REVERSE ENGINEERING MICROSERVICES

REVERSE ENGINEERING MICROSERVICES FOR ENHANCED INSIGHTS

BY

MUHAMMAD WAQAR UL HASSAN AWAN, M.Eng.

A Report

submitted to the Computing and Software

and the School of Graduate Studies

of McMaster University

in partial fulfilment of the requirements

for the degree of

Masters of Engineering

Masters of Engineering (2024) McMaster University

(Computing and Software) Hamilton, Ontario, Canada

TITLE:	Reverse Engineering Microservices for Enhanced Insights
AUTHOR:	Muhammad Waqar Ul Hassan Awan M.Eng. Computing and Software, McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Sébastien Mosser
NUMBER OF PAGES:	xiii, 15

Lay Abstract

A lay abstract of not more 150 words must be included explaining the key goals and contributions of the thesis in lay terms that is accessible to the general public.

Abstract

Abstract here (no more than 300 words)

Your Dedication

Optional second line

Acknowledgements

Acknowledgements go here.

Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Notation

$A \leq B$	A is less than or equal to B

Definitions

Challenge	With respect to video games, a challenge is a set of goals presented to the player that they are tasks with completing; challenges can test a variety of player skills, including accuracy, logical reasoning, and creative problem solving

Abbreviations

SST	Single source of truth
UDS	Unified data source

RE	Reverse engineering
SRE	Software reverse engineering
DSL	Domain-specific language

Declaration of Academic Achievement

The student will declare his/her research contribution and, as appropriate, those of colleagues or other contributors to the contents of the thesis.

Chapter 1

Introduction

Every thesis needs an introductory chapter

While you're here, you need to go into `definitions.tex` to set all the information needed for the front matter (e.g. title, author) and page header/footer.

You will also find the School of Graduate Studies' preparation guide (August 2021) for theses and reports. I would give it a quick read so you know what's expected.

Chapter 2

Background

Every thesis needs an Background chapter

While you're here, you need to go into `definitions.tex` to set all the information needed for the front matter (e.g. title, author) and page header/footer.

You will also find the School of Graduate Studies' preparation guide (August 2021) for theses and reports. I would give it a quick read so you know what's expected.

Chapter 3

Vision and Technical Strategy

The development of a large software system is a very critical process and there are a lot of parts that are involved in a project that needs to follow key software development paradigms so that it is easy to maintain. Its not just about writing code, its about creating a system that is bug free, easy to maintain, scalable and efficient.

Since an already built large software system is quite complex, so understanding this system requires certain strategies and tools. (Folmer et al., 2005) discusses about the useability issues in the software systems post-development and they require significant architectural changes. In order to deal with such architectural changes, understanding of the whole system is required and if the system is large enough, major resources are spent to fix even minor issues.

(Canfora and Cimitile, 2001) states that several surveys indicate that software maintenance consumes 60% to 80% of the total life cycle costs. Also the maintenance costs are largely due to enhancement (often 75-80%), rather than corrections. So a framework should be developed that could solve these issues by doing reverse engineering on the software systems and showing the importants aspects of the software

architecture in order to make it easy to understand and perform the required tasks.

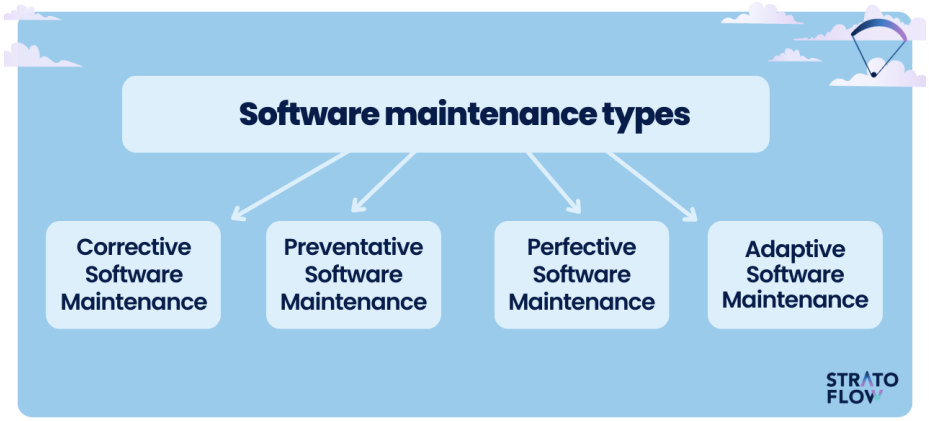

Figure 3.1: Software Maintenance Types (adapted from Stratoflow (2025))

3.1 Vision

Keeping in mind the above discussed issues, there is a need for an approach in which the software system could be analyzed, undergo processing and produce useful informations that can be used to provide enhanced insights about the project. The vision is to use a framework consisting of 3 components: Probes, Single Source of Truth (SST) and Visualizer. All three components will be discussed in detail in section 3.2.

In this report, we will discuss,implement and validate this framework by extrac ting static information from a project. In the future, this can be implemented/integrated with the CI/CD pipeline and provide dynamic information from the projects. The test project used in this report is Java spring framework based project called petclinic. Read more about this project from spring petclinic microservices github repository (Spring-Team, 2025).

Moreover, our approach will be mainly focused on the unified data source (UDS)

approach. (Statistics-Easily, 2025) states that Unified Data refers to the integration and consolidation of data from various sources into a single, cohesive framework. This approach allows organizations to streamline their data management processes, ensuring that all data is accessible and usable across different departments and applications. By unifying data, businesses can eliminate silos, reduce redundancy, and enhance the overall quality of their data analytics efforts. It means that consistent, up-to-date and valid data will be available using the UDS technique.

3.2 Technical Strategy

3.2.1 Framework Components

The framework contains three main components.

- Probes
- Single Source of Truth (SST)
- Visualizer

3.2.2 Probes

The first component of the framework consists of **probes**. Probes represent distinct informational artifacts that are systematically extracted from software systems to provide insights and actionable data. In the context of this report, we have identified and extracted specific pieces of information that are detailed in the below subsections. These probes serve as the foundational elements for gathering critical data points that enable analysis and decision-making.

Looking forward, this concept can be expanded to more complex and targeted informational needs. By refining the scope and nature of the probes, we can tailor the probes according to our needs and to capture more refined data, addressing evolving requirements and insights that drive meaningful outcomes. This flexibility ensures that as the systems grow or change, the framework remains relevant and capable of producing deeper, more impactful information.

- **Authors Contributions**

  What it is?, why we need it, what it solved? type of data extracted...

- **File Authors**

  What it is?, why we need it, what it solved? type of data extracted...

- **Microservices Endpoints**

  What it is?, why we need it, what it solved? type of data extracted...

3.2.3 Single Source of Truth (SST)

3.2.4 Visualizer

If you need to use quotes, type it “like this”.

3.3 Referencing

These are some sample references to GAMYGDALA (Popescu et al., 2014) from the `references.bib` file and state effects of cognition (Hudlicka, 2002) from the `references_another.bib` file. These references are not in the same .bib file.

3.4 Figures

This is a single image figure (Figure 3.2):

Figure 3.2: This is a single figure environment

This is a multi-image figure with a top (Figure 3.3a) and bottom (Figure 3.3b) aligned subfigures:

3.5 Tables

Here is a sample table (Table 3.1):

A	$\longleftrightarrow$	B
C	$\longleftrightarrow$	D

Table 3.1: A sample table

3.5.1 Long Tables

A sample long table is shown in Appendix B.

3.6 Equations

Here is a sample equation (Equation 3.6.1):

$$y = mx + b \tag{3.6.1}$$

(a) Figure 1

(b) Figure 2

Figure 3.3: A Multi-Figure Environment

Chapter 4

Conclusion

Every thesis also needs a concluding chapter

Appendix A

Your Appendix

Your appendix goes here.

Appendix B

Long Tables

This appendix demonstrates the use of a long table that spans multiple pages.

Col A	**Col B**	**Col C**	**Col D**
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D

Continued on the next page

Continued from previous page

Col A	**Col B**	**Col C**	**Col D**
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D
A	B	C	D

Bibliography

Gerardo Canfora and Aniello Cimitile. 2001. Software Maintenance. *Handbook of Software Engineering and Knowledge Engineering* 1 (01 2001). https://doi.org/10.1142/9789812389718_0005

Eelke Folmer, Jilles van Gurp, and Jan Bosch. 2005. Engineering Human Computer Interaction and Interactive Systems. In *Engineering Human Computer Interaction and Interactive Systems*, Rémi Bastide, Philippe Palanque, and Jörg Roth (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 38–58.

Eva Hudlicka. 2002. This time with feeling: Integrated model of trait and state effects on cognition and behavior. *Applied Artificial Intelligence* 16, 7-8 (2002), 611–641.

Adrian Popescu, Joost Broekens, and Maarten van Someren. 2014. GAMYGDALA: An emotion engine for games. *IEEE Transactions on Affective Computing* 5, 1 (2014), 32–44.

Spring-Team. 2025. Spring Petclinic Microservices. https://github.com/spring-petclinic/spring-petclinic-microservices. Accessed: 2025-01-11.

Statistics-Easily. 2025. What is Unified Data? Understanding its Importance. https://statisticseasily.com/glossario/

what-is-unified-data-understanding-its-importance/?utm_source=chatgpt.com#google_vignette Accessed: 2025-01-11.

Stratoflow. 2025. The Software Maintenance Process: What You Need to Know. https://stratoflow.com/software-maintenance-process/ Accessed: 2025-01-11.