UFMG

PROGRAMA DE PÓS-GRADUAÇÃO
CIÊNCIA DA COMPUTAÇÃO
UFMG

lab-soft
software engineering laboratory

Microservice Smells and Automated Detection Tools: A Systematic Review

Software Engineering Approaches

Denis Pinheiro, Johnatan Oliveira and Eduardo Figueiredo

Microservices 2022 - International Conference on Microservices

Summary

- Introduction
- Research Design
  - Research Questions
  - Digital Libraries and Query String
  - Filtering Process
  - Research Execution
- Results and Discussions
- Threats to Validity
- Related Work
- Conclusion and Future Work

Introduction

- The Microservice Architecture style (MSA) is becoming popular in the development of modern applications.
- Architectural and Code Smells are symptoms of poor design that can hinder code understandability and decrease maintainability.
  - Several architectural bad smells have been defined in literature.
  - Research field for service-based anti-patterns and bad smells is not as cohesive and organized.
- The goal is to present a systematic review of the literature related to microservices smells and related detection tools.

Background

- Microservice Architecture (MSA):
    - **Lews and Fowler (2014)**

"an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API."

- - **Architecture Style (Fowler, 2015)**
        - An alternative to monolithic applications (MA)
        - Shorter cycles for testing, build and release
        - Less frequently affected by downtime than MA
    - **Zimmermann (2017)**

"Microservice Architecture (MSA) may also be described as one way of implementing Service Oriented Architecture (SOA)."

Background

- **Architecture and Code Smells:**
  - The result in the source code of bad software design or implementation decisions.
    - Code or Architecture smells (anti-patterns) lead to Technical debt (quality degradation).
  - Fowler (1999) provided a catalog of 22 bad smells (and related Refactorings).
- **Microservice Smells:**
  - Architecture or Code Smells identified in the development of microservices.
  - Related microservice smells terms from the literature considered in this work:
    - Microservice smell (MsS)
    - Microservice Pitfall (MsP)
    - Microservice Technical Debt (MTD)
    - Architecture Smell (ArcS)
    - Architectural Technical Debt (ATD)
    - Symptoms of Low Manutenibility (LoM)
    - Challenges for Evolution (ChE))
    - Migration Smell (MigS)

Research Design

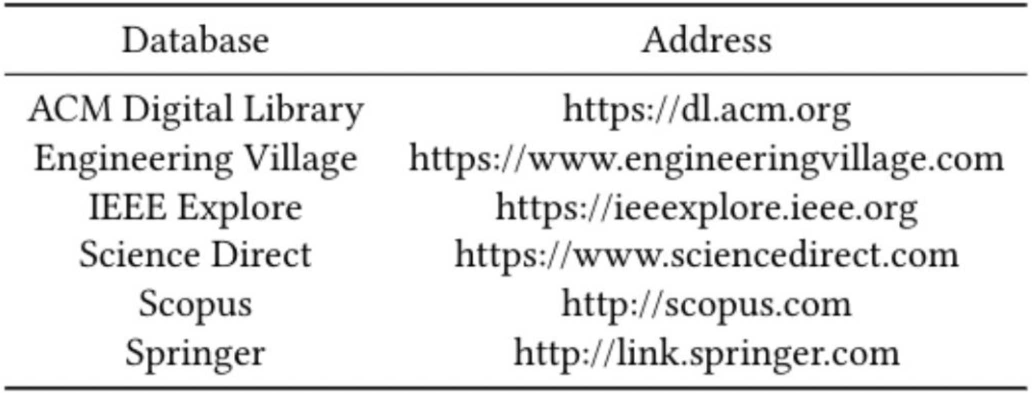
- Systematic Literature Review (SLR) following protocol of Kichenham (2007).

- Planning:
    - Identified the topics to be investigated (research questions)
    - Selected the Digital Libraries
    - Defined search string
    - Applied filtering process with include/exclude criteria

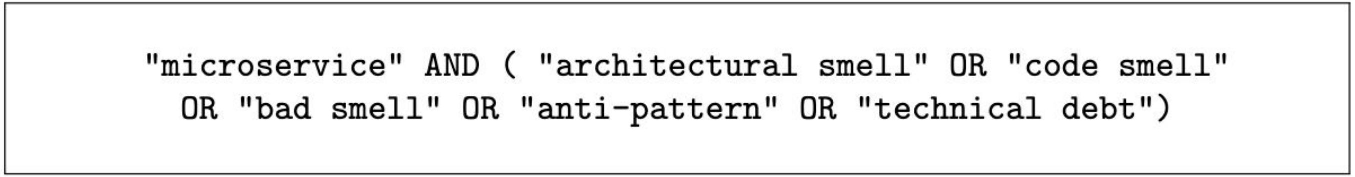
Research Questions

- **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?*

Digital Libraries and Query String



Database	Address
ACM Digital Library	https://dl.acm.org
Engineering Village	https://www.engineeringvillage.com
IEEE Explore	https://ieeexplore.ieee.org
Science Direct	https://www.sciencedirect.com
Scopus	http://scopus.com
Springer	http://link.springer.com



```
"microservice" AND ( "architectural smell" OR "code smell"
OR "bad smell" OR "anti-pattern" OR "technical debt")
```

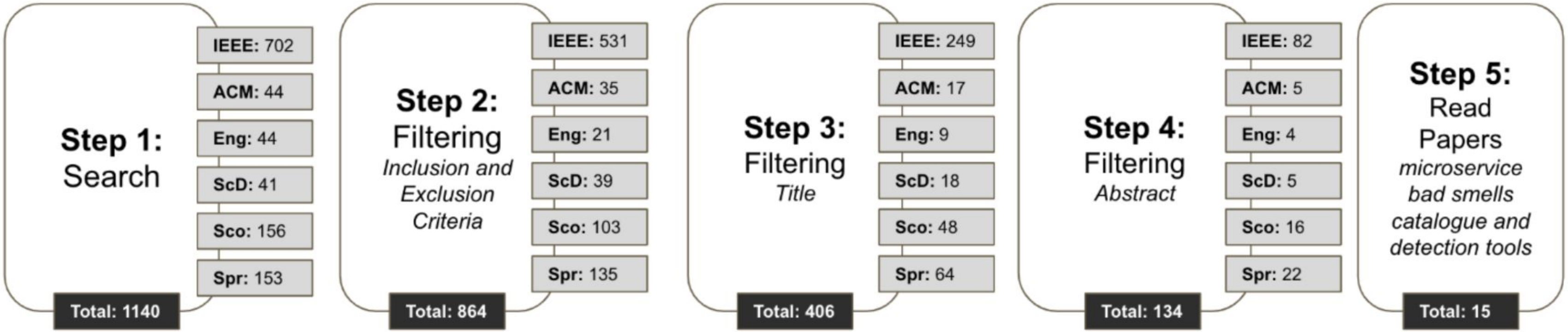
Filtering Process

- ## Inclusions Criteria
  - Papers published in Computer Science
  - Papers peer-reviewed and published in conferences, journals, book chapters or science magazines
  - Papers written in English and available in electronic format.
  - Papers published until mid-21 (Jun 1st).

- ## Exclusions Criteria
  - Papers with less than 5 pages
  - Thesis, dissertation, tutorial and courses
  - Websites, leaflets, and grey literature (blogs and non scientific magazines)

Research Execution

- **Step 1:** Applying the search string in all Digital Libraries (all metadata);
- **Step 2:** Applying the inclusion/exclusion criterias;
- **Step 3:** Title filtering;
- **Step 4:** Abstract filtering;
- **Step 5:** Reading remaining papers:
    - collecting the *microservice smell* catalog and related *detection tools/methods*.



Step 1: Search
IEEE: 702
ACM: 44
Eng: 44
ScD: 41
Sco: 156
Spr: 153
Total: 1140

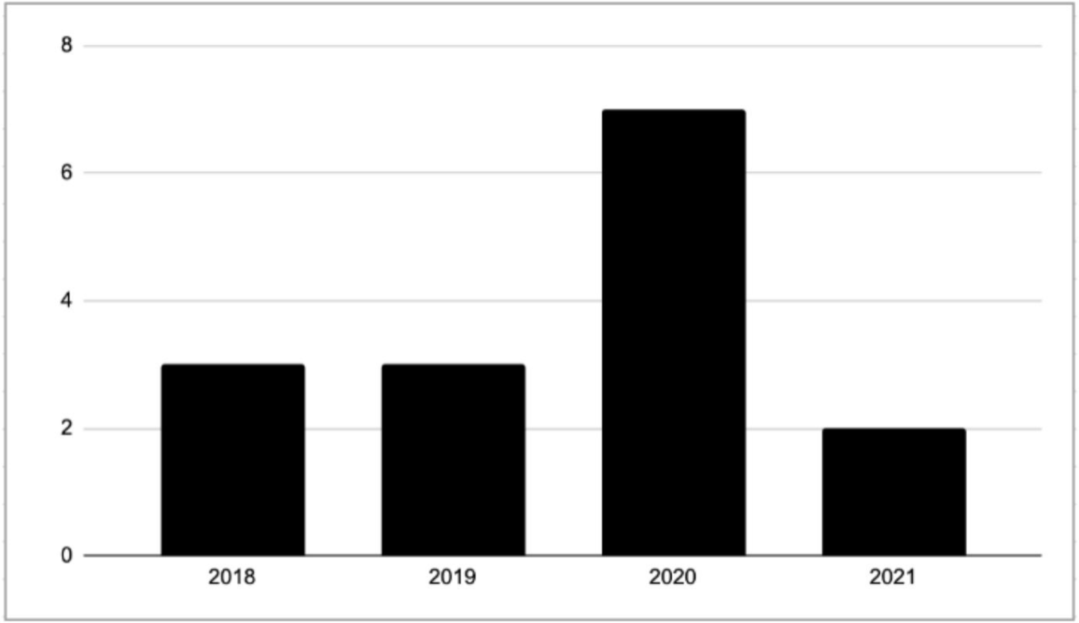
Step 2: Filtering
Inclusion and Exclusion Criteria
IEEE: 531
ACM: 35
Eng: 21
ScD: 39
Sco: 103
Spr: 135
Total: 864

Step 3: Filtering
Title
IEEE: 249
ACM: 17
Eng: 9
ScD: 18
Sco: 48
Spr: 64
Total: 406

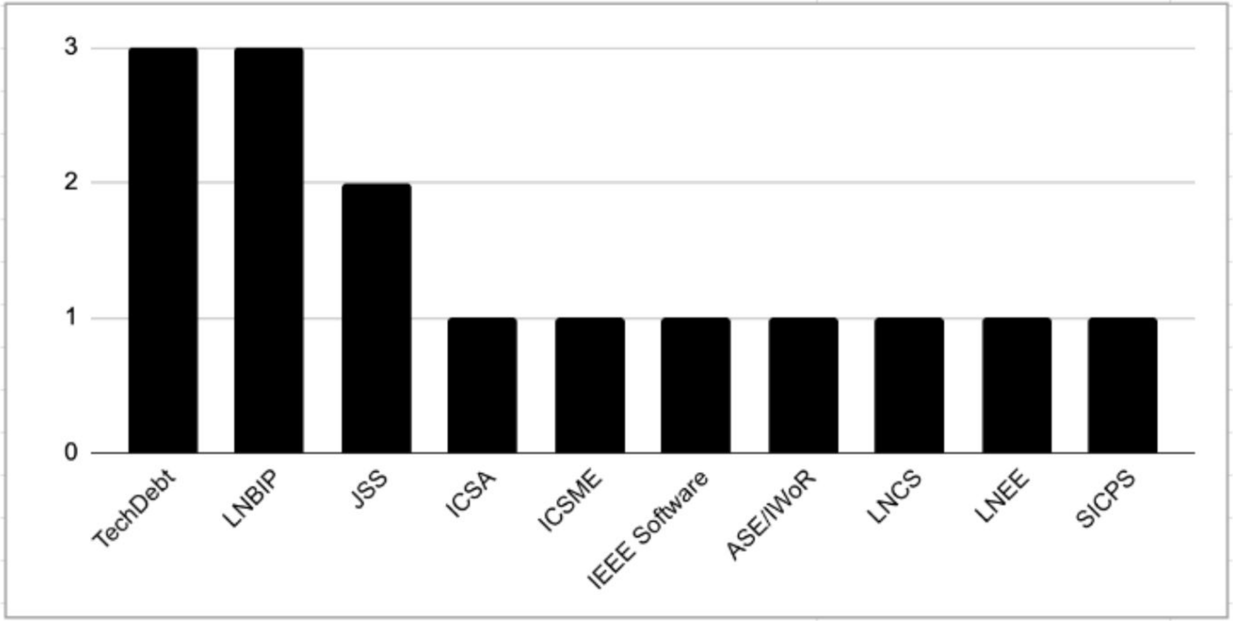
Step 4: Filtering
Abstract
IEEE: 82
ACM: 5
Eng: 4
ScD: 5
Sco: 16
Spr: 22
Total: 134

Step 5: Read Papers
microservice bad smells catalogue and detection tools
Total: 15

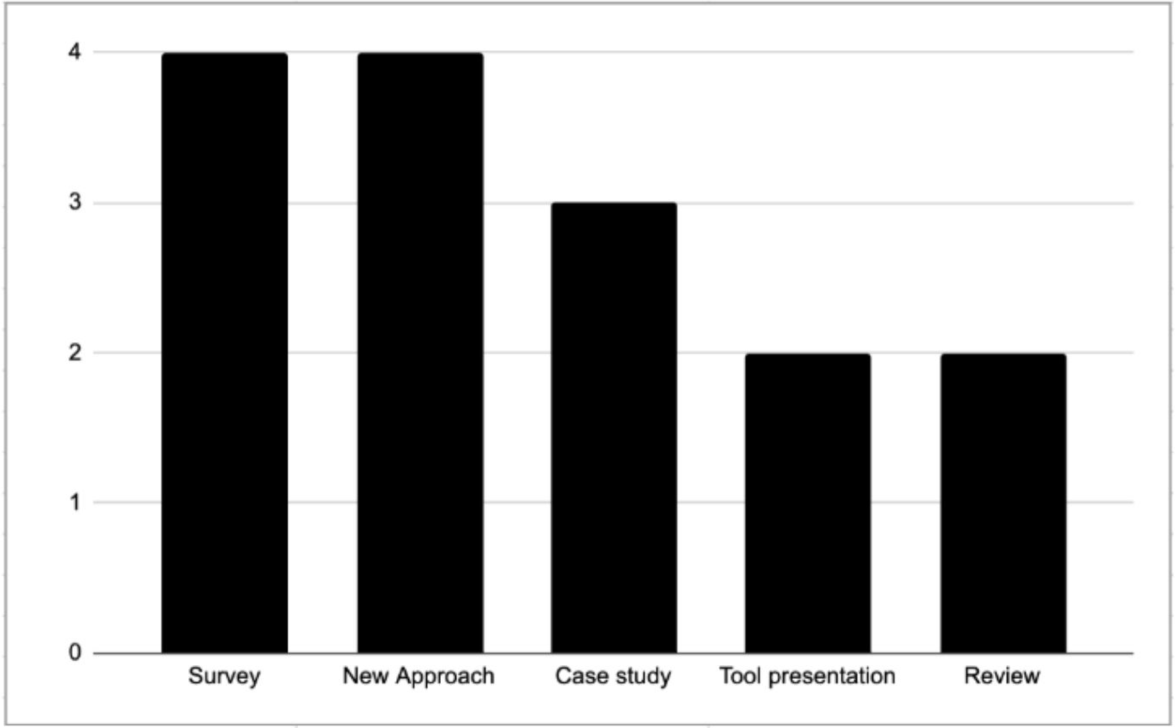
RQ1 (When): Research Over Time



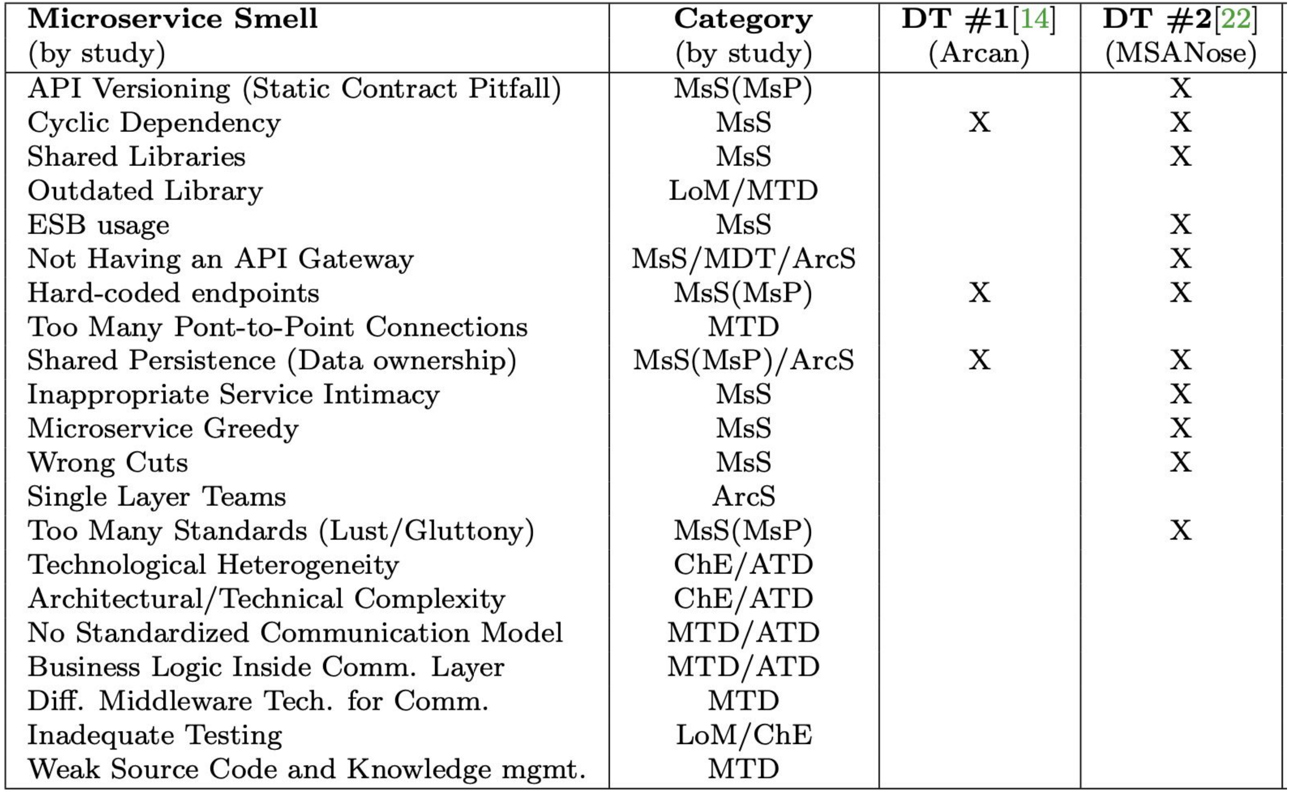
RQ2 (Where): Paper Venues



RQ3 (How): Type of Empirical Studies



RQ4 (What): Microservice Smells (21 selected)



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

*** All 77 microservice smells**
https://bit.ly/3ow5dSn

RQ5 (What): Microservice Smells Detection Tools

- **New Arcan** (Pigazzini et al., 2020):
  - Cyclic Dependencies
  - Hard-Coded Endpoints
  - Shared Persistence

- **Evaluation highlights:**
  - Evaluated manually the detection of 3 microservices smells using 30 open-source projects presented by Marques and Astudillo (2018).

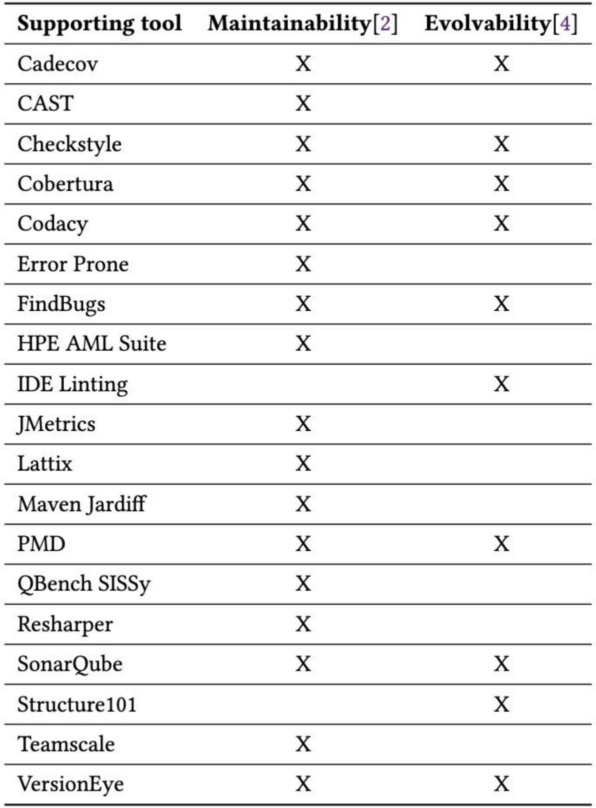
RQ5 (What): Microservice Smells Detection Tools

- **MSANose** (Walker et. al., 2021) :
  - API Versioning
  - Cyclic Dependency
  - ESB Usage
  - Hard-Coded Endpoints
  - Inappropriate Service Intimacy
  - Microservice Greedy
  - Not Having an API Gateway
  - Shared Libraries
  - Shared Persistency
  - Too Many Standards

- **Evaluation highlights:**
  - Evaluated the automatic detection by presenting 2 study cases that uses fictional microservices projects

RQ5 (What): Microservice Smells Detection Tools

- **Related Detection Tools:**
  - Bogner et. al. (2018, 2019)
  - 19 smells detection Tools
  - Supports microservices adoption on:
    - Symptoms of Low Maintainability
    - Challenges on Evolvability



Supporting tool	Maintainability[2]	Evolvability[4]
Cadecov	X	X
CAST	X	
Checkstyle	X	X
Cobertura	X	X
Codacy	X	X
Error Prone	X	
FindBugs	X	X
HPE AML Suite	X	
IDE Linting		X
JMetrics	X	
Lattix	X	
Maven Jardiff	X	
PMD	X	X
QBench SISSy	X	
Resharper	X	
SonarQube	X	X
Structure101		X
Teamscale	X	
VersionEye	X	X

Threats to Validity

- **External Validity**
  - It concerns the ability to generalize the results.
  - Major threat was during the identification of primary studies:
    - Digital libraries quality (Six)
    - Missed studies due to keywords (Synonyms)

Threats to Validity

- **Internal Validity**
  - Are results reliable enough?
  - Is this reproducible?
  - This threat was addressed by following:
    - Adopted a systematic protocol (Kichenham, 2007).
    - Three researchers have been involved.

Threats to Validity

- **Conclusion Validity**
    - Ability to draw correct conclusions from the study.
    - Potential threat is the reliability of data extraction:
        - Cross-discussions between reviewers.
        - In the event of disagreement of two, the third researcher act as an arbitrator.

Related Work

- **Taibi and Lenarduzzi (2018)**
  - Provided a catalog with definitions for 11 microservice-specific bad smells.
  - Result of 72 interviews of experienced microservice developers.

Related Work

- **Bogner et al. (2018)**
  - Performed a SLR focused on service-based antipatterns and bad smells.
  - Presented a holistic data model to describe and document service anti-patterns.
  - Reported 36 anti-patterns.

Related Work

- **Pereira-Vale et al. (2021)**
  - Performed a SLR on scientific papers (370) and gray literature (620).
  - Focused on investigating security on microservice-smells.
  - Provided a comprehensive catalog security solutions for microservice-based systems.

Conclusions and Future Work

- We presented a literature review
- Research on this topic has grow.
- We found 77 *microservice smells*.
- Small number of detection tools was reported (2)
- Future work:
    - Consolidate guidelines for microservices adoption.

lab-soft
software engineering laboratory

Software Engineering Approaches

Microservice Smells and Automated Detection Tools: A Systematic Review

Q & A

Denis Pinheiro, Johnatan Oliveira and Eduardo Figueiredo

Microservices 2022 - International Conference on Microservices