

# Binance Stream

Léo Héritier, Cyril Roumegous, Sylvain Reynaud

# Table of Contents

1. Business context.....	1
1.1. Functional requirements.....	1
2. Technical specifications.....	3
2.1. Architecture.....	3
2.2. Final result.....	4
3. Time management .....	5
3.1. Léo Héritier.....	5
3.2. Cyril Roumégous.....	5
3.3. Sylvain Reynaud .....	6
4. PostMortem .....	7

# Chapter 1. Business context

The user wants to trade cryptocurrencies. Since trading is risky, a beginner trader can expect some big losses through his learning journey.

A beginner trader needs an application to test his strategies and his trades. To meet the demand, a trading application with fake money enables a beginner to trade like a professional trader without real loss or gain.

## 1.1. Functional requirements

Here is a list of features that the user should be able to use : - A user can create, sign in to an account - A logged user have a balance of fake money - A user can trade (buy or sell) on a pair using the balance of fake money - Simulated trades are based on the order book - A user sees the trade graph (candlesticks) of a pair, ie. real data of BTCUSDT from Binance - A user sees the order book of a pair (real data from Binance) - A user sees the financial indicator MACD on the trade graph - All data shown are live and real data from Binance



Figure 1: A trade graph with candlesticks (green and red lines) and MACD (purple line)



Figure 2: Order book of the pair BTCUSD (buy orders in green, sell orders in red)

## Chapter 2. Technical specifications

The application is a web application. The front-end shows information, any complex data aggregation are computed in the back-end.

Front-end and back-end are reactive. The back-end is made with Akka HTTP and Akka stream in Scala. The front-end framework is React with MUI for the UI. The back-end is an HTTP server and serves 2 web sockets.

The web sockets serve the order book and the candlesticks data in real time.

### 2.1. Architecture

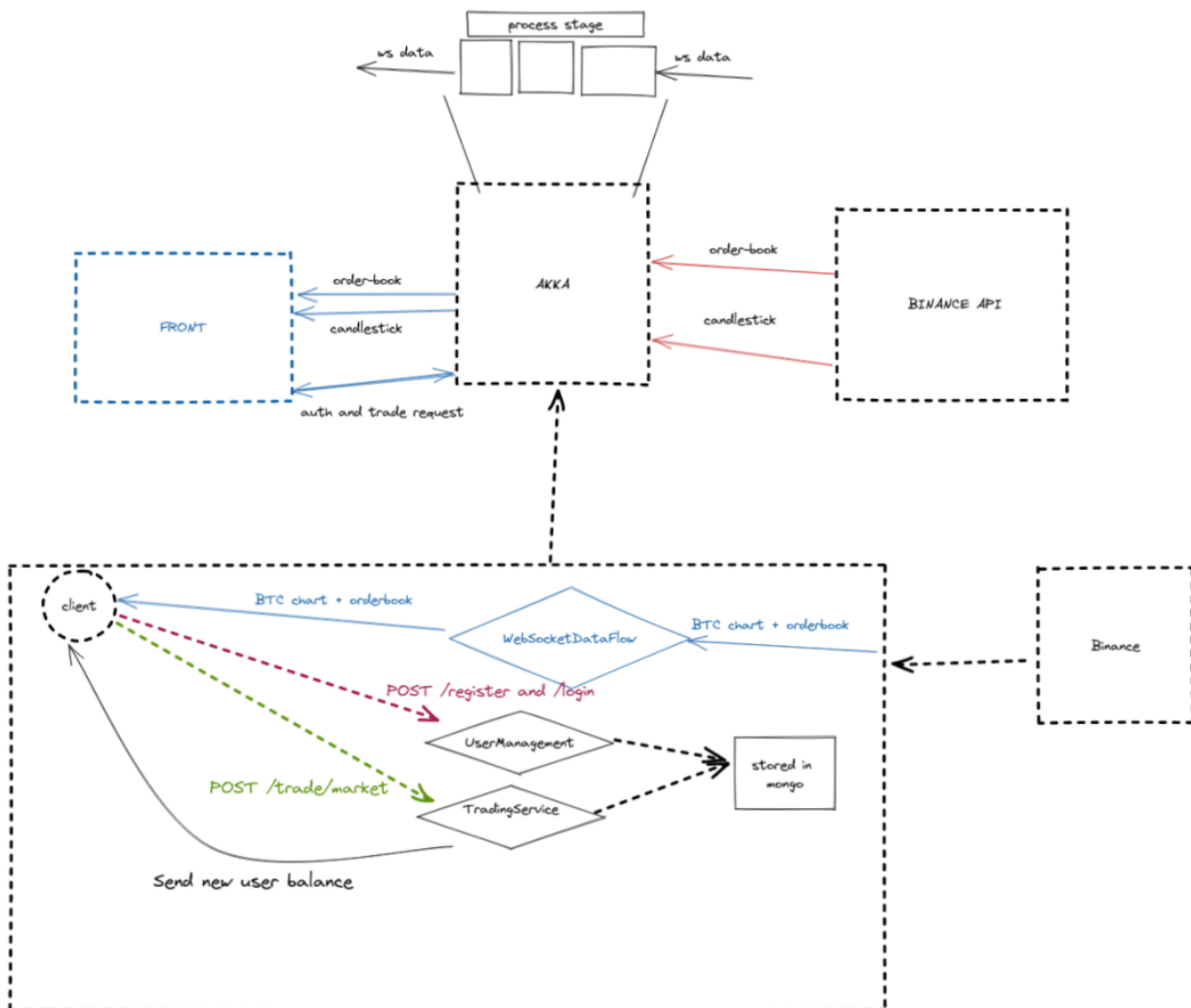


Figure 3: Architecture diagram

## 2.2. Final result



The application can be found at: <http://polytrade.cluster-2020-11.dopolytech.fr/>

We achieved a fully functional web application that allows users to authenticate themselves and do fake trading. This web application shows the user a candlestick chart of the Bitcoin price history. As well as the Binance BTC/USD order book. Both of them are real time data coming from Binance.

A panel allows the user to do fake trading with a view on his wallet balance. Only market orders are implemented, which means that the user decides what amount of btc he is ready to buy/sell and the order will be executed at the current market price.

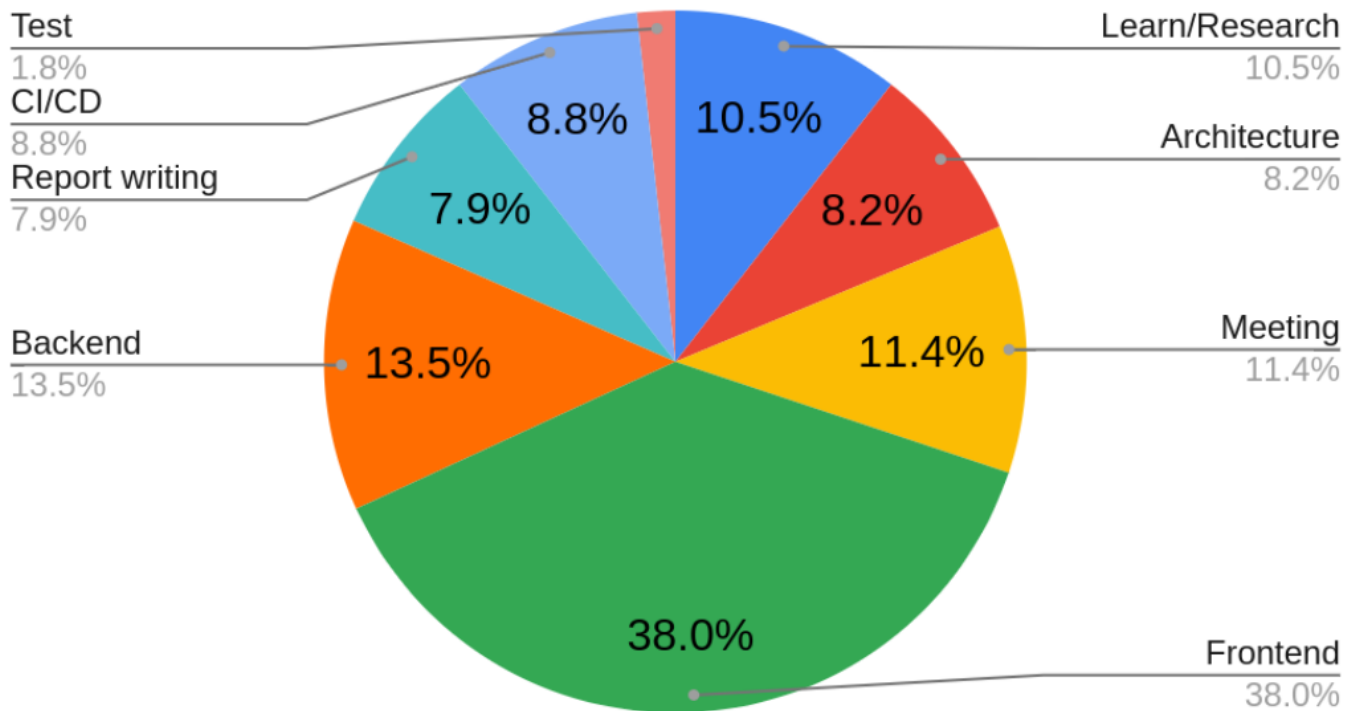
Interesting features that we could add in the future would be to: - Allow user to trade other assets than BTC - Allow user to activate graphical indicator on the candlestick chart - Add tools for the candlestick chart to make drawings on it - Add limit order in addition to current market order

## Chapter 3. Time management

### 3.1. Léo Héritier

Total time spent: ~25h

#### Time spent on each type of task

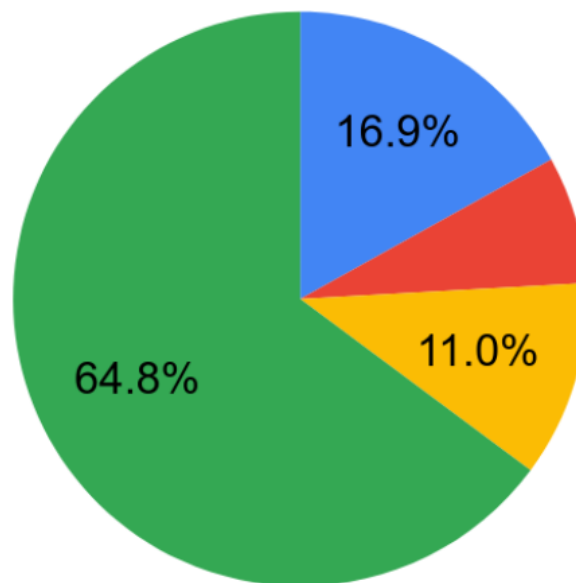


### 3.2. Cyril Roumégous

Total time spent: ~45h

## Time spent on each type of task

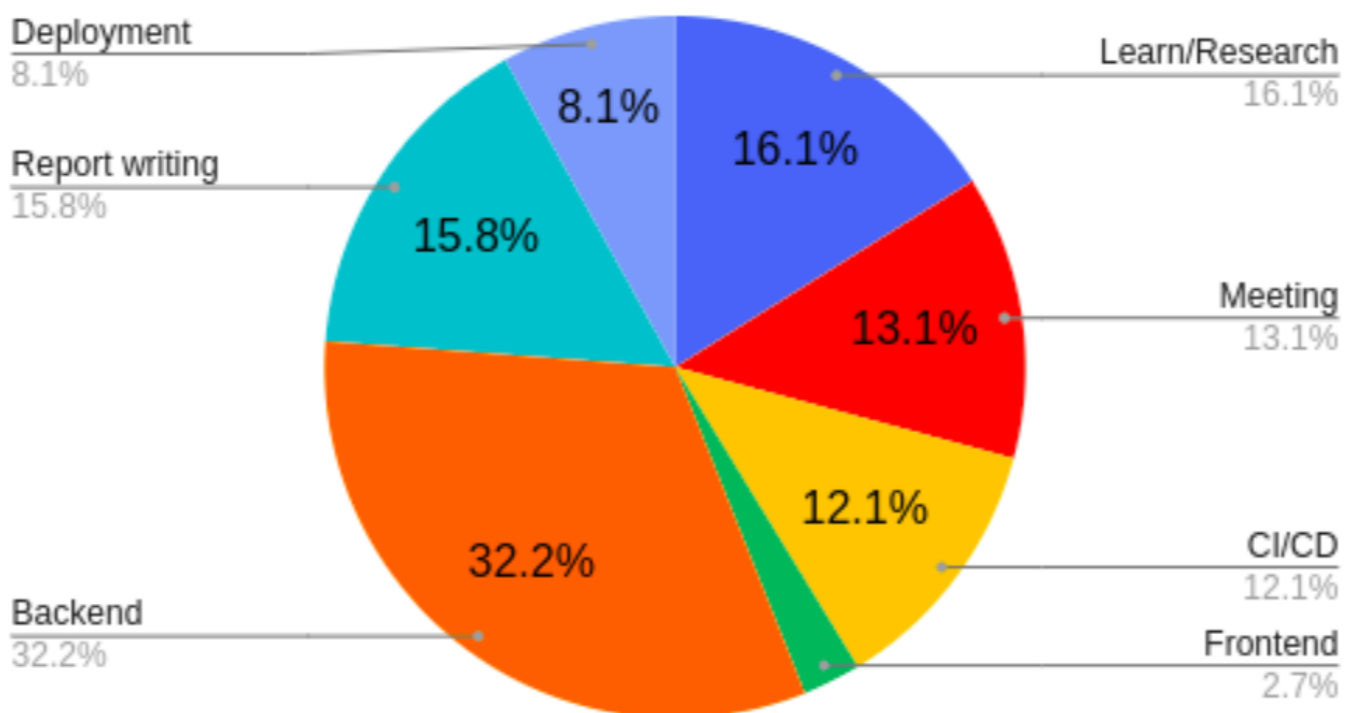
- Learn/Research
- Meeting
- Frontend
- Backend



### 3.3. Sylvain Reynaud

Total time spent: ~25h

## Time spent on each type of task







## Chapter 4. PostMortem

We achieved to match almost all requirements however we were quite late, and we had to work a lot at the end of the project. Also the workload was quite unbalanced. On the backend part, the use of Play framework would have saved us some time compare to using “low-level” akka-http. Also, the development with Akka Stream was hard, and we had some difficulty to handle the websocket flow.