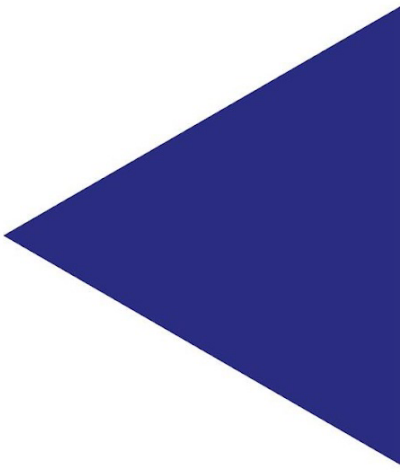
Project definition

Real-Time Stock Market Dashboard

HBO-ICT/ Big Data Minor
WS 22/23
Tobias Henning, Anh Thu Bui, Ivan Osipchyk, Ismail , Max, Akila



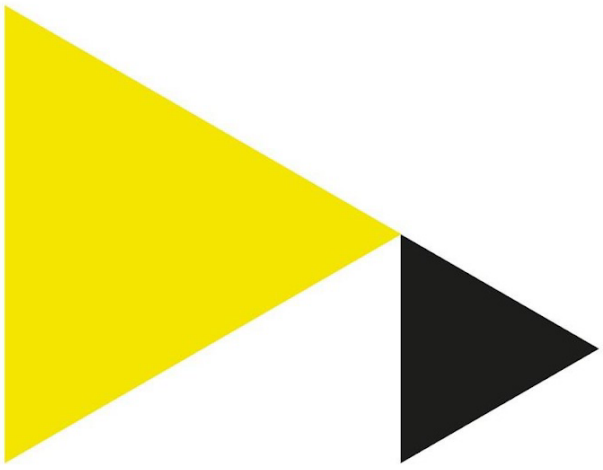
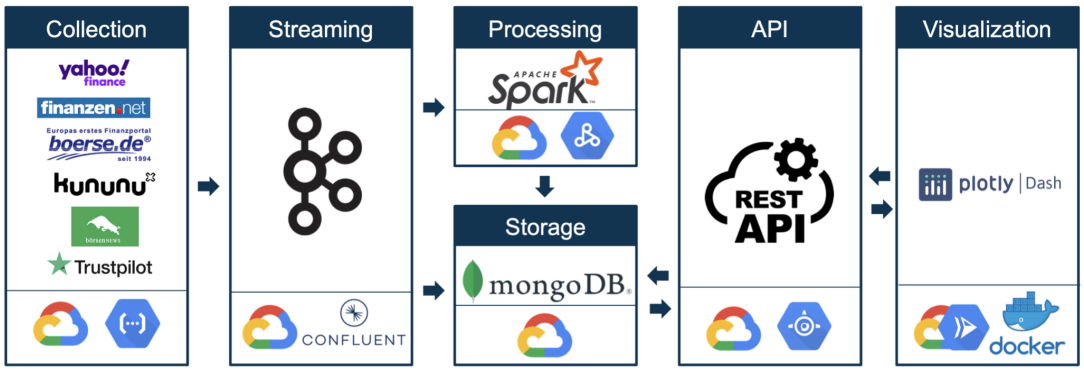


Table of contents

1. Proposal for the Data Solution

To solve the needs of customers we decided to use the following pipeline.



Our project consists of six main technical parts: Data Collection, Data Streaming, Data Processing, Data Storage, API and Data Visualization.

To collect the data for our project we will use web-scraping. Python has two great tools to do web-scraping: BeautifulSoup and Selenium. Different technologies are applicable for different websites, so we might use both of them. As sources of information we will use websites that provide relevant information for stock price prediction. We will collect not only the data about stock prices in different periods of time, but also news about that company and the kpi's because it also influences future stock prices.

Collected data will be streamed in two directions: for Processing and to Storage. For Data Streaming we will use the Apache-Kafka-Streaming-Platform. The Kafka Producer API allows applications to send data streams to the Kafka cluster. With the Kafka Consumer API, applications can read data streams from the cluster.

The main purpose of the Data Processing stage is to make predictions about future stocks prices. We decided to use Apache Spark because it has three big advantages in this situation. Firstly, it's the best solution for real-time data streaming. Secondly, it also helps to reduce the amount of code in the project. Lastly, it is a good solution to work with a big amount of data. That can help to increase project scale in the future.

Our data will be stored in a non-relational MongoDB database. We decided to use an unstructured database, because it's the best solution for real-time data. And MongoDB is one of the most popular unstructured databases.

The next step in the pipeline is using REST API. It provides secure access to the data. Our project is designed for one company and we don't want others to have access to its data.

All previous steps are the backend part of the application. The last step is the only step in the pipeline, which refers to the frontend. On the website there will be several things visualized. Home, Key Performance, Investor Relations, Company Environment are among them. The most

interesting part of the webpage for the customer is graphs of stock prices. There are several methods to visualize our predictions. We decided to use plotly library. It's a good solution, because with the help of plotly it's easy to create interactive graphs. These graphs are the most relevant for our situation.

2. Description of the deliverables

The manager's demand is to be delivered with an end product which will allow him to look at the current market situation, his company's current market performance compared to rival companies and for it to be able to predict the future course of the stock exchange, giving him a way better overview.

We will be using a more classical data(BI) situation to solve the needs of the customer as the end goal is to help make the manager more informed business decisions and the whole purpose of business intelligence is to utilize data for analytical purposes so that essential information can be determined, which will then be presented in the form of graphs, charts and tables. It is an opportunity as these charts and graphs will end up help the manager avoid making mistakes.

Our project consists of six main technical parts: Data Collection, Data Streaming, Data Processing, Data Storage, API and Data Visualization. Data is essential for this project and its collection will happen from sites like Yahoo! Finance, finanzen.net and Trustpilot.

Data when talking about Stock-Exchange Data is separated into pre-trade and post-trade data.

Pre-Trade Data will give you the details of a specific asset, its bid and ask price and will also help you in analyzing the market as you will have a better understanding of the asset's value based on historical trends.

Post-Trade Data will give you the details of how the transaction of a specific asset took place, when an asset was sold and what its last trade price was.

With this being said we believe with the data that we will have, will be able to execute descriptive, prescriptive and predictive data analysis.

The delivered product should look the following way. The navigation bar is on the left side of the webpage. It contains:

- The company, which graph will be presented. The customer can choose the company
- Homepage button
- Key Performance page button
- Investor Relations page button
- Company Environment page button

This part is the same for all webpages on the website.

To the right of the navigation bar are two blocks. First of them contains an interactive graph with stock prices. The customer can choose the time period of the displayed graph. The buttons to switch the time period are above the graph. Second block represents the latest news about the chosen company.

3. Description of intermediate results

As a starting point we decided which companies we will make our project for. This is going to be any of the tech companies from the nasdaq-100. There are 41 companies related to that field.

Our intermediate results are the following:

Scraping Server
This script will be able to call multiple api's and scrape different websites to get data regarding different performance indicators and descriptions of different companies of the before mentioned scope. The server will run a Python script for that.

Confluent Server
The Confluent server will run kafka streams to connect the scraping server with the processing server.

Processing Server
This server will receive raw data from the scraping server through kafka streams. The data will be processed by the server and saved in a mongo database using the Python language. It will also train different classifiers to predict future stock courses.

Mongo Database
The Mongo database will be deployed using docker and kubernetes. The results from the processing will be saved in this Database. The api server will also call this database to get the data it needs.

Api Server
The api server is the interface between the backend processing and the frontend. It will be written in Python. The React website will request different data for visualizing the dashboard from this server. It will contain multiple routes for different graphs, diagrams and texts.

React Website
The dashboard will be implemented using the framework ReactJS. It is written in JavaScript for the website setup and in Python for the visualizations. The website contains a drop down menu for the different companies as well as multiple sections for each selected company where the performance indicators, news, description of the company and so on can be found.

4. Roles

Product owner: Ismail

For the following sprint we decided to assign roles this way:

SCRUM Master: Tobias
Developers: Max, Akila, Anh Thu, Ivan

Backend:

- Collection -> Max, Akila, Anh Thu, Tobi, Ismail
- Streaming -> Ismail, Tobi
- Processing -> Ivan, Anh Thu
- Storage -> Ivan, Ismail
- API -> Max, Akila, Ismail

Frontend:

- Visualization -> Tobi, Max
- Docker -> Tobi, Max, Anh Thu

Other:

- Google Cloud -> Ismail, Ivan, Tobi