Modular monolith

Jessie Liauw A Fong

May 9, 2019

Contents

Chapter 1

Preface

I am Jessie Liauw A Fong, 20 years old. I was born in Amsterdam but moved to Zaandam and still live there. I started programming in 2010 when I was in the first class of middle school. After a year of programming my interest stagnated but in 2015 I chose to begin the study software engineering and I immediately felt that passion again and I haven't lost that pasion since. In the end of 2015 I started my first software engineering job at The EsportsWall. This was a voluntary job because I did not have enough skills to get paid. 3 months later I did to start my internship at Endouble. I worked at Endouble for 1 year. 5 months as an intern and 7 months as an part time employee.

After I finished my internship at Endouble I started by own company JCB Development. Where I build high-end websites.

At the end of my time at Endouble I started a new parttime internship at Ximedes where I learned a lot about infrastructure. This is also the company I met my now co-worker Erik Schouten. He worked at CargoLedger and that is also where I work now. In the beginning of September 2018 I started a new company together with Stijn Claessen and Siebe Goos called EFFE Planning. This is also the company I will do my thesis in.

This thus means I know the ins and outs of the company.

Chapter 2

Summary

Chapter 3

Introduction

3.1 Motive

EFFE as a company uses the SaaS model in order to comply to it's expected growth. The basic SaaS model includes the basic application or MVP. This is in order to keep the application as abstract as possible. So that every company can connect their scheduling procedure to EFFE. But EFFE also wants to cater to the needs of bigger clients. This is why we created building blocks.

Building blocks are features that can be added/removed from the application. This can be done by the user or by EFFE. Examples of building blocks are white labeling, integration with frontend system and payroll integration. These building blocks are not required when acquiring EFFE but can be added one by one.

3.2 Intention

So the question is how are we going to implement these building blocks. We have a few requirements:

- They need to be interchangeable. Meaning the same building block can be changed with another one that does the same job with maybe extra functionality.

- They should be able to do everything programming related. From if else to database calls.
- They have impact on the frontend as well as the backend
- Building block should be completely separate from the application (loosely coupled)

Chapter 4

Research design

This chapter contains the information on how the research will be conducted.

4.1 Research objective

4.1.1 The problem

Right now EFFE is developing an application for employment agencies in which those employment agencies can schedule their employees automatically. The current application is really basic and there are requests from potential clients to implement certain features. We decided to add something to the business model called building blocks.

> "Building blocks are interchangeable implementations of business logic that can be reused as efficient as possible"

EFFE is looking how to implement the building blocks in such a manner where scalability and maintainability are the focus.

4.1.2 Objective

The objective is to create a recommendation for an infrastructure on how to create and maintain that infrastructure. Where the focus lays on interchangeability and scalability of the different functionalities.

Stakeholder	Interest to the objective
EFFE	The obvious stakeholder is EFFE. EFFE will enhance its business model. But not only that. We will also create a better infrastructure which means that we can implement functionalities faster and cater more to the clients' need.
Client	The client is also the one interested in this process. They are probably not interested into what happens behind the scenes but they are interested in the possibilities it adds for them to EFFE's application

4.2 Research framework

4.2.1 Objects

This chapter describes who/what the objects are for this research and why.

4.2.1.1 Backend architecture

Arguably the most important object is the backend architecture. There is already a lot of research available regarding backend architecture. The backend is also the place where the business logic will be expressed. The backend connects to the database and thus needs a lot of attention when creating this section of the application.

4.2.1.2 Frontend architecture

The second object, frontend architecture, is a lesser known subject when looking at modularity of the actual system. Most of the big companies have a single frontend application per platform.

4.2.1.3 Deployment lane

The backend and the frontend are the software side of the equation but the hardware is also important. Where does the software run? How does it run? The deployment lane is the section that pieces it all together. This object

creates the hardware or virtual hardware. Sets this hardware up so it can then proceed to deploy the frontend and backend on the just created hardware. This process is very important and EFFE is not the first company wanting to adopt this. Which means there is already a lot of research in this area.

4.2.1.4 Project manager

The last object we want to research is the project manager. Because the research is focussed on implementing business logic in a modular way it is important to research what can go wrong when the business logic is translated to code.

4.2.2 Research perspective

The research perspective is straight forward. Because I am one of the founders of EFFE and I am also doing this research in name of EFFE it has my best interest to approach this research from the side of EFFE. This means that I will put more emphasis on sustainability than for example on the performance. Because for now performance can be dealt with later but if you want something to be sustainable you have to think about it from the ground up. Otherwise you will need to rewrite the whole architecture.

4.2.3 Research sources

This section will describe which sources will be used when evaluating the research objects. This will not include everything but a broad spectrum of the sources that may be used in the research:

- **Modular architecture books:** In the end everything I need to know all comes down to modular programming. Modular programming is a very broad term and it is important to find how someone else may look at this term.
- **Implementations of modular programming:** Theory is one side of the coin. Everything can work perfectly in theory but when implementing the theoretic side you will find problems you haven't thought of before.
- **Critique from outside:** It is known that software architecture is an opinion based subject. This is because especially this area of software is fairly young. Software architecture did not have a lot of time to develop

itself as far as some other aspects of software engineering such as operating systems. Because software architecture is young there are a lot of people voicing their opinions and it is important to look at the criticism on some of the architectures.

- **Researches on deployment of architecture:** I will be researching more than one architecture. Each architecture has its own development environment and deployment environment. The architecture of the servers on which the program runs is important but that will be heavily influenced by the architecture of the software. Nevertheless should it be researched separately from the software architecture

4.2.4 Evaluation criteria

These are the criteria or leading questions that will be asked to research objects. Note: not all evaluation criteria apply to all research objects:

- **What is the biggest pitfall when implementing business logic:** As mentioned in 4.2.1.4 Project manager there will also be considered how business logic is translated to code. Because the building blocks will eventually be different based on business logic. The research will include what can go wrong in which way.
- **What are the most used architectures in this area:** There is always a reason why one architecture is very common and the other one isn't. In the research the reasoning will be extracted and reflected on.
- **What are the most upcoming architectures that are focused on modularity:** Again the whole research is based on the building blocks. These modular functionalities that can be designed via a common interface. Which architecture has which solution for this?
- **Which programming languages has the best attributes to complement the modularity:** Some languages are written purely for scripting or some are written to be focused on implementing algorithms more easily. Each programming language has its attributes and which of these attributes are most defining and important to a modular system.
- **Which quality attributes are deemed most important to EFFE?:** The quality attributes from ISO 205010 [1] are the backbone of an the architecture. In the research will describe which are most important to

EFFE. It is then important to reflect the quality attributes we chose on the architectures.

4.2.5 Research framework

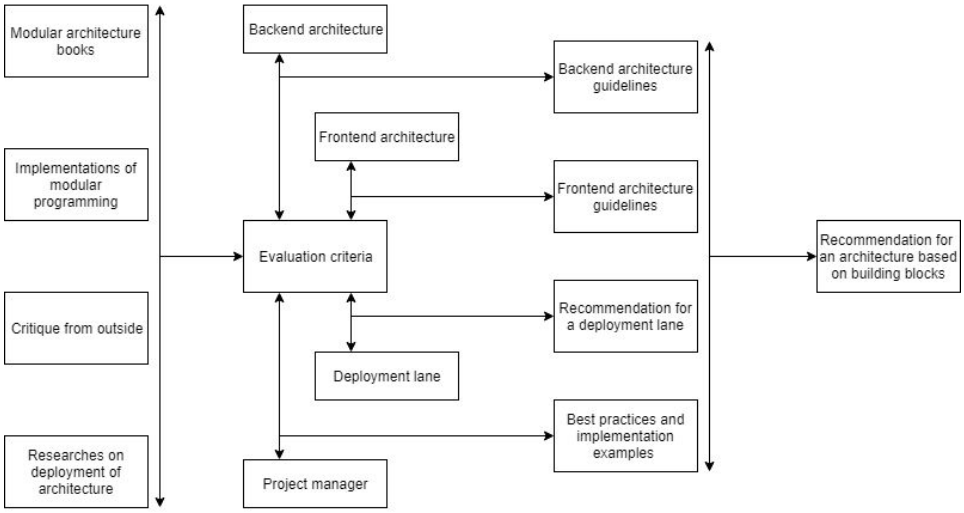

Figure 4.1: Research framework

4.2.6 Expected Results

The results will be the guidelines on which the practical part of this research will be based.

- **Backend architecture guidelines:** These are the guidelines on which the backend architecture will be based on. These guidelines will indicate why I choose for a certain approach and what the specific approach is.
- **Frontend architecture guidelines:** These are the guidelines for the frontend. Such as the backend guidelines these guidelines will also contain the reasoning for a certain guideline.
- **Recommendation for a deployment lane:** As mentioned in 4.2.1.3 Deployment lane the deployment lane can impact the backend architecture and vise versa. This recommendation will be implemented and should thus work perfectly with the backend and frontend.

- **Best practices and implementation examples:** What the project manager experiences and what can go wrong is important to then again pass to the evaluation criteria.

4.3 Research Questions

Note: question 2 and 3 will be handled separately for both backend and frontend.

4.3.1 Main question

From this we can derive that the main question is:

"What is the best way to transform a monolith into a modular architecture, where the services are interchangeable from each other"

4.3.2 Question 1

The first question that will be asked is what is the purpose of this question. The first question is about software architecture. How does a software architect create a software architecture. The model can be found in the appendix under Creating a software architecture.

The thicker red lines show the parts of the model I want to explore in the question. Thus the question is:

"What was the thought process behind choosing certain implementations for the quality attributes of a software architecture?"

This question will explore how a software architect chooses the architecture. This will give more insights into what they consider when choosing an implementations so that their rational can be extracted and taken into consideration.

These are some of the sub questions that will be handled based on this central question

- Which techniques are used when mapping the priority and the drawbacks in order to make a decision?

- How does the priority of a quality attribute influence the end result or software architecture?
- How does the software architect combine the priority, drawbacks and possible implementations to a software architecture?

4.3.3 Question 2

"Which software architectures that focus on modularity are available?"

This question focuses on the architecture that are available. The knowledge of how a software architect chooses the architecture is answered in the previous question (Chapter 5.3.1). In this question there will be a search on the architectures that are available and how they came to be.. Because of the new perspective gotten from the previous question there can be a more nuanced look at the architecture.

Here are some of the sub questions:

- Which are the upsides and downsides of each architecture?
- On what level is the documentation and research surrounding the architecture?
- Which architecture implements the quality attributes I deemed important best?

4.3.4 Question 3

"Which implementations are there of the solutions provided for modular architecture?"

The solutions or architectures provided from question 2 will have implementations or frameworks. It is important to see which implementation implements a certain choice of the architecture in what way. Other questions that will be answered are:

- How mature is the architecture in contrast to the implementations?
- How does the language chosen in the implementation reflect to the architecture?

- On what level does the framework compromise which is not reflected in the architecture?
- How is the community of this implementation?

4.3.5 Question 4

> "What are the key elements of in which a software architecture will influence the deployment lane?"

This question hints at the relation between a software architecture and the deployment lane. Right now there is a limited view on how the deployment lane should be and how it can be. In order for the practical research to work there needs to be an answer to these questions:

- Which infrastructure fits best with my chosen architecture?
- What are the costs of different infrastructures?
- How does the infrastructure implement our quality attributes

Chapter 5

Methods

Chapter 6

Creating an architecture

Chapter 7

Sources

[1] ISO 2500. *ISO/IEC 25010*. URL: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?limit=3%5C&limitstart=0 (visited on 03/06/2019).