservices practitioners, as well as personal development plans of practitioners should focus on these topics. In addition, educators can use the primary competences identified to design the topics of their educational content related to MSAs and microservices.

As a next step, studies should be conducted to empirically validate and extend our initial taxonomy of technical competences. We believe that case studies with organizations that have fairly advanced MSAs, similar to the work of Zhou et al. (2023), can validate the taxonomy in light of the practical status quo of microservices practitioners. In addition, researchers can evolve and complement the competences taxonomy using additional sources to identify competences that could not be derived from StackOverflow. Importantly, researchers can evolve and complement the taxonomy with more competences and specifically social competences and soft skills that appear in the professional contexts of working with MSAs.

There are 6 fundamental competences that professional microservices training programs should account for. Our analysis can help in shaping professional training programs that prepare microservices practitioners. We argue that the identified primary competences that typically appear along with other competences are fundamental to MSAs. That is because they are essential for practitioners, but do not consist exclusive topics of interest. Therefore, microservices training programs should start from those topics, to establish solid bases for the more advanced contents and specialization competences. The fundamental competences of microservices are *(1) Fullstack development, (2) Frontend development, (3) Security and Networking, (4) Version Control and Quality Assurance, (5) Data Management, (6) Programming, Data Structures and Algorithms*. These competences are derived from the competences with fewer profiles having high association scores in Figs. 12, 13, 14.

Professional training programs can focus on 5 competences for deep specialization (after covering the 6 fundamental competences). For engineers to achieve a 'T'-shaped profile of competences they need to obtain the 6 fundamental competences, but also master at least one specialization competence. Based on the blueprint of confident (in microservices) StackOverflow profiles, the competences that engineers are mostly specialized at are *(1) API development and Service integration, (2) Monitoring CI/CD, (3) Data Analytics and Engineering, (4) Specific cloud vendor's services, and (5) Mobile development*. Therefore, advanced training programs that aim at deep specialization of engineers should focus on one of those competences. Of course, as indicated in the results of RQ3 (Fig. 8), deep specialization is accompanied by complementary competences and therefore other topics should be included in specialization training programs.

Web technologies competencies are at the heart of microservice development. Web technologies competences are substantially overproportionally present in the profiles we analyzed, both as primary or complementary competences. Web technologies are less likely to be complementary competences and Web-based software engineers

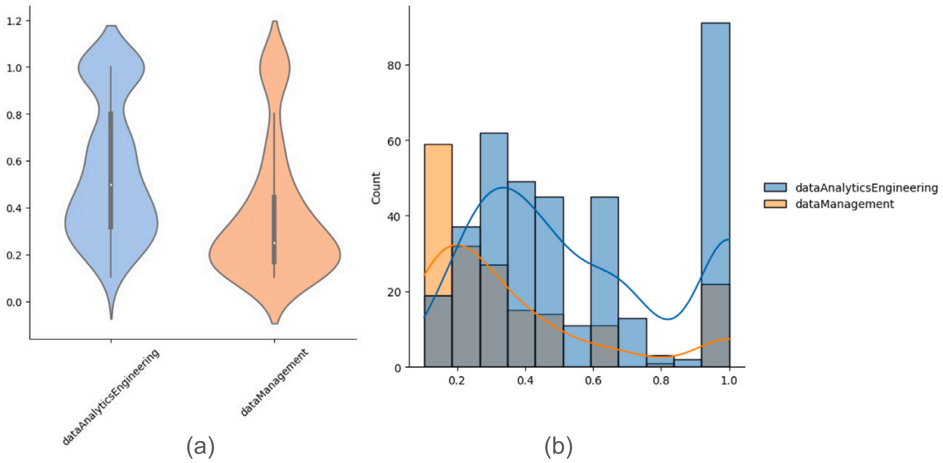


Fig. 14. Analysis results of the *Data Technologies* competences cluster with *(a)* the association score (Y-axis) of each competence cluster in the collection and *(b)* the distribution of each competence in the cluster (Y-axis is the count of profiles and X-axis the association score).

are more likely to be specialized. Profiles in API and service integration in MSAs are specialists, more likely to be exclusive in their primary competences. Associations with front-end development competences are low, indicating that are highly likely to be accompanied by other competences as well. As expected, from their cross-functional nature, security-concerned competences are also complemented with other competences. Full-stack development has exclusivity but this can be taken with a grain of salt since full-stack development can be characterized as de facto multi-competence.

We see two potential explanations for this and we argue that in reality, a combination of the two explain the core role of web technologies competences. (a) On the one hand, it is plausible that web technologies are simply so central to the idea of microservices that most MSA practitioners need some level of competence in (some) web development topics. (b) Alternatively, the high prevalence of web development competences in our dataset could also be an artifact of these competences simply being more accessible (more commonly taught in formal education and easier to learn individually) than, for instance, specialized DevOps competences.

We observe that most current microservices research focus on deployment, integration, communication and the infrastructure of services as well as transitions to microservices. However, it seems that web technologies are still essential for practitioners. Hence, researchers can take further into account the prominence of web technologies in MSAs and investigate more specifically their importance within MSAs. For example, we think it is important to clarify which web technologies tools and frameworks are most suitable for microservices-based designs and development processes. In addition, future research can investigate which tools and web technologies are used across the industry for different purposes.

Microservices teams consist of a combination of generalists and specialists. The results of our analysis indicate that practitioners with different primary competences are concerned with those competences at a different intensity. Practitioners of MSAs can have different levels of specialization across different topics, depending on their responsibility. For example, they can be deeply concerned about a specific topic (e.g., APIs and service integration), without direct responsibility for other aspects of MSAs. However, many practitioners are generalists rather than specialists, with skill-sets distributed across two or three of the identified collections of competences.

In fact, a large majority (>80%) of profiles working with MSAs one way or the other have to deal with APIs and service integration competences, Full-stack development competences, and Monitoring and CI/CD competences. Therefore, programs that develop or educate the technical competences of microservices practitioners require to include the topics of APIs and service integration, full-stack development, as well as monitoring and CI/CD.

Future research shall investigate more sources of practitioners working with MSAs, understanding more aspects of their profiles. For example, subsequent studies can investigate how technical profiles of practitioners working with MSAs differ from other developers. In addition, researchers can even investigate how the personality traits and profiles of engineers compare across different technologies. Another important angle is to investigate how developers learn different new technologies and what tools need specialization or what tools can be used sufficiently with not so deep knowledge about them. Therefore, qualitative-based research such as interview studies and observations studies can be very beneficial for investigating the practitioners' perspectives.

Data technologies and DevOps mainly complement the competences of a microservices engineer. Practitioners that have Data Technologies competences have a consistently low association score. However, Data Technologies are very apparent to other clusters as a complementary competence, which indicates the complementary nature of data management, analytics, and processing tasks that engineers have. The complementary nature of these tasks is in the sense that they take place along with other work tasks that microservices practitioners do. This applies (to an even larger extent) to DevOps competences, where most practitioners in our dataset have some competence from DevOps, even though it is not their primary competence. Version control and quality assurance is a cross-cutting competence in generalists engineers. Monitoring and CI/CD competences are exclusive to specialists engineers.

Future studies shall focus on investigating the different detailed responsibilities that engineers take across their different roles (i.e., the description of their specialty and responsibilities). Specifically, case studies in organizations that work with microservices and questionnaires to microservices practitioners can lead to valuable insights, by aggregating what practitioners in different teams specialize on.

There are additional competences clusters of foundational or orthogonal knowledge that are important to miroservices practitioners. In addition to the three core collections of competences, three competences clusters exist among practitioners. On the one hand, competences in programming, data structures, and algorithms are foundational, since such knowledge is the basis for any advanced knowledge of the technical structures of software systems. On the other hand, mobile development competences can be seen as complementary — they often appear in the context of microservices, but are in no way central to the idea of this architectural style. Finally, we observe that the Microsoft technology stack is different in the sense that technologies from this vendor tend to be used with other Microsoft technologies to a much larger degree than for other technology stacks. Hence, Microsoft-focused MSA practitioners are often Microsoft experts more than specifically DevOps or Data management experts.

This suggests that additional compentences and roles for MSA practitioners exist, which future research should attempt to elucidate. Particularly promising in this regard are studies that draw from different data sources than StackOverflow profiles, e.g., GitHub, job postings or practitioner surveys.

6. Threats to validity

The threats to validity of our study are discussed below in the form of internal, external, and construct validity.

Internal validity. The tags that are associated with profiles are based on the posts to which they gave answers, which might not be their only competences. In fact, it is expected that the roles identified are of practitioners that have more competences than just the ones that they discussed in StackOverflow. Therefore, we do not claim that those roles indicate the only competences that practitioners have. In addition, the associated tags are based on topics that practitioners felt confident enough to discuss publicly in the online forum of StackOverflow. However, it could be the case that their confidence in the discussed topics does not necessarily mean that they are deeply specialized in them, but rather that they are knowledgeable about specific aspects of the topics. Furthermore, it is possible that tags do not contain information regarding the context of developers' work in different technologies, or in some cases, tags are selected incorrectly in order to get more views and quicker responses. Hence, our findings are framed and interpreted in light of the definition we give of what a competence is — the basic ability and knowledge of a specific topic and not the deep specialization on the topic. The scale of the analysis (13,517 profiles) in combination with the fairly aggressive filtering of tags and profiles we conduct partially mitigates this threat since it is unlikely that the vast majority of profiles with answers in StackOverflow do not have at least the basic competences of the topics they discuss about.

External validity. Our dataset solely includes practitioners that are active in StackOverflow, which of course is not the entire population of practitioners who work with MSAs. Therefore, it is possible that the identified competences and roles are an artifact of the types of practitioners that get involved with such forums. Hence, we cannot claim that our taxonomy of competences is complete or entirely generalizable to the entire population of practitioners working with MSAs. In addition, it is possible that the analyzed dataset (after our filtering and sampling) excludes a part of the population of microservices practitioners that are already in StackOverflow. In fact, there might be more competences that are not visible in profiles active in StackOverflow and some of the competences or roles identified might interplay differently in different contexts. Nevertheless, our findings are still an empirical contribution that evolves our collective understanding of practitioners working with MSAs. In addition, the threat does not also mean that there is nothing to learn from StackOverflow profiles since many of the profiles can actually be advanced users that articulate and share their knowledge and experiences. Consequently, the potential incompleteness of the analyzed data does not necessarily hinder generalizability, since we can have confidence that the findings are representative for a large part of the microservices practitioners population.

Construct validity. As in all such large-scale analyses of online data, it is possible that our filtering of the data could not exclude all false positives, or actually excluded some profiles that are fairly advanced. We try to mitigate the risk of including falsely advanced users by including only user-tags combinations that are associated with answers. Hence, we assume that if a practitioner articulated and communicated an answer to a discussion then the practitioner has certain knowledge on the topic. Furthermore, the decision to filter out tags that appear in fewer than 100 user profiles poses a threat since several datapoints gathered are omitted from the analysis. However, the number of user profiles filtered out from this criterion is not significant and this filtering helped to address the issue of having several different tags regarding the same technology but in different versions. Future work could investigate further other potential research methodologies, to include datapoints with tags that are less commonly used. Moreover, we recognize that the association scores that we calculate for profiles with competences are just an indication of the reality of the people behind the profiles and not the absolute truth of what characterizes them. Having said that, we argue that the relative association and the re-occurring patterns of our analysis are still a step towards our further understanding of the practitioners working with MSAs.

7. Conclusion

MSAs are becoming a popular architectural style structuring many software systems (Newman, 2015). In addition, research calls for investigations for understanding the technical roles and competences of software engineering practitioners (Montandon et al., 2021b,a). On the one hand, this study contributes to engineering systems that use MSAs and on the other hand, to understanding different profiles of engineers in terms of their technical competences.

We contribute to the body of knowledge of understanding software engineers, by deriving 11 microservices-related technical competences clusters that software engineers that work in MSAs have. In addition, we identify the 3 main roles that practitioners have in microservices, based on identified collections of competences, along with their specific characteristics. We complement existing research on microservices with the roles of the people that engineer microservices-based systems as well as the characteristics of these roles. Our results are derived from investigating an initial dataset of 21,189 public profiles of software engineers, filtered to 13,517 profiles with 508 tags.

The results of this study indicate how practitioners are concerned with different topics of MSAs. In fact, we derive that most practitioners of MSAs are dominant in Web Technology competences. In addition, some are specialists in DevOps. Even though there is a significant amount of practitioners that are not specialists in DevOps, many practitioners have DevOps as complementary competence. Finally, few practitioners are specialists in Data Technologies, even though many more practitioners have Data Technologies as their complementary competence.

We use tags associated with answers as indicators or a proxy for deriving the competences of users. Finally, we derive competences clusters to associate together related competences (i.e., competences that cover distinct topics or areas of expertise). In this way, we aggregate together the technical topics that particularly interest different practitioners, providing a view of the collections of competences that are found in MSAs. Thereafter, we use the identified collections of competences to derive the roles that exist in developing software systems that use MSAs and we describe the detailed characteristics of these roles. The large-scale analysis of practitioners' profiles complements existing research on the design, development, and maintenance of systems based on microservices, by giving a comprehensive perspective of their roles and competences.

CRediT authorship contribution statement

Hamdy Michael Ayas: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Regina Hebig:** Writing – review & editing, Validation, Supervision. **Philipp Leitner:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

No funding agency had any input on the research design or analysis of this study.

Data availability

The publicly available replication package contains data and code files.

References

Baškarada, S., Nguyen, V., Koronios, A., 2018. Architecting microservices: Practical opportunities and challenges. J. Comput. Inf. Syst.

Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. 2008 (10), P10008.

Brooks, Jr., F.P., 1995. The Mythical Man-Month: Essays on Software Engineering. Pearson Education.

Buchgeher, G., Winterer, M., Weinreich, R., Luger, J., Wingelhofer, R., Aistleitner, M., 2017. Microservices in a small development organization: An industrial experience report. In: Software Architecture: 11th European Conference, ECSA 2017, Canterbury, UK, September 11-15, 2017, Proceedings. Vol. 11. Springer, pp. 208–215.

Capiluppi, A., Izquierdo-Cortázar, D., 2013. Effort estimation of FLOSS projects: a study of the Linux kernel. Empir. Softw. Eng. 18, 60–88.

Cerny, T., Al Maruf, A., Janes, A., Taibi, D., 2023. Microservice anti-patterns and bad smells. In: How to Classify, and How to Detect Them. A Tertiary Study.

DeMarco, T., 1999. It ain't broke, so don't fix it [software engineering]. IEEE Softw. 16 (6), 67.

Di Francesco, P., Lago, P., Malavolta, I., 2019. Architecting with microservices: A systematic mapping study. J. Syst. Softw. 150, 77–97.

Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R., Safina, L., 2017. Microservices: yesterday, today, and tomorrow. In: Present and Ulterior Software Engineering. Springer, pp. 195–216.

Dragoni, N., Lanese, I., Larsen, S.T., Mazzara, M., Mustafin, R., Safina, L., 2018. Microservices: How to make your application scale. In: Petrenko, A.K., Voronkov, A. (Eds.), Perspectives of System Informatics. Springer International Publishing, Cham, pp. 95–104.

Fritzsch, J., Bogner, J., Wagner, S., Zimmermann, A., 2019. Microservices migration in industry: Intentions, strategies, and challenges. In: 2019 IEEE International Conference on Software Maintenance and Evolution. ICSME, pp. 481–490. http://dx.doi.org/10.1109/ICSME.2019.00081.

Hassan, S., Bahsoon, R., Kazman, R., 2020. Microservice transition and its granularity problem: A systematic mapping study. Softw. - Pract. Exp. 50 (9), 1651–1681.

Jamshidi, P., Pahl, C., Mendonça, N.C., Lewis, J., Tilkov, S., 2018. Microservices: The journey so far and challenges ahead. IEEE Softw. 35 (3), 24–35.

Kim, M., Zimmermann, T., DeLine, R., Begel, A., 2016. The emerging role of data scientists on software development teams. In: Proceedings of the 38th International Conference on Software Engineering. ICSE '16, Association for Computing Machinery, New York, NY, USA, pp. 96–107. http://dx.doi.org/10.1145/2884781.2884783.

Lambiotte, R., Delvenne, J.-C., Barahona, M., 2008. Laplacian dynamics and multiscale modular structure in networks. arXiv preprint arXiv:0812.1770.

Liang, J.T., Zimmermann, T., Ford, D., 2022. Understanding skills for OSS communities on GitHub. In: Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 170–182.

Marlow, J., Dabbish, L., Herbsleb, J., 2013. Impression formation in online peer production: activity traces and personal profiles in github. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work. pp. 117–128.

Meesters, M., Heck, P., Serebrenik, A., 2022. What is an AI engineer? An empirical analysis of job ads in The Netherlands. In: Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI. pp. 136–144.

Michael Ayas, H., Fischer, H., Leitner, P., De Oliveira Neto, F.G., 2022. An empirical analysis of microservices systems using consumer-driven contract testing. In: 2022 48th Euromicro Conference on Software Engineering and Advanced Applications. SEAA, pp. 92–99. http://dx.doi.org/10.1109/SEAA56994.2022.00022.

Michael Ayas, H., Hebig, R., Leitner, P., 2023a. Replication package: An empirical investigation on the competences and roles of practitioners in Microservices-based Architectures. http://dx.doi.org/10.5281/zenodo.7898566, Zenodo.

Michael Ayas, H., Leitner, P., Hebig, R., 2021. Facing the giant: A grounded theory study of decision-making in microservices migrations. In: Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. ESEM, pp. 1–11.

Michael Ayas, H., Leitner, P., Hebig, R., 2023b. An empirical study of the systemic and technical migration towards microservices. Empir. Softw. Eng. 28 (4), 85. http://dx.doi.org/10.1007/s10664-023-10308-9.

Miranda, G., Montandon, J.E., Valente, M.T., 2022. TechSpaces: Identifying and clustering popular programming technologies. In: Proceedings of the 16th Brazilian Symposium on Software Components, Architectures, and Reuse. SBCARS '22, Association for Computing Machinery, New York, NY, USA, pp. 60–67. http://dx.doi.org/10.1145/3559712.3559715.

Montandon, J.E., Politowski, C., Silva, L.L., Valente, M.T., Petrillo, F., Guéhéneuc, Y.-G., 2021a. What skills do IT companies look for in new developers? A study with Stack Overflow jobs. Inf. Softw. Technol. 129, 106429.

Montandon, J.E., Valente, M.T., Silva, L.L., 2021b. Mining the technical roles of GitHub users. Inf. Softw. Technol. 131, 106485.

Newman, S., 2015. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, Inc.

Sarma, A., Chen, X., Kuttal, S., Dabbish, L., Wang, Z., 2016. Hiring in the global stage: Profiles of online contributions. In: 2016 IEEE 11th International Conference on Global Software Engineering. ICGSE, IEEE, pp. 1–10.

Singer, L.-G., 2013. Improving the adoption of software engineering practices through persuasive interventions. Lulu. com.

Smith, E.K., Bird, C., Zimmermann, T., 2016. Beliefs, practices, and personalities of software engineers: A survey in a large software company. In: Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering. CHASE '16, Association for Computing Machinery, New York, NY, USA, pp. 15–18. http://dx.doi.org/10.1145/2897586.2897596.

Soldani, J., Tamburri, D.A., Van Den Heuvel, W.-J., 2018. The pains and gains of microservices: A systematic grey literature review. J. Syst. Softw. 146, 215–232.

Waseem, M., Liang, P., Ahmad, A., Shahin, M., Khan, A.A., Márquez, G., 2022. Decision models for selecting patterns and strategies in microservices systems and their evaluation by practitioners. In: Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice. pp. 135–144.

Waseem, M., Liang, P., Shahin, M., Di Salle, A., Márquez, G., 2021. Design, monitoring, and testing of microservices systems: The practitioners' perspective. J. Syst. Softw. 182, 111061. http://dx.doi.org/10.1016/j.jss.2021.111061, URL: https://www.sciencedirect.com/science/article/pii/S0164121221001588.

Wu, M., Zhang, Y., Liu, J., Wang, S., Zhang, Z., Xia, X., Mao, X., 2022. On the way to microservices: Exploring problems and solutions from online Q&A community. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering. SANER, IEEE, pp. 432–443.

Zhou, X., Li, S., Cao, L., Zhang, H., Jia, Z., Zhong, C., Shan, Z., Babar, M.A., 2023. Revisiting the practices and pains of microservice architecture in reality: An industrial inquiry. J. Syst. Softw. 195, 111521. http://dx.doi.org/10.1016/j.jss.2022.111521, URL: https://www.sciencedirect.com/science/article/pii/S0164121222001972.

Zimmermann, O., 2017. Microservices tenets: Agile approach to service development and deployment. Comput. Sci. - Res. Devel. 32 (3–4), 301–310.

Hamdy Michael Ayas is a Ph.D. candidate in Chalmers University of Technology and University of Gothenburg, Sweden. His main research interests are about the procedures, decision-making as well as the technical and organizational change that take place during migrations of software systems towards microservices-based architectures. Hamdy has an interdisciplinary background with experience in both software development and business analysis in the service management industry.

Regina Hebig is a Professor in Software Engineering at the Institute of Computer Science at the University of Rostock in Germany since 2023. She received her Ph.D. from the University of Potsdam, Germany in 2014 and worked as a PostDoc at the University Piere and Marie Curie in Paris, France between 2014-2015. Between 2015 and 2023 she worked as Assistant Professor and later Associate Professor at Chalmers | University of Gothenburg, Sweden. Her research interests include software evolution, software modeling, and software processes.

Philipp Leitner is an Associate Professor of Software Engineering and leads the Software Engineering 2 unit at the joined Computer Science and Engineering department at Chalmers and University of Gothenburg, Sweden. He holds a Ph.D. degree in business informatics from Vienna University of Technology. Philipp's research interests are software performance engineering, especially in Web and cloud computing contexts.