# Web application designed and develop in containerized environment

* Index
* State of the art
* Architecture
  + Streaming bus
    - Apache Kafka
  + Micorservices
  + Monolithic Service
* Technologies
  + Kubernetes
  + Helm Charts
  + Java 11
    - Spring Boot
    - JBoss (?) WildFly(?)
  + React JS
  + Time Series Database
  + Relational Database
* Security & Privacy
  + JWT
    - Not exposure of the token (?)
  + Login using JWT
    - Realms
    - Roles
    - Claims
  + Keycloak
* Devops – Continuous integration
* Unit testing
  + JUnit
  + Cypress
* Agile
  + Scrum
* Benchmarks
  + Scalability definition
  + *Streaming* Bus: Apache Kafka vs Moquitto
  + *Services*: Micro vs Monolithic
  + *Time Series DB*: Influx vs Redis (?) vs Mongo (?)
* Future developments

## Index

During my working experience in algoWatt s.p.a, I was employed to develop a very complete web application that were built with some of the advanced and challenging technologies that are currently used in these contexts. In this experience, I was involved in writing code for a large part of the components of the project, which I learned to manage, and which enriched my knowledge. However, there was much more to discover about the entire web application.

The idea for this master thesis project was born from this industrial web application with the goal of studying the technologies used and test them in different conditions. My work was to develop a system that could handle and communicate with many **IOT devices** with the best performance possible, I also want to describe the procedures for the release of the web application and demonstrate the efficiency of them.

My project relied on two main software for the deployment of all the component of the web application, which are Kubernetes and Helm. The components are deployed using different technologies such as React, Spring Boot and Python. Some crucial containers are Kafka, which is the chosen streaming bus, Grafana and Prometheus. Other two important containers are MySql and InfluxDb, that are both databases, and they are used to store the data of the web application.

Kubernetes is a widely used software created by Google, who set it open source in 2014. Some of the biggest companies who use Kubernetes together with Helm are, of course, Google, Spotify, The New York Times, and Adidas. The Java framework Spring Boot is widely used by the biggest companies like Netflix, Udemy and Trivago due to his security, scalability, and simplicity. Python, instead, is even more used than Spring Boot, especially in Data Science due to the simplicity in reading and writing code. Some big companies that use Python are Wikipedia, CERN, again Google and Spotify, Amazon, Meta, and NASA. Finally, React is a JavaScript library created by Facebook back in 2011 and it’s use by obviously Meta, Netflix, AirBnB, and Dropbox. Speaking about Databases, for first let’s look to MySql, it’s a consolidated DBMS based on SQL and used by Sony, BBC, and Uber. Instead, InfluxDb used as a Time Series DataBase is widely used by many big companies like PayPal, Adobe, CERN, Cisco, and much more.

In the end, in order to maintain a backup of all the files used in my project and keep track of the changes during the development, was used a Version control system call GitHub.

## State of the art

My work is divided in two branches that I can identify as **Backend** and **Frontend**, connected each other via the streaming bus. The Backend is the part that strictly operate with the IOT devices and collect data from them. Instead, the Frontend contains all the components dedicated to the interaction with the **final User**, which can check the entire system status and the detail of a single IOT device.

There is a component which is dedicated to the User Interface, and it was developed using React. Some other components are developed using the Spring Boot Framework for the rest services, doing some scheduled jobs and maintain a socket service.