

# Emergent architecture design

## Byzantine Generals

CTGGTGGTGCTCAGCTGCAAGTCAAGCTGCTCTCTGGGCTGTGATCTCCCTGAGACC  
CACAGCCTGGATAACAGGAGGACCTTGATGCTCCTGGCACAAATGAGCAGAATCTCT  
CCTTCCTCCTGTCTGATGGACAGACATGACTTTGGATTTCCCCAGGAGGAGTTTGAT  
GGCAACCAGTTCCAGAAGGCTCCAGCCATCTCTGTCCTCCATGAGCTGATCCAGCAG  
ATCTTCAACCTCTTTACCACAAAAGATTCATCTGCTGCTTGGGATGAGGACCTCCTA  
GACAAATTCTGCACCGAACTCTACCAGCAGCTGAATGACTTGGAAGCCTGTGTGATG  
CAGGAGGAGAGGGTGGGAGAACTCCCCTGATGAATGCGGACTCCATCTTGGCTGTG  
AAGAAATACTTCCGAAGAATCACTCTCTATCTGACAGAGAAGAAATACAGCCCTTGT  
GCCTGGGAGGTTGTCAGAGCAGAAATCATGAGATCCTCTCTTTATCAACAACTTGC  
AAGAAAGATTAAGGAGGAAGGAATAA, TGTGATCTCCCTGAGACCCACAGCCTGGA  
TAACAGGAGGACCTTGATGCTCCTGGCACAAATGAGCAGAATCTCTCCTTCCTCCTG  
TCTGATGGACAGACATGACTTTGGATTTCCCCAGGAGGAGTTTGATGGCAACCAGTT  
CCAGAAGGCTCCAGCCATCTCTGTCCTCCATGAGCTGATCCAGCAGATCTTCAACCT

# Preface

*Ali Smesseim, 4386248, asmesseim*

*Samuel Sital, 4225139, ssital*

*Kamran Tadzhibov, 4280784, ktadzhibov*

*Ravi Autar, 4361172, raviatar*

*Adam el Khalki, 4348435, aelkhalki*

***Byzantine Generals***

*Delft, April 2016*

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Design goals . . . . .	1
<b>2</b>	<b>Software architecture views</b>	<b>2</b>
2.1	Overarching architecture . . . . .	2
2.2	Sub-systems . . . . .	2
2.3	Hardware to software mapping . . . . .	2
2.4	Persistent data management . . . . .	2
2.5	Concurrency . . . . .	3
	<b>Glossary</b>	<b>4</b>

# Introduction

This document will represent the architecture of our context project Programming Life. We will update this document each sprint to keep it as recent as possible.

## 1.1. Design goals

### Availability

The product must always be working. It can be found at the master branch so that the client can always give feedback that can be used to adjust the product to the clients needs.

### Interactivity

An efficient way has to be provided to visualize DNA sequences in an interactive way.

### Scalability

The product must handle data of more than 6000 Genomes. It must not take too long or a reminder system has to be provided to remind the user if the data is processed.

### Maintainability

We would like to maintain the product easily. This is realised by providing the right documentation. It should be open for extension and closed for modification. We try to follow the SOLID principles.<sup>1</sup>

### Useability

The users of the system are biologists. They should be able to use the product easily and know how to use the product.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/SOLID\\_\(object-oriented\\_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))

# 2

## Software architecture views

The product loads data that is provided by the server. The files are then processed and stored in a database. The client that the user is using will send requests of different type to get information from the server.

### 2.1. Overarching architecture

The project core is based on the client-server model. We use a web application to represent the client and a proper Web server to represent the server.

The architecture is used because of the portability that comes with it. The product must be used all over the world and that is something our product will have as a core principle.

### 2.2. Sub-systems

There are currently 3 sub-systems. Below we will give more data about these sub-systems.

#### Parser

The parser module parses .GFA file files to the datastructure we are using. At this moment we are only parsing the id's and data of a segment and the id, source and destination of links.

#### Collapser

To provide the client the proper data to visualize the genomeset efficiently. The bubbles are recognized in here and the data gets ranked into a Zoomlevel.

#### Server

The server waits for input from the client to process. Based on the kind of input, the server processes it and sends back data to the client.

#### Webapp

The web application is used to visualize data that is received by the server after a request from the web application.

### 2.3. Hardware to software mapping

The product must be run on a workstation. At this moment in time we dont have any requirements for the workstation. Java and javascript are used. Further more the webapp can run on the workstation or on any other computer that has a webbrowser. As we will try to use a database, data persistency will not be an issue.

### 2.4. Persistent data management

The persistent data that is used by the product mainly is the datastructure that represents a graph. The data the webapp is looking at and a reasonable amount of data around this webapp data will be saved in main memory. This is to provide fast response to the webapp.

## 2.5. Concurrency

As multiple users can use the product, concurrency is quite important. At this point in time we support using multiple users on one dataset. These users cant edit the data. This does not need any concurrency. In the future we would like to add the feature that multiple users can each use their one dataset and even dataset sharing.

# Glossary

**DNA sequences** A nucleic acid sequence is a succession of letters that indicate the order of nucleotides within a DNA (using GACT) or RNA (GACU) molecule. 1

**Genomes** In modern molecular biology and genetics, the genome is the genetic material of an organism. It consists of DNA (or RNA in RNA viruses). 1

**.GFA file** This is the Graphical Fragment Assembly format, or GFA in abbreviation. Used to represent data of a set of genomes and their mutations. 2

**Web server** A Web server is a program that uses HTTP (Hypertext Transfer Protocol) to serve the files that form Web pages to users, in response to their requests, which are forwarded by their computers' HTTP clients. Dedicated computers and appliances may be referred to as Web servers as well. 2

**Zoomlevel** To visualize the data and to get situational awareness we use zoomlevels to represent the toplevel overview of a set of mutations. 2