

# Individual - Architecture Recovery

(Nicklas Jeppesen)

Course code: KSSOARC2KU

May 9, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Real work life motivation . . . . .	2
<b>2</b>	<b>Methodology</b>	<b>4</b>
2.1	Tools . . . . .	4
2.1.1	Creating the sequence diagram . . . . .	4
<b>3</b>	<b>Results</b>	<b>5</b>
3.1	what the code provide . . . . .	5
3.2	workflow . . . . .	5
<b>4</b>	<b>Discussion</b>	<b>6</b>
4.1	Challenges that I did not expect when starting . . . . .	6
4.1.1	Filename not equal module name . . . . .	6
4.2	General limitation of Static analys . . . . .	6
4.3	Limitation for this codebase . . . . .	6
4.3.1	import from new line . . . . .	7
4.3.2	import is not starting of the file . . . . .	7
4.4	Further development . . . . .	7
<b>5</b>	<b>Time allocation</b>	<b>9</b>
<b>6</b>	<b>Github</b>	<b>10</b>
<b>A</b>	<b>Appendix Figures</b>	<b>11</b>

# Chapter 1

## Introduction

The goal for this project is create a Static Analysis of the code base: <https://github.com/zeeguu/API>. Where the goal should be to be able to take a given function and create a sequence diagram from the methods.

### 1.1 Real work life motivation

Why this project? why is this usefull and and why static analysis instead of dynamic analysis?

Motivation from this project, came by when I got a new job where I had to maintain an existing relative large code base, where the code base was part of a eco-system, of multiple systems, where the system could get a request from another system,

and in the process of handling thar request performing other request to other system, and to then return a response. There didnt exists documentation that explain every single workflow, so to maintain and finding bugs became very hard. The idea was then, what if a system, could crossing systems create a sequence diagram, to virsualize the dependency of the other system. Now micro-services and monolith distributed system has become more popular, so this idea might be usefull for micro-services or monolith distributed systems. This report take the first step on this journey, and try to create a static analyzer which can analyze a single request method to create a sequence diagram from it, in the programming language python.

### Why static analyze and not dynamic analyzer?

Its well known static analyzer have serveral limitation that danamic analyser dont have, for this it can be difficult to determine dead code, wheter it used

or not, also if you have implemented the factory patterns in your code, it is hard to say, if you have code that is never executed. So Dynamic code seems to be a better option. Now in my real work case. I had two issues to.

first, I was not able to run the program local on my machine, because it needed specific installation files and licence, it was not possible. Even then, we had servers where the system was installed, but then the second problem occurred. It needed specific knowledge to trigger a given request in the system, from another system, that I didn't have, and required a colleague. These are the motivations for using a static analyzer, that can analyze not running code, to do reverse engineering to get an overview of the system. lastly, dynamic analysis is easier in some languages than others like it is easy in python, but more challenging in languages like java. Where static analysis does not depend on built-in functions in the languages.

# Chapter 2

## Methodology

### 2.1 Tools

to creat this project. First this project used to provided from to colab pynb file. Thereafter it used the knowledge about dynamic analysis, and static and dynamic analyzer. To actual analyse the non running code, rich set of regex function was needed to be developered in the progress, for the following reason:

- Used to extract the imported modules in a file, first inspired by the provided code, but later modified.
- regex is also used to get a list of all functions definiton in a file, used later to figure out what module/file a function call in a function is comming from.
- Used to extract a function body from a string file, and a function name, to analyze.
- regex pattern matching is also used to get a list of function called in a function, for further analysis.

#### 2.1.1 Creating the sequence diagram

For drawing the sequence diagram, I use a third party tool called mermaid-cli, my code create a string that this tool can transform into a sequence diagram.

# Chapter 3

## Results

### 3.1 what the code provide

By providing a filepath as string and function name to the system, it output a png file naming the provided function name containing the the sequence diagram of function. The sequence diagram module names as object, that is because sometimes a file can import a function from a module and not a class, so safe choice is to name the object as the module, and then horientizal line as the function.

### 3.2 workflow

So how does the work really work? In the `Static_analysis.ipynb` file, I have defined, a function called `masterFunction`, that take a function name and a path to the python file. First the `masterFunction` get the content of the python file, thereafter it get the function body as a string. thereafter it a list of called functions in the input function,

Then it get a list of all imported modules that are used in the file. Thereafter it get a list of strings in the format `<modulename>;<function_nane>`, where all the functions name are the called function in the methods, that are defined inside `zeeguu` project. Thereafter it return this list. After that, a function named `"createMermaid"` take that list and conert into a string, that `mermaid-cli` can understand at transform into a sequence diagram. Lastly, with the `mermaid-cli` string, function `"generate_sequence_diagram"` are called to transform this string to a sequence diagram, and then a png file is created with the sequence diagram, with the filename of the start function.

# Chapter 4

## Discussion

### 4.1 Challenges that I did not expect when starting

There was several surprises/challenges I faces, when started on this project, one that is worth of mention is:

#### 4.1.1 Filename not equal module name

One challenge, I didnt expect, when I started this project, was that the imported module, might not be equal to the filename. so to be able to extract a filepath from its module name, became very challenges, and I had the check the init.py file, to search for the correct name, and then again try to find the file name.

### 4.2 General limitation of Static analys

### 4.3 Limitation for this codebase

This project has in some way, ended by becomming a kind of a parser, that can detext code snippet, so to make this project not explored, I had to limit some kind of general code snippet, that python normal can accept, but this code cannot, and these limitation are the following:

### 4.3.1 import from new line

The example below, shows an example, that my project cannot handle, because of the use of parantes and new line to import modules. This is one of the limitations I have to live with for now.

---

```
#file: zeeguu/api/endpoints/activity_tracking.py
from
    zeeguu.core.user_activity_hooks.article_interaction_hooks
import (
    distill_article_interactions,
)
```

---

To fix this issue, I change the code the the import module(s) is on a single line, which my code can handle. as shown below.

---

```
#file: zeeguu/api/endpoints/activity_tracking.py
from <path> import distill_article_interactions
```

---

### 4.3.2 import is not starting of the file

another issue I found, was if module is imported in the middle of the file, in a block, then the analyzer cannot register that, like the following issue:

---

```
#file: zeeguu/api/endpoints/activity_tracking.py, line 41
if request.form.get("event") == "AUDIO_EXP":
    from zeeguu.core.emailer.zeeguu_mailer import ZeeguuMailer

    ZeeguuMailer.notify_audio_experiment(request.form,
        flask.g.user)
```

---

if the if statement becomes true, then this function call wont be registered, because the import module, was not registered. i change this file too, so I added the from-import code in the top of the file, with the rest, and then, it was registered.

## 4.4 Further development

First thing is to correct the already found issues in the discussion section, This would be the first task. Second, With the known knowledge, it could be great to expend this project, so it could have access to multiple code bases/microservices so it would be able to create sequences diagrams for a



kubernetes/docker swarm network. third, to be able to create a deeper code analyzer, that either can analyze multiple code bases, or go deeper, until other methods call, one solution could be to create a tree structure, where the root node, could be the first method, and each sub node, could be method calls in the methods, that calls methods either from the code base, or other code base we want to examine, so we don't get system code into our tree. There from that tree, it could be possible to draw a sequence diagram, where each node could hold information about, codebase, module path, etc.

# Chapter 5

## Time allocation

Coding: 80%

Writing report: 10%

# Chapter 6

## Github

`https://github.com/nicklasjeppesen/software\_individual\_report`

# Appendix A

## Appendix Figures

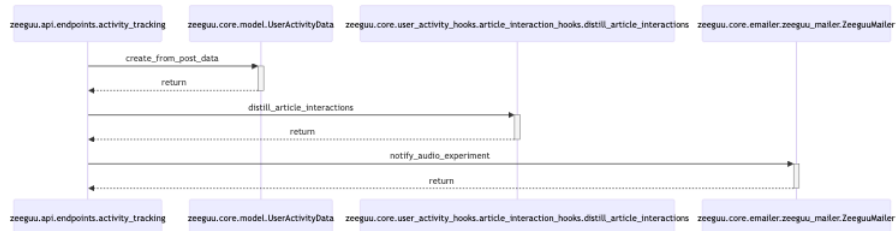


Figure A.1: Methods call for method "upload\_user\_activity\_data" in "activity\_tracking.py"

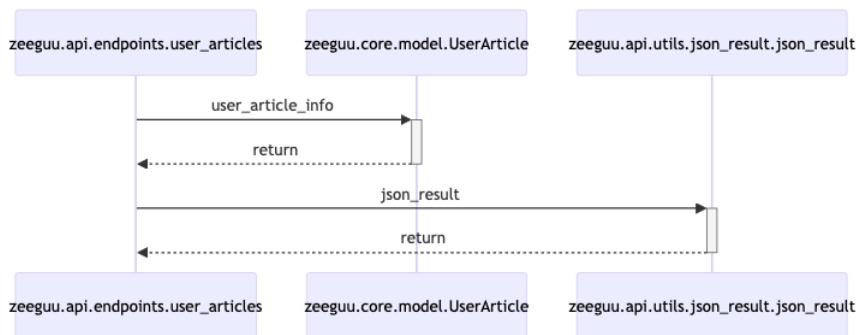


Figure A.2: Methods call for method `user_articles_recommended` in file: `user_articles.py`

# Bibliography

- [1] Google colab code, url: [https://colab.research.google.com/drive/1oe\\_TV7936Zmmzbbgq8rzqFpxYPX7SQHP#scrollTo=0ruTtX88Tb-w](https://colab.research.google.com/drive/1oe_TV7936Zmmzbbgq8rzqFpxYPX7SQHP#scrollTo=0ruTtX88Tb-w), (accessed: 09.05.2024).
- [2] Mermaid-cli: <https://github.com/mermaid-js/mermaid-cli>, (accessed: 09.05.2024).
- [3] reconstruction part 1: [https://github.com/mircealungu/reconstruction/blob/master/1\\_Introduction.md](https://github.com/mircealungu/reconstruction/blob/master/1_Introduction.md), (accessed: 09.05.2024).
- [4] reconstruction part 4, dynamic analysis: [https://github.com/mircealungu/reconstruction/blob/master/4\\_Dynamic\\_Analysis.md](https://github.com/mircealungu/reconstruction/blob/master/4_Dynamic_Analysis.md), (accessed: 09.05.2024).
- [5] Dynamic analysis, older material, but have other great examples of issues with static analysis: [https://github.com/mircealungu/reconstruction/blob/master/materials/Dynamic\\_Analysis.md](https://github.com/mircealungu/reconstruction/blob/master/materials/Dynamic_Analysis.md), (accessed: 09.05.2024).