Automated Microservice Identification: an Approach to Decomposition into a Modular Monolith Architecture

University of Turku
Department of Computing
Master of Science (Tech) Thesis
March 2024
Florian Dejonckheere

UNIVERSITY OF TURKU
Department of Computing

Florian Dejonckheere: Automated Microservice Identification: an Approach to Decomposition into a Modular Monolith Architecture

Master of Science (Tech) Thesis, 4 p., 4 app. p.
Department of Computing
March 2024

The modular monolith architecture emerged in recent years as the harmonization of the monolithic and microservices architectures. The paradigm offers a compromise between modularity, flexibility, and scalability. Many monolithic applications are being migrated to modular monoliths or microservices entirely, to satisfy increasingly complex and volatile business requirements. This process is labour-intensive, slow, and may take months to years for larger codebases. Modularization of a codebase typically requires the developer to have an intimate knowledge of both the application code and domain.

In this thesis, we investigate the modular monolith software architecture, and how modules are typically determined as part of the modularization efforts. We propose an automated solution based on dependency analysis and machine learning algorithms to aid in the identification of module boundaries, and evaluate its effectiveness using a case study. We discuss the results and draw conclusions about the propsed solution.

Keywords: software architecture, monolith, microservices, modular monolith

Contents

List of Figures

List of Tables

List of Acronyms

DSRM Design Science Research Methodology

DSRP Design Science Research Process

1. Introduction

1.1. Scope and goal

This research is centered around three research questions:

Research Question 1: Which challenges and opportunities arise when considering adoption of the modular monolith architecture in an existing codebase?

Research Question 2: What are the existing approaches and tools for automated modularization of monolith codebases?

Research Question 3: How can *(automated technology)* effectively identify optimal module boundaries in a modular monolith architecture?

To answer the first research question, we will first define the modular monolith architecture, and examine what sets it apart from monolithic and microservices architectures. Then, we will proceed to investigate the merits and drawbacks of the software architecture when applied to an existing codebase.

For the second research question, we will enumerate the existing technologies to aid modularization of monolithic codebases, and choose one automated technology for further examination. *(Automated technology)* will then be implemented for a given use case, and compared to manual modularization efforts in terms of accuracy, efficiency, development velocity. This comparison will help us to answer the third research question.

The goal of this research can be summarized as follows:

1. Investigate the merits and drawbacks of the modular monolith architecture
2. Investigate the use of automated technologies to modularize a monolithic architecture

The proposed solution will add value to the field of software engineering, and will be able to be used as a base for future improvements regarding automated modularization of monolith codebases.

1.2. Motivation

1.3. Methodology

A literature review is conducted to answer the first and second research question. For the first research question, the study aims to find a definition of the modular monolith architecture, and to list the advantages and disadvantages of the architecture based on existing literature. For the second

research question, the state of the art in automated modularization technologies is reviewed and summarized.

The third research question is answered by choosing the most appropriate automated technology, and implementing it for a given use case. The implementation is then evaluated based on quantitative and qualitative metrics, and compared to manual modularization efforts.

Finally, the findings are summarized, and an outlook on future work is given.

For the case study, a Design Science Research Methodology (DSRM) is adopted, which is a research paradigm for information systems research focused at creating and evaluating artifacts. In particular, the research and design of the proposed solution follows the six-step Design Science Research Process (DSRP) model [1]. Their model is based on prior research and is designed to guide researchers through the process of analysis, creation, and evaluation of artifacts.

The six steps of the process are:

1. **Problem identification and motivation**: Research problem statement and justification for existence of a solution.
2. **Objectives of a solution**: Definition of the objectives, derived from the problem statement.
3. **Design and development**: Creation of the artifact.
4. **Demonstration**: Usage of the artifact to demonstrate its effectiveness in solving the problem.
5. **Evaluation**: Observation and measurement of how well the artifact supports a solution to the problem.
6. **Communication**: Transfer of knowledge about the artifact and the problem solution to the relevant audience.

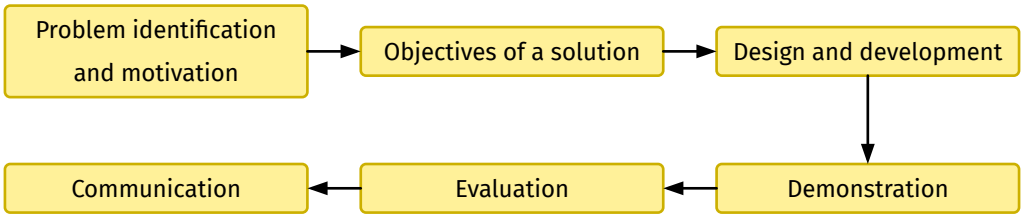

Figure 1: Design Science Research Process (DSRP)

The process is structured sequentially, however the authors suggests that researchers may proceed in a non-linear fashion, and start or stop at any step, depending on the context and requirements of the research.

In this thesis specifically, the DSRP is used to guide the design and development of the automated modularization technology, with a particular focus on the design and development, demonstration, and evaluation steps.

1.4. Outline

The thesis is divided into three parts.

The first part comprises the background and related work. In Chapter 1, the scope and goal of the research is defined, and the research questions are formulated. The stakeholders are identified, and the methodology is explained. Chapter 2 introduces the reader to the research background and necessary concepts. In Chapter 3, the existing literature is reviewed, and the state of the art is presented.

The second part of the thesis, starting with Chapter 4, is dedicated to the first research question. The modular monolith architecture is defined, and its merits and drawbacks are discussed.

The third part aims to solve the second and third research question. Chapter 5 gives an introduction into the automated modularization of monolith codebases, listing the existing technologies. It then continues to focus on one automated technology, *(automated technology)*, and explains its implementation. Chapter 7 applies *(automated technology)* on a given case study, and compares it to manual modularization efforts.

Finally, Chapter 8 summarizes the findings, and gives an outlook on future work.

2. Background

2.1. Monolith architecture

2.2. Modular programming

2.3. Microservice architecture

3. Related work

4. Modular monolith architecture

4.1. Background

4.2. Challenges and opportunities

4.3. Modularization

5. Automated modularization

In this chapter, we will investigate the state of the art in automated technologies for modularization of monolith codebases. We will answer the following research question:

Research Question 2: What are the existing approaches and tools for automated modularization of monolith codebases?

Using a systematic literature review, we will identify and categorize existing literature regarding automated modularization of monolith codebases. We will also provide a brief overview of the most relevant approaches and tools.

A systematic literature review is used to identify, evaluate and interpret research literature for a given topic area, or research question [2]. The systematic nature of systematic literature reviews reduce bias through a well-defined sequence of steps to identify and classify existing literature. Studies directly researching the topic area are called *primary* studies, systematic studies aggregating and summarizing primary studies are called *secondary* studies.

The literature review is conducted using a three-step protocol as defined by B. Kitchenham and S. Charters, [2]:

	Step	**Activity**
1	Plan	Identify the need for the review, specifying the research questions, and developing a review protocol
2	Conduct	Identification and selection of literature, data extraction and synthesis
3	Report	Evaluation and reporting of the results

6. Proposed solution

7. Case study

7.1. Background

7.2. Analysis

7.3. Evaluation and results

7.4. Discussion

8. Conclusion

8.1. Future work

References

[1] K. Peffers, M. Rothenberger, T. Tuunanen, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research”, vol. 24, no. 3.

[2] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering”.