Microservices in a DevOps context

A review of "A Systematic Mapping Study on Microservices Architecture in DevOps"

Christoph Bühler
University of Applied Science of Eastern Switzerland Rapperswil OST
MSE Seminar "Design Science and Empirical Software Engineering"
Supervisor: Olaf Zimmermann
Semester: Fall 2020

Abstract

Short abstract about the goal of this paper. It should summarize the topics briefly and explain the sections very shortly.

1 Introduction

This paper is a review of the systematic mapping study "A Systematic Mapping Study on Microservices Architecture in DevOps" [WLS20]. The goal is to introduce the reader to the topic, explain the used methods in empirical software engineering, and give a critical review of the conducted study and the results.

TODO: Describe the used methods to analyze the paper
TODO: Describe and summarize the result

The remainder of the paper will introduce terms and principles that are used in the reviewed study, as well as topics that are needed for the understanding of the study and this review. Furthermore, a neutral summary of the reviewed study [WLS20] is provided. After the summary a critical review is conducted and then a conclusion is derived from the previous sections.

1.1 Design Science

Design Science has the main purpose of achieving knowledge and a general understanding about a domain [HMPR04]. Design Science contains several guidelines according to Alan R. Hevner (among other authors). Those guidelines are [HMPR04]:

- *Design as an artefact*: Design Science must produce an artifact
- *Problem relevance*: The objective is to develop solutions to relevant business problems
- *Design evaluation*: The utility of an artifact must be demonstrated with evaluation methods
- *Research contributions*: It must provide clear and verifiable contributions to the topic
- *Research rigor*: The research relies on rigorous methods in construction and evaluation of the model
- *Design as a search process*: The search for artifacts requires satisfying laws to be in place
- *Communication of research*: The targeted audience should be technology-based as well as management based

1.2 Empirical Software Engineering

Empirical Software Engineering (ESE) provides a solid base for discussion and methods for empirical research regarding software engineering topics. It has a strong emphasis on *empirical*.

1.2.1 Systematic Mapping Study (SMS).

A SMS is a defined method to gather, analyze, classify, and structure a field of interest. The result of an SMS allows readers and researchers to determine the coverage of the given field of interest [PFMM08]. The analysis focuses on frequency and topics for a field. It is a defined process in which the following steps take place [PFMM08]:

1. Define research questions (RQs) and topic
2. Define search query and parameter
3. Search for articles and publications in given databases
4. Analyze and screen the results (i.e., quality assessment and data analysing)
5. Classify and map the given articles

1.2.2 Systematic Literature Review (SLR).

A SLR is a method of ESE to systematically analyze and review a given topic. It uses methods to collect secondary data and critically reviews the given research study. The search for additional data can involve published as well as unpublished work on the subject [SWH19].

1.2.3 Systematic Gray Literature Review (SGLR).

A SGLR is essentially the same as an SLR with a very important difference: It does not only consider published and unpublished *peer reviewed* work, but also "Gray Literature". Gray literature is evidence and material that is not published in commercial and peer reviewed publications [Pae17]. In the context of computer science, gray literature can provide important statements and evidence towards topics that are more driven by businesses that by the academia.

1.3 Microservices and DevOps

Since the topic of the reviewed paper does conduct an SMS over "Microservices in DevOps", it is utterly important to define those terms so that any reader of this paper understands the base of the terms on which the conclusions are built upon.

1.3.1 Microservices. Microservices (sometimes referred to as "Microservice Architecture") is an application structural style. The style focuses on building several small services that cooperate together to create an application [Ric20]. Those services are often deployed on distributed systems. Most microservices adhere to the following principles [Ric20]:

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by small teams

The following image should provide a vague idea of how microservices are structured and organized [Ric20]:

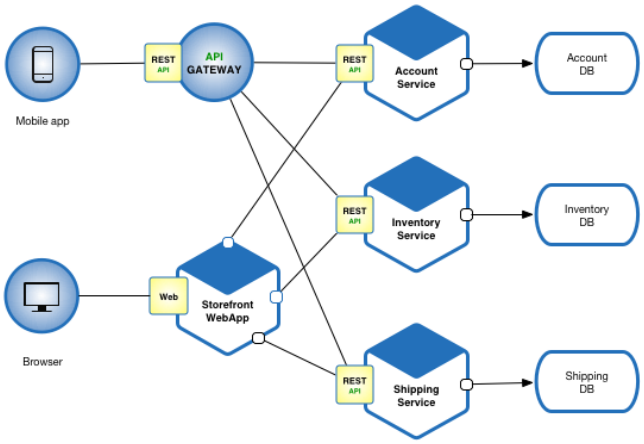

Figure 1. Example of a Microservice Architecture

1.3.2 DevOps. The term "DevOps" is a mash-up between "Development" and "Operations". Two strong and important terms in modern software engineering. The goal of DevOps is to automate end-to-end in software development and delivery [EGHS16]. It enables developers to deploy their software autonomously and - in smaller teams - without any specialized "Ops" department. It closes the gap between the classical software engineers and the operations team which both had their specific tasks. DevOps provides the means and tools to provide an agile method of software engineering with *continuous integration* and *continuous deployment* [Hü12].

2 The reviewed mapping study

This section does summarize the goal, idea and result of the paper. This part will be a neutral summary. Give an abstract about the paper of [WLS20].

2.1 Motivation

Describe and analyze the motivation of the researcher group that made the SMS.

2.2 Methodology

Analyze and summarize their methods. Describe the process they used and the additional changes they made to conduct their SMS.

2.3 Research Questions

Give a brief summary about their questions that they want to answer in the study.

2.4 Results

Summary of the results of the paper.

2.5 Discussion

Summary of the given "discussion" in the paper.

2.6 Threats to validity

Summary of their threads to validity.

2.7 Conclusion

Summary of the given conclusion.

References

[EGHS16] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. Devops. *IEEE Software*, 33(3):94–100, 2016.

[HMPR04] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Quarterly*, 28(1):75–105, 2004.

[Hü12] Michael Hüttermann. *DevOps for Developers*. Apress, 2012.

[Pae17] Arsenio Paez. Gray literature: An important resource in systematic reviews. *Journal of Evidence-Based Medicine*, 10(3):233–240, 2017.

[PFMM08] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 12:1–10, 2008.

[Ric20] Chris Richardson. What are microservices? https://microservices.io/, 2020. Accessed: 2020-10-25.

[SWH19] Andy P. Siddaway, Alex M. Wood, and Larry V. Hedges. How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses. *Annual Review of Psychology*, 70(1):747–770, 2019. PMID: 30089228.

[WLS20] Muhammad Waseem, Peng Liang, and Mojtaba Shahin. A systematic mapping study on microservices architecture in devops. *Journal of Systems and Software*, 170:110798, 08 2020.