Environmental system

AWS solution

Fontys | ICT

V0.1

Alexandru Emilian-Mihai

Table of Contents

1 Intro 3
2 Goal 3
3 How have I tackled the project? 3
3.1 Device solutions 3
3.2 Cloud services 3
3.3 Software solutions 3
4 Diagrams 4
4.1 Context diagram 4
4.2 Container diagram 5
5 Motivation of choices 5
5.1 Devices 5
5.1.1 ESP32 5
5.1.2 Nordic Thingy:52 5
5.2 Cloud 6
5.3 Frameworks 6
5.3.1 Serverless framework 6
5.3.2 Angular framework 6
6 What have I learned? 6
7 Conclusion 6

1 Abbreviations and Definitions

Terms	Description
AWS	Amazon Web Services
BLE	Bluetooth Low Energy
GATT	Generic Attribute Profile
API	Application Programming Interface
MVVM	The Model-View-ViewModel
CLI	Command-Line Interface

2 Intro

This project is a cloud solutions that uses Nordic Thingy:52 and ESP32 as a demo devices to monitor environmental data. The ESP32 is going to serve as a BLE to MQQT bridge, exposing the BLE GATT characteristics of the Nordic Thingy:52. The bridge device is registered and transmits the data over MQQT to the IoT Hub.

The following environmental sensors are supported: Temperature, Humidity, Air pressure, Air quality (CO2 and TVOC) from the Thingy:52. The weather data from AWS IoT is stored in AWS S3 and extracted using a serverless Lambda backend created using serverless framework as foundation. The data is visualized in a dashboard created using Angular framework as the foundation and D3.js for chart generation. The visualization will contain the values over time, feature comparisons and distribution.

3 Goal

The goals I want to achieve by the end of this project:

- Deployment and development of an IoT application
- Deployment and development of AWS serverless Lambda for backend
- Expose data using serverless back-end
- Deployment and development of AWS API gateway
- Visualize data inside a web application
- Learn and use the D3 JavaScript library

4 How have I tackled the project?

4.1 Device solutions

4.2 Cloud services

4.3 Software solutions

5 Diagrams

5.1 Context diagram

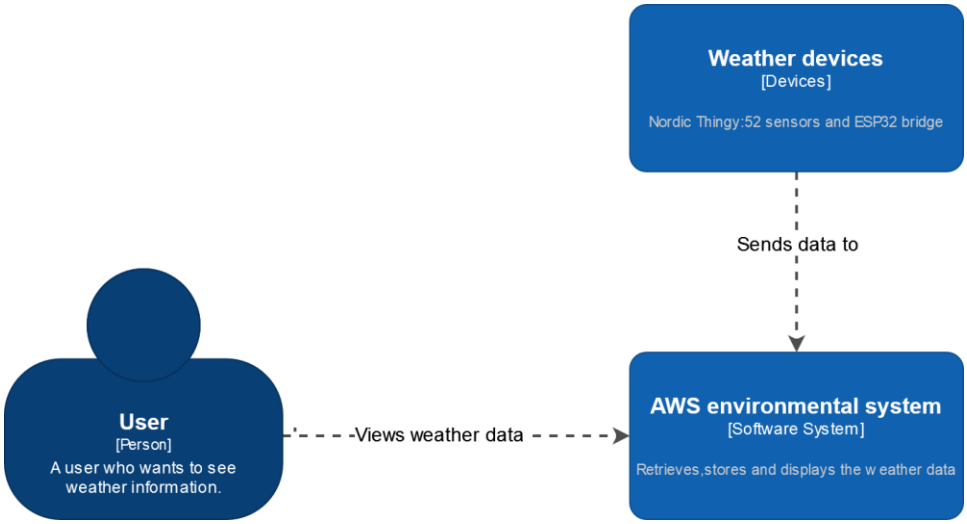

Figure 1-Context diagram

5.2 Container diagram

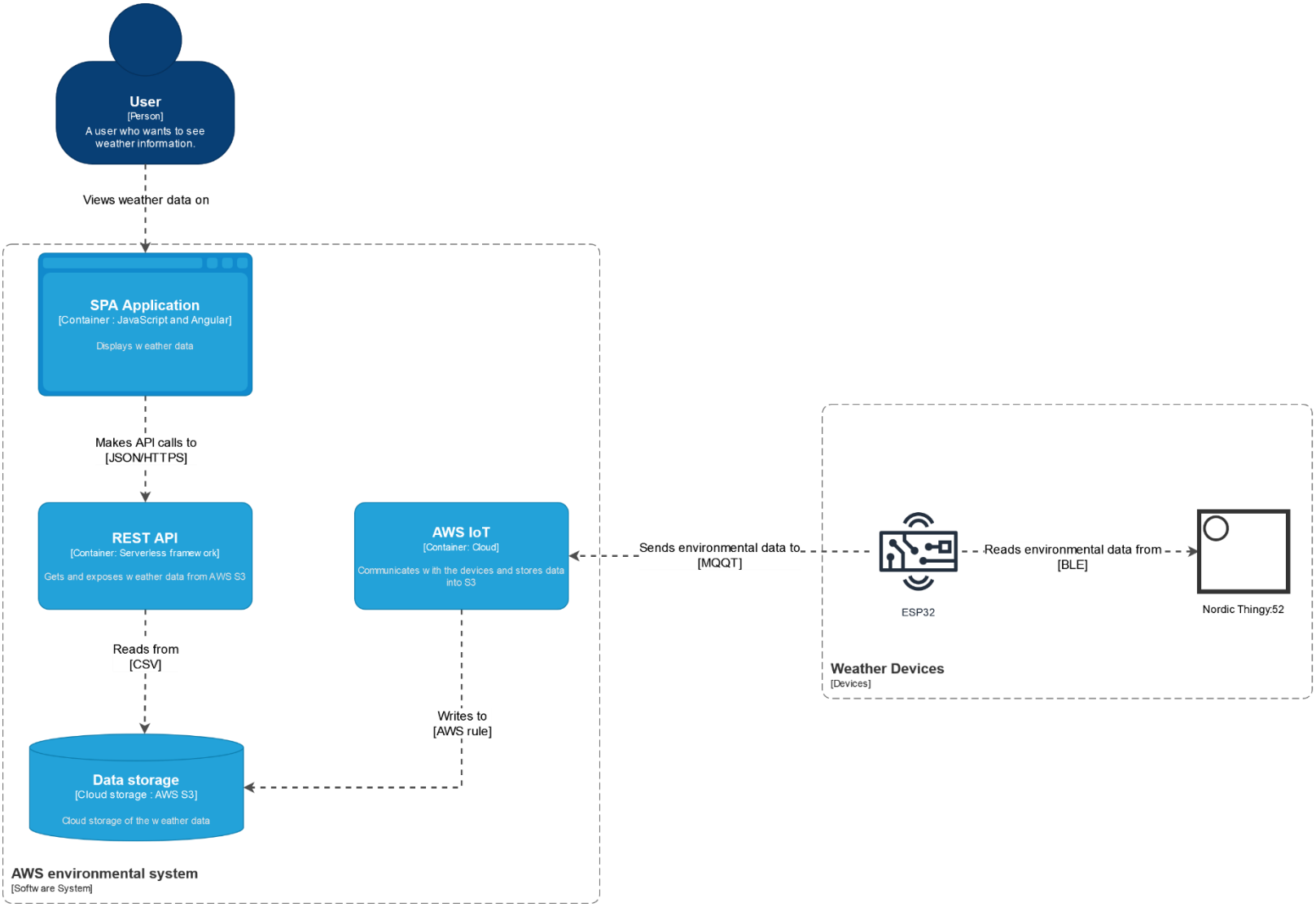

Figure 2-Container diagram

6 Motivation of choices

6.1 Devices

6.1.1 ESP32

The reason I choose to add the ESP32 is because it has the capability to communicate with other devices via MQQT and BLE. I believe that using this board will provided me with an easier and less time consuming solution for the communication with the cloud but also with the sensors.

6.1.2 Nordic Thingy:52

The reason I choose to use the Thingy:52 is because it offers a built-in solution for setting up a GATT environmental server. Additionally, the devices comes with a handful sensors for gathering environmental data.

6.2 Cloud

Amazon Web Services is the first platform I have experimented and researched. It has a very active community with lots of good documentation. With a very attractive free plan it offers a wide variety of services, including compute power, database engines, resources for authentication, services for migration of data and storage solutions. Due to all those services that AWS offers, the pace of development is significantly faster. Additionally AWS provides lots of ways to experiment, try different services, tiers and servers before deploying your solution.

6.3 Frameworks

6.3.1 Serverless framework

Serverless framework is an all-in-one development & monitoring of auto-scaling apps on AWS Lambda making the back-end implementation and development faster. It does not present a challenging learning curve and it has good documentation.

6.3.2 Angular framework

Angular is my framework of choice because it exceeds the others JavaScript frameworks regarding detailed documentation, community support, and is highly opinionated. As a framework, it provides an easy and complete startup by making use of the Angular CLI. Being the most “opinionated” Angular provides all the tools needed to make it easy to build my web application, including but not limited to state management, routing, and dependency management. The enforced MVVM pattern is advantageous as it provides a separation of concern.

Due to its variety of structures, Angular scores are lower in ease of use and ease of learning criteria. These structures will make the project more challenging but this is not my first time tackling the framework.

7 What have I learned?

8 Conclusion