Accelerometer Activity Recognition

Made with ThingsBoard and Node.js

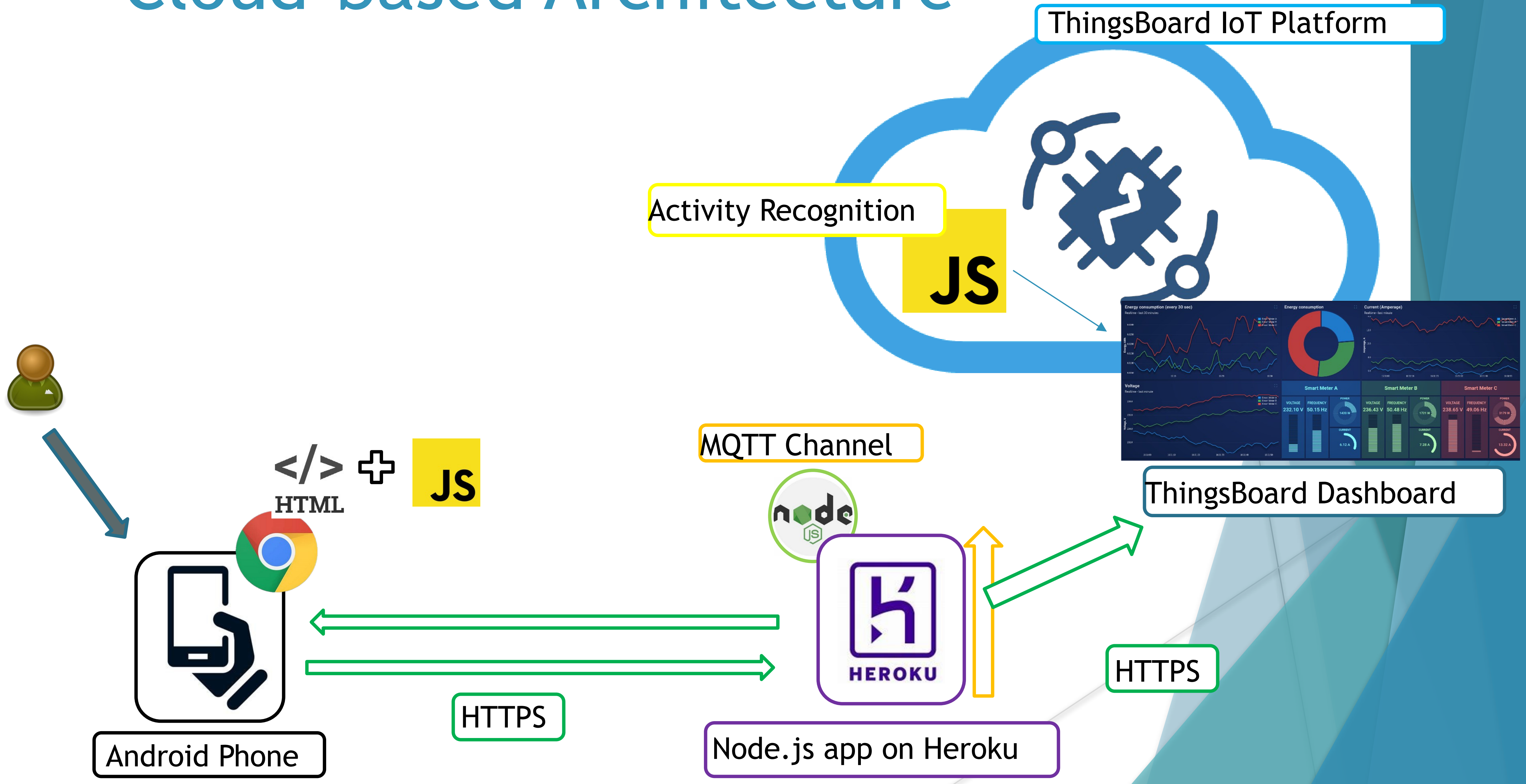
Main idea

- We want to build a web application that collects data from the accelerometer in a mobile phone and does some activity recognition (it will be possible to see if the user is standing still, is walking or if he’s running).

- Cloud-based deployment: activity recognition is done in the cloud platform when new data arrives.
  The data is visualized in the dashboard we used for the previous assignments, where we’ll see the latest values collected and the activity detected, together with the history of the values collected in the last hour and the resulting activity.

- Edge-based deployment: activity recognition is done in the mobile phone, when new values are read from the accelerometer, and then the data is sent to the cloud platform.
  The activity detected will be visualized in real time on the page displayed to the user, and it will also be possible to look at the activities collected during the last hour.

Cloud-based Architecture



ThingsBoard IoT Platform

Activity Recognition

ThingsBoard Dashboard

MQTT Channel

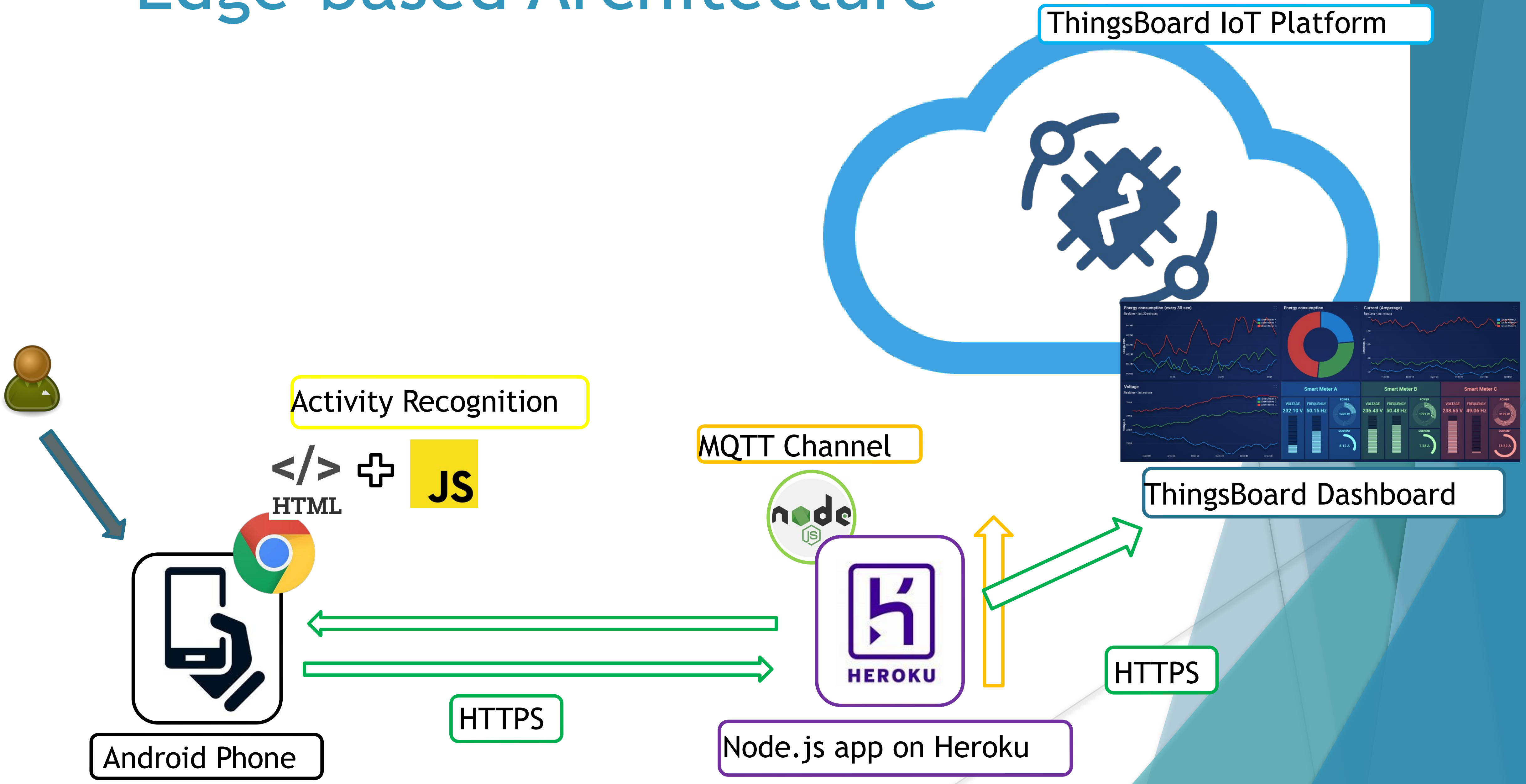
HTTPS

Android Phone

HTTPS

Node.js app on Heroku

Edge-based Architecture



ThingsBoard IoT Platform

Activity Recognition

MQTT Channel

ThingsBoard Dashboard

HTTPS

HTTPS

Android Phone

Node.js app on Heroku

Implementation

- We can have access to the accelerometer data thanks to the Generic Sensor API. To use them, we need to use Google Chrome as a browser, activating some flags. The client is served an HTML page with some JavaScript that collects data and makes the page responsive. Please note that you can activate these flags only on an Android phone.

- The web app is built with Node.js, which forwards data to the cloud platform via MQTT. To exchange data from the front-end to the back-end and viceversa we use HTTP methods GET and POST.
  The deploy is done with Heroku.

- The cloud platform used is ThingsBoard. Thanks to its rule chain we can easily process data in the cloud-based scenario, and the data is visualized on the dashboard.

Technology: client-side

- Generic Sensor API: is a framework promoted by the W3C to expose data collected through sensors in a consistent way, especially on mobile phones.

- Google Chrome: a web browser developed by Google. It’s currently the most used browser.

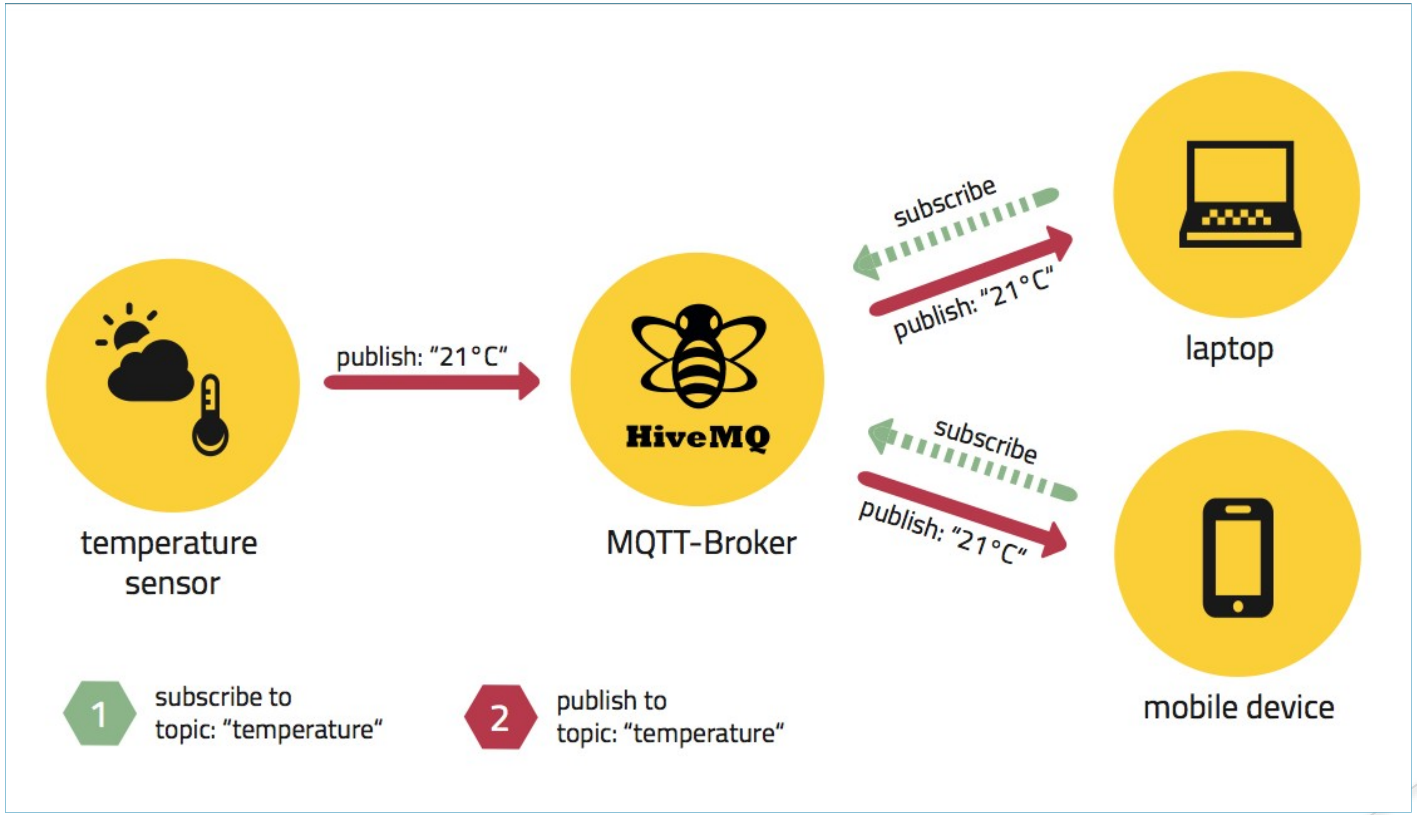
- HTML: is a markup language, the standard used to define documents to be displayed by web browsers. Often used with CSS for styling and with JavaScript to enhance functionalities.

- JavaScript: is an object oriented and event oriented programming language. It’s very versatile, and is used in the majority of web application to program the client-side.

Technology: MQTT Protocol

- MQTT is a machine-to-machine connectivity protocol, designed to be lightweight and work well even in situations where little resources are provided. Because of this, it’s ideal for sensor communication.

- It’s based on the publish subscriber paradigm



Technology: Node.js

- Node.js is a JavaScript runtime environment that uses V8 JavaScript Engine (which makes it very smooth and powerful).

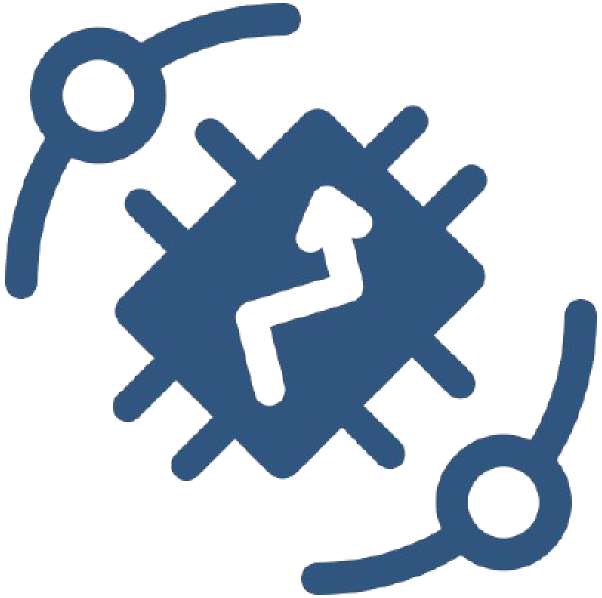
- It’s run in a single process, without creating a new thread for every connection. In general, it’s thought for an asynchronous and event-driven programming style.

- It’s very popular, and has a huge amount of frameworks and libraries.

Technology: Heroku

- Heroku is a cloud platform where you can deploy your applications. It has been in development since 2007, and supports many programming languages, such as Ruby, Scala, Go, NodeJS, Clojure, Python, PHP and Java.
- It has different payment plans, but for our purposes the free plan is more than enough.

Technology: ThingsBoard



- ThingsBoard is an open-source IoT platform to collect, process, visualize data and manage devices.

- Enables connectivity through standard IoT protocols like CoAP, HTTP, MQTT and supports cloud deployment.

- The pillars of this platform are scalability, fault-tolerance and performance.

References

- Generic Sensor API: https://w3c.github.io/sensors/
- HTML: https://html.spec.whatwg.org/multipage/
- JavaScript: https://en.wikipedia.org/wiki/JavaScript
- Google Chrome: https://en.wikipedia.org/wiki/Google_Chrome
- Node.js: https://nodejs.org/en
- MQTT: https://mqtt.org/
- Heroku: https://www.heroku.com/
- ThingsBoard: https://thingsboard.io

Thank you for you attention!