ENSURING MICROSERVICE ARCHITECTURAL CONSISTENCY THROUGH AUTOMATED DETECTION OF DRIFTS AND EROSIONS

A.Z Jabir

229336D

Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

April 2024

ENSURING MICROSERVICE ARCHITECTURAL CONSISTENCY THROUGH AUTOMATED DETECTION OF DRIFTS AND EROSIONS

A.Z Jabir

229336D

Thesis submitted in partial fulfillment of the requirements for the degree
Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

April 2024

DECLARATION

I declare that this is my own work and this Thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date: 11.07.2024

The supervisor should certify the Thesis with the following declaration.

The above candidate has carried out research for the Master of Science in Computer Science Thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Sunimal Rathnayake

Signature of the Supervisor: Date: 12/07/2024

DEDICATION

It is with great honor and a sense of deep appreciation that I dedicate this project to my parents. Their unwavering love, support, and encouragement have been a source of strength and inspiration throughout my entire life and have made this achievement possible.

ACKNOWLEDGEMENT

First and foremost, I offer my sincerest gratitude to the Almighty for providing me with the strength, knowledge, and perseverance to complete this challenging yet rewarding journey.

I would also like to extend my sincere gratitude to my project supervisor, Dr. Sunimal Rathnayake, who guided me through every step of this journey with his expertise, patience, and support. His invaluable insights and encouragement have been instrumental in the successful completion of this research.

Finally, I would like to extend my heartfelt thanks to my colleagues, friends, and family who have supported me emotionally and provided me with the motivation to see this project through to its completion.

ABSTRACT

In the shift from monolithic to microservices architecture within the software development industry, the benefits of improved system resilience, agility, and decentralized data management are clear. Yet, this also introduces significant challenges in maintaining architectural consistency. The decentralized and heterogeneous nature of microservices imposes challenges in managing and maintaining the architecture integrity, which can lead to architectural degradation. This poses maintenance challenges and risks project success. The role of architectural integrity is crucial across the software lifecycle, affecting the development, deployment, and maintenance phases. Inconsistencies within the architecture can confuse the developers and make existing documentation outdated. Moreover, manual or semi-automated checks for architectural conformance are not only time-consuming but also prone to errors, highlighting the need for automated solutions. Recent studies have discovered the need for mechanisms that can proactively address architectural problems in microservices. The findings vouch that software architecture reconstruction is useful for addressing these architectural issues, with static analysis showing significant promise for the reconstruction of software architecture. However, there remains a notable gap in the utilization of language-neutral artifacts for automated reasoning and generating reports about the integrity of microservices architectures. This research presents a novel approach to ensure the consistency of microservice architecture by automating the detection of drifts and erosions. By adopting Model-Driven Engineering principles, which is widely in practice in the industry, our approach uses a textual technique to create a metamodel for each microservice and its communications, achieving technology neutrality. The process begins with a static analysis of the metamodel files and the intended architecture, followed by the construction of service dependency graphs to compare the intended and modified architectures. This systematic approach allows for the accurate identification of deviations, offering a preventive way to address architectural inconsistencies. To assess the effectiveness of this method, we evaluated it using the popular microservices demo application, OnlineBoutique. The findings demonstrate that our approach can precisely identify all microservices within the application and detect violations. These results suggest that our method is not only effective but also efficient in proactively identifying architectural drifts and erosions, contributing significantly to the maintenance of microservices architecture.

Keywords: Microservice Architecture, Software Architecture Degradation, Software Architecture Reconstruction, Architecture Conformance Checking , Model-Driven Engineering

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
ACC	Architecture Conformance Checking
CI/CD	Continuous Integration and Continuous Delivery
CLI	Command-Line Interface
FN	False Negatives
FP	False Positives
IDE	Integrated Development Environment
MDE	Model-Driven Engineering
SAR	Software Architecture Reconstruction
TP	True Positives

LIST OF APPENDICES