Microservice Smells and Automated Detection Tools: A Systematic Literature Review

Denis Pinheiro, Johnatan Oliveira and Eduardo Figueiredo

Federal University of Minas Gerals (UFMG)
Department of Computer Science (DCC)
{denis.pinheiro,johnatan.si,figueiredo}@dcc.ufmg.br

Abstract

Several microservice smells and automated detection tools has been proposed and studied in the literature, but they still lack a cohesive and organized documentation and consistent terminology. To fill this gap, this work presents a systematic literature review of 15 papers that investigated the most recent microservice smells and related detection tools. As a result, we present the landscape (where, when and how they occur) and document of 77 microservice smells (what), 2 detection tools and other 19 related tools commonly used to automate detection of architecture smells and support adoption of microservices.

1 Introduction

Microservice is an architectural style emerging out of service-oriented architecture, emphasizing self-management and lightweightness as the means to improve software agility, scalability, and autonomy [7]. Microservices put a strong emphasis on loose coupling and high cohesion of services, which are expected to result in a better scalability, adaptability, and quality of software architectures [16]. The microservice style is becoming popular in the development of cloud-based applications, but it also requires careful management of software quality attributes [17, 1].

Architectural smells are symptoms of poor design that can hinder code understandability and decrease maintainability [17]. Several architectural smells have been defined in the literature for both generic architectures and specific architectures [6, 17, 18]. However, cloud-native applications based on microservices can be affected by other types of issues [17] and the research field for service-based antipatterns and microservice smells is not as cohesive and organized [1].

The main goal of this work is to understand the literature related to microservices smells and automated detection tools (what); which venues have published related papers (where); the research interest on microservice smells over the years (when); and which types of researches being performed related to microservice smells (how). We performed a systematic literature review (SLR) following the guidelines proposed by Kitchenham [8]. We found 15 primary studies related to microservices smells and detection tools, and we also have collected 77 smells, 2 detection tools and 19 related tools commonly used to automate detection of architecture smells and support adoption of microservices. The way we present can help architects on understanding potential context of their occurrence.

2 Research Design

Following the guidelines proposed by Kitchenham [8], we planned the following steps: (i) identify the topics to be investigated (research questions); (ii) select the digital libraries; (iii) define the search string to retrieve relevant studies; and (iv) apply a filtering process with include/exclude criteria. The research questions to guide this study are:

RQ1. (When) How has the research interest on microservice smells evolved over time?
RQ2. (Where) Where have the papers related to microservice smells been published?
RQ3. (How) How are microservice smells researches being performed?
RQ4. (What) What are the microservice smells being reported by literature so far?
RQ5. (What) What are the microservice smells detection tools presented in the literature?

Database	**Address**	**Search String**
ACM Digital Library	`https://dl.acm.org`	"microservice" AND
IEEE Explore	`https://ieeexplore.ieee.org`	("architectural smell"
Engineering Village	`https://www.engineeringvillage.com`	OR "code smell"
Science Direct	`https://www.sciencedirect.com`	OR "bad smell"
Scopus	`https://scopus.com`	OR "anti-pattern"
Springer	`https://link.springer.com`	OR "technical debt")

Table 1: Digital Libraries and Search String

Table 1 presents the list of digital libraries used to perform this SLR. In addition, this table also presents the search string we run in June 1st, 2021. The reasoning for the search string is that different papers might use architecture smell or any synonym term, and it aims to reach a higher number of works related to microservices smells. The filtering process allows to classify each study under review as a candidate to be included or excluded from the SLR based on specific criteria. The inclusion and exclusion criteria used to select the primary studies are:
- Inclusion criteria: papers published in Computer Science, written in English and peer-reviewed (published in conferences, journals, workshops, scientific magazines).
- Exclusion criteria: papers with less than 5 pages, thesis, dissertations, tutorials, and grey literature (non scientific).

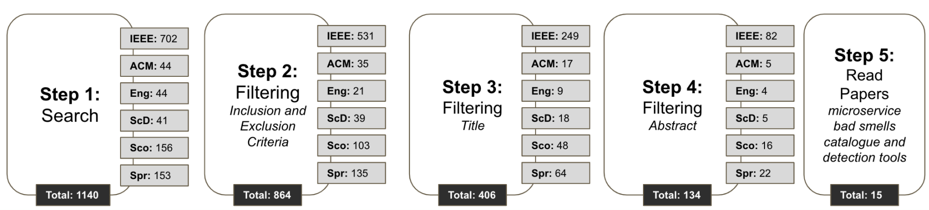

Figure 1: Research Execution Results

Figure 1 presents the SLR execution process in 5 steps by using the selected Digital Libraries (Table 1). The execution of the search steps were guided to fulfill and respond the research questions. The Step 1, resulted in all papers with the keywords ***microservice*** and ***architecture smells*** (or its synonyms) occurring somewhere in the paper text content, including all metadata. The search was performed considering all published works and resulted in 1,140 papers. In Step 2, all duplicated papers has been filtered and the inclusion and exclusion criteria have been applied ending this step with 864. Then, title and abstract were analyzed, ending Steps 3 and 4 with 486 and 134, respectively. Finally, in Step 5, it was selected 15 primary studies that have reported microservice smells or microservice smells detection tools.

3 Results and Discussion

RQ1 (When) - Figure 2 (A) shows the overall raising interest given when the selected primary studies that presented research related to microservice smells was published.

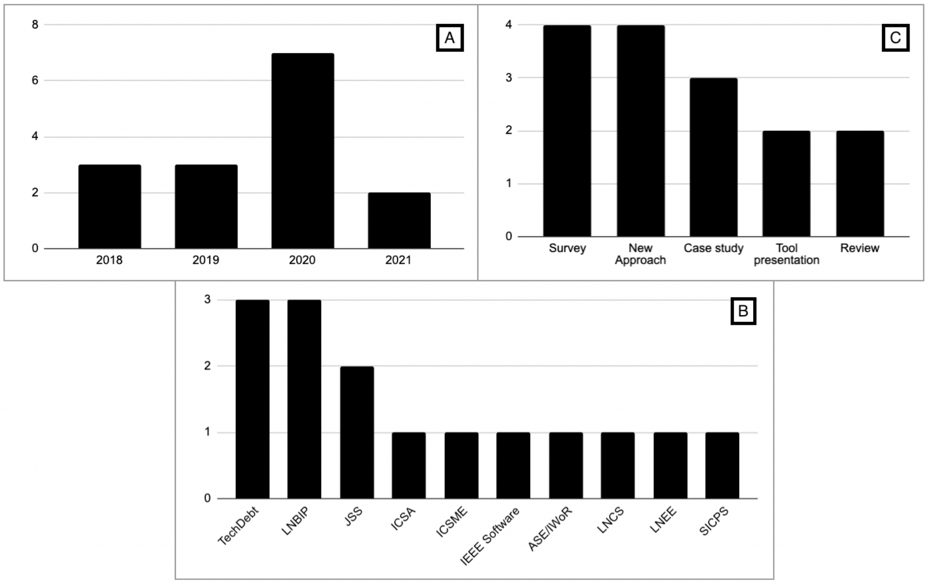

Figure 2: (A) Papers over time. (B) Publication venues. (C) Research types.

We found 3 relevant papers published in 2018 [17, 4, 2], which marks in time the starting interest on microservice smells. We have also selected 3 primary studies published in 2019 related to microservice smells [3][19][16]. The year of 2020 we have selected 7 studies related to microservice smells [12][14][5][20][9][15][10]. Finally, from Jan to May of 2021 we selected 2 primary studies [22][21]. Therefore, we can conclude that the interest on the subject of microservice smells is increasing over time, reaching its top interest in 2020.

RQ2 (Where) - Figure 2 (B) shows the publication venues and how many publications occurred in each one. **TechDebt** and **LNBIP** are both with most of the publications, total of 3 each. Then, we have **JSS** with 2 publications of the selected primary studies. All other publication venues have one publication each, including recognized architecture conference as **ICSA** and scientific magazine, such as **IEEE Software**.

RQ3 (How) - We considered following research types: **Survey** as the research performed by using a questionnaire or interviewing participants; **New Approach**, research proposing a new method that solves a common problem in microservice-based systems; **Case study**, research that studied a real or hypothetical problem related to microservice smells **Tool presentation**, research presenting a new tool or a new version of a tool; and **Review**, research that made a scientific review of white and gray literature.

Figure 2 (C) presents the research types identified in the selected studies. We found 4 papers that published the result of **Surveys** and also 4 studies that presented **New Approaches**. For instance, Taibi and Lenarduzzi (2018) presented at first time the 11 microservice smells catalogue as a result of interviewing 72 developers with experience in developing microservice-based systems [17]. As another example, Gaidels and Kirikova (2020) presented a service dependency graph analysis very useful in the process of identifying microservice smells [5]. We found 3 works that presented **Case study** as research method. For instance, Toledo et. al. [20] performed a multiple case study in 4 international companies, supported by 6 architects, and, as a result, the research reported 9 issues related to Shared Library and 2 solutions were proposed. Two (2) works performed a **Tool presentation** of automated microservice smells detection. Other two (2) researches presented a **Review** of white and gray literature related to architecture design and the migration process of microservice-based systems.

RQ4 (What/Smells) - Table 2 presents a reduced list of 21 microservice smells found in the selected studies. We reported the complete list of 77 microservices smells in all 15 selected studies in a supplementary website[1]. In this paper, we focus on smells that: (i) were reported by two or more studies; or (ii) were identified by a detection tool. In the first column, we present exactly the microservice smell name presented by the selected studies. The second column presents the smell categories, indicating the existence of symptoms of bad design and implementation decisions on microservices projects, given the classification made by their studies: **MsS** (Microservice Smell); **MsP** (Microservice Pitfall); **MTD** (Microservice Technical Debt); **ArcS** (Architecture Smell); **ATD** (Architecture Technical Debt); **LoM** (Symptom of Low Maintainability); **ChE** (Challenge for Evolution); **MigS** (Migration Smell). Detection tools are presented in the third and fourth columns (Arcan [14] and MSANose [22], respectively), as well as the indication of the detected microservice smells by each tool (more details in RQ5). Finally, the fifth column shows the selected studies that reported smells.

Microservice Smell (by study)	**Category** (by study)	**DT #1**[14] (Arcan)	**DT #2**[22] (MSANose)	**Studies**
API Versioning (Static Contract Pitfall)	MsS(MsP)		X	[17]
Cyclic Dependency	MsS	X	X	[17]
Shared Libraries	MsS		X	[17] [20]
Outdated Library	LoM/MTD			[2] [9]
ESB usage	MsS		X	[17]
Not Having an API Gateway	MsS/MDT/ArcS		X	[17][9][12]
Hard-coded endpoints	MsS(MsP)	X	X	[17]
Too Many Pont-to-Point Connections	MTD			[19] [15]
Shared Persistence (Data ownership)	MsS(MsP)/ArcS	X	X	[17] [12]
Inappropriate Service Intimacy	MsS		X	[17]
Microservice Greedy	MsS		X	[17]
Wrong Cuts	MsS		X	[17]
Single Layer Teams	ArcS			[4] [12]
Too Many Standards (Lust/Gluttony)	MsS(MsP)		X	[17]
Technological Heterogeneity	ChE/ATD			[3] [21]
Architectural/Technical Complexity	ChE/ATD			[3] [21]
No Standardized Communication Model	MTD/ATD			[15] [19] [21]
Business Logic Inside Comm. Layer	MTD/ATD			[15] [19] [21]
Diff. Middleware Tech. for Comm.	MTD			[15] [19]
Inadequate Testing	LoM/ChE			[2] [3]
Weak Source Code and Knowledge mgmt.	MTD			[15] [19]

Table 2: Microservice smells

RQ5 (What/Tools) - Piggazini et. al. [14] presented a new version of ***Arcan*** which aims to detect 3 microservice smells from the Taibi and Lenarduzzi catalogue [17]. The authors validated the results of the detection strategies by manually inspecting the detection of the smells in the open-source microservice projects first used by Marques and Astudillo [11] to analyze the actual use of architecture patterns. Recently, Walker et. al. [22] presented the ***MSANose*** tool that detects the 11 microservice smells reported by Taibi and Lenarduzzi [17] and it was evaluated on the detection accuracy by using 2 benchmarks projects [22].

[1]The exaustive list of 77 microservice smells are available at https://bit.ly/3ow5dSn.

Moreover, microservice-based system is also a software with source code which has potential of introducing both architecture and code smells that may be identified by regular detection tools, such as, *SonarQube*. Bogner et. al. [2][3] presented two studies of 19 tools to support and deal with symptoms of low maintainability and challenges to evolve microservices.

4 Threats to Validity and Related Work

The main threat to validity when performing the SLR is the validity of the results. We discussed the SLR validity with respect to the 3 groups of common threats to validity: internal, external, and conclusion [23]. A major *external validity* to this study was during the identified primary studies. The search for the papers was conducted in 6 relevant scientific databases in order to capture as much as possible the available papers. However, the quality of search engines could have influenced the completeness of the identified primary studies. A limitation of this study with respect to *internal validity* has been addressed by involving three researchers, and by a protocol that was piloted and evaluated. If the study is replicated by another set of researchers, it is possible that some studies that were removed in this review could be included and other studies could be excluded. Furthermore, from the authors' perspective, a potential threat to *conclusion validity* is the reliability of the data extraction from the primary studies, since not all information was obvious to answer the research questions and some data had to be inferred. Therefore, in order to ensure validity, sometimes cross-discussions among the paper authors took place to reach a common agreement. Furthermore, in the event of a disagreement between the two researchers, a third reviewer acted as an arbitrator to ensure a position to be reached.

The related work that aimed to build a comprehensive documentation on microservice smells started with Taibi and Lenarduzzi [18], the first work that provided a catalog with definitions and guidelines for 11 microservice-specific bad smells as a result of 72 interviews of experienced microservice developers. Bogner et al. [1] also performed a SLR focused on service-based anti-patterns and bad smells. They reveled 36 anti-patterns, but focused the work on presenting a holistic data model to describe and document service anti-patterns. Pereira-Vale et al. [13] focused on investigating security on microservice-based systems, performed a SLR on scientific papers (370) and grey literature (620). They aimed to provide a comprehensive catalog of security solutions and mechanisms to address security problems in microservice-based systems.

5 Conclusion and Future Work

This paper presented a systematic literature review on microservice smells and detection tools. We found that the interest on microservice smells is increasing over time since 2018 and Survey is the most recurring research type published in recognized scientific venues, such as TecDebt, ICSA and IEEE Software. We reported 77 microservice smells categorized by the selected primary studies, all representing symptoms of design or implementation bad decisions in microservice projects. We also found 2 microservice smells detection tools (Arcan and MSANose), both detecting microservice smells first presented by Taibi and Lenarduzzi (2018) [17]. As future work, we intend to compile a guideline based on the reported microservice smells to help architects to identify and avoid bad practices during development of microservices.

References

[1] J. Bogner, T. Boceck, M. Popp, D. Tschechlov, S. Wagner, and A. Zimmermann. Towards a collaborative repository for the documentation of service-based antipatterns and bad smells. In *International Conference on Software Architecture Companion (ICSA-C)*, pages 95–101, 2019.

[2] J. Bogner, J. Fritzsch, S. Wagner, and A. Zimmermann. Limiting technical debt with maintainability assurance: An industry survey on used techniques and differences with service- and microservice-based systems. In *International Conference on Technical Debt*, page 125–133, 2018.

[3] J. Bogner, J. Fritzsch, S. Wagner, and A. Zimmermann. Assuring the evolvability of microservices: Insights into industry practices and challenges. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 546–556, 2019.

[4] A. Carrasco, B. Van Bladel, and S. Demeyer. Migrating towards microservices: Migration and architecture smells. In *International Workshop on Refactoring (IWoR/ASE)*, pages 1 – 6, 2018.

[5] E. Gaidels and M. Kirikova. Service dependency graph analysis in microservice architecture. In *Lecture Notes in Business Information Processing (LNBIP)*, pages 128 – 139, 2020.

[6] J. Garcia, D. Popescu, G. Edwards, and N. Medvidovic. Toward a catalogue of architectural bad smells. In *Architectures for Adaptive Software Systems*, pages 146–162, 2009.

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

[8] B. A. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, 2007.

[9] V. Lenarduzzi, F. Lomio, N. Saarimäki, and D. Taibi. Does migrating a monolithic system to microservices decrease the technical debt? *Journal of Systems and Software*, 169:110710, 2020.

[10] P. Mahanta and S. Chouta. Translating a legacy stack to microservices using a modernization facade with performance optimization for container deployments. In *LNCS*, 2020.

[11] G. Márquez and H. Astudillo. Actual use of architectural patterns in microservices-based open source projects. In *Asia-Pacific Software Engineering Conference (APSEC)*, pages 31–40, 2018.

[12] D. Neri, J. Soldani, O. Zimmermann, and A. Brogi. Design principles, architectural smells and refactorings for microservices: a multivocal review. volume 35, pages 3 – 15, 2020.

[13] A. Pereira-Vale, E. B. Fernandez, R. Monge, H. Astudillo, and G. Márquez. Security in microservice-based systems: A multivocal literature review. *Computers & Security*, 2021.

[14] I. Pigazzini, F. A. Fontana, V. Lenarduzzi, and D. Taibi. Towards microservice smells detection. In *International Conference on Technical Debt (TechDebt)*, page 92–97, 2020.

[15] F. Rademacher, S. Sachweh, and A. Zundorf. A modeling method for systematic architecture reconstruction of microservice-based software systems. In *LNBIP*, pages 311 – 326, 2020.

[16] F. Rademacher, S. Sachweh, and A. Zündorf. Aspect-oriented modeling of technology heterogeneity in microservice architecture. In *International Conference on Software Architecture (ICSA)*, 2019.

[17] D. Taibi and V. Lenarduzzi. On the definition of microservice bad smells. *IEEE Software*, 2018.

[18] D. Taibi, V. Lenarduzzi, and C. Pahl. Architectural patterns for microservices: A systematic mapping study. In *Int'l Conf. on Cloud Computing and Services Science (CLOSER)*, 2018.

[19] S. Toledo, A. Martini, A. Przybyszewska, and D. Sjøberg. Architectural technical debt in microservices: A case study in a large company. In *TechDebt*, 2019.

[20] S. Toledo, A. Martini, and D. Sjøberg. Improving agility by managing shared libraries in microservices. *Lecture Notes in Business Information Processing (LNBIP)*, 2020.

[21] S. Toledo, A. Martini, and D. Sjøberg. Identifying architectural technical debt, principal, and interest in microservices: A multiple-case study. *Journal of Systems and Software*, 2021.

[22] A. Walker, D. Das, and T. Cerny. Automated microservice code-smell detection. In *Lecture Notes in Electrical Engineering (LNEE)*, pages 211 – 221, 2021.

[23] C. Wohlin. *Experimentation in Software Engineering: An Introduction.* The sKluwer ISECS. 2000.