**IoT Use Case: Smart Container**

# Overview

The smart container serves as a critical component within the cold chain logistics system, specifically tailored for temperature-sensitive transportation scenarios. Its primary goal is to deliver a steady stream of accurate, real-time data with minimal delays to external applications. This data encompasses live readings of temperature, humidity, and precise GPS coordinates.

For instance, consider a situation involving blood transportation: At Groote Schuur Hospital in Cape Town, there arises an urgent need for a patient requiring a blood type O transfusion. Unfortunately, the hospital's current supply falls short. In response, they promptly reach out to their closest blood bank for an emergency provision. The blood bank, recognizing the necessity for a controlled environment, entrusts the blood to our smart container, ensuring it is maintained at a precise temperature of 6°C. A dedicated driver is dispatched for the timely delivery.

Throughout the transportation process, both the hospital and the blood bank are empowered to closely monitor the container's whereabouts and the current temperature status. This level of real-time tracking and data accuracy ensures the safety and efficacy of this vital medical delivery.

# Technical Requirements and Boundary conditions

## Requirements / User Stories

* Visualization of real-time position track for client
* Visualization of current temperature in the cooler box
* Notifications e.g via sms or mobile app when temperature drastically changes
* Notifications e.g when destination is reached

## Hardware

* Arduino ESP8266
* NEO-6M / NEO-7M GPS module
* Temperature and Humidity Sensor module
* Insulated container

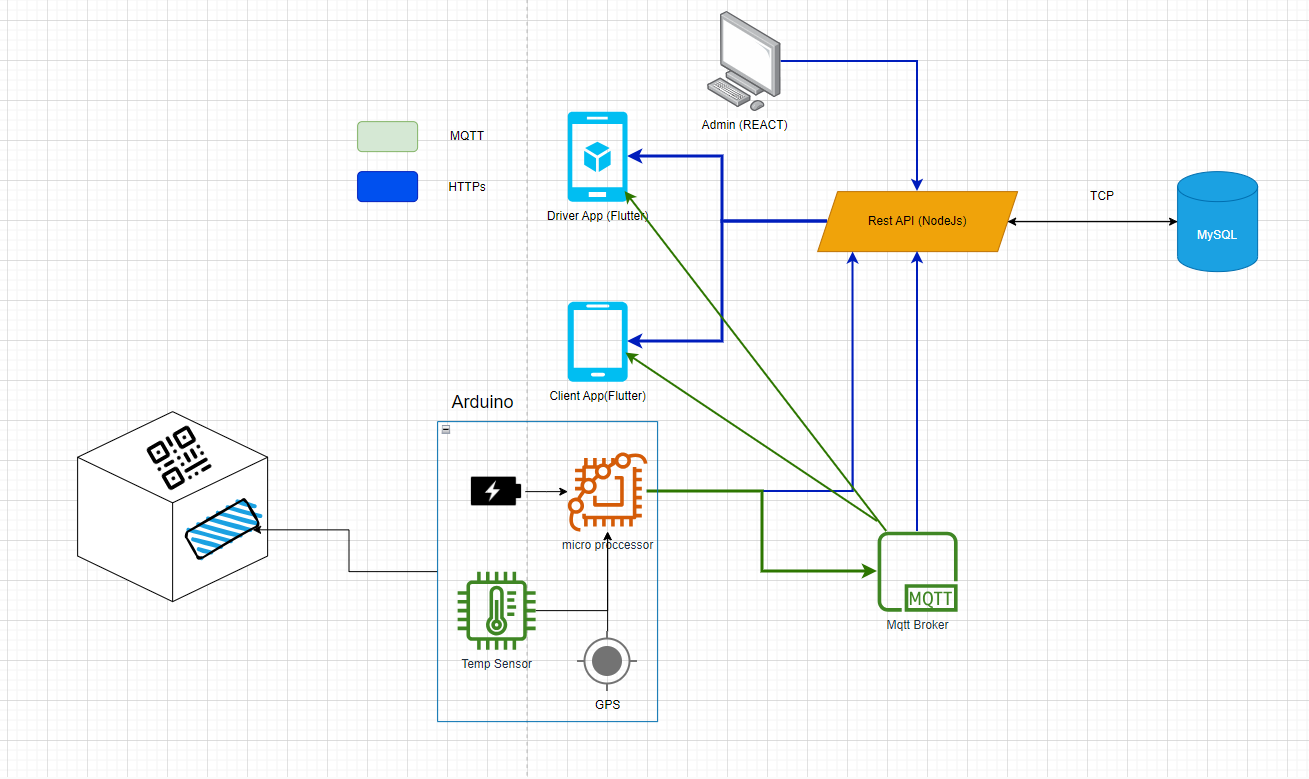
## Recommended software development environment

* Arduino IDE
* HiveMQ for MQTT tests
* Flutter (Mobile apps)
* React for admin (web-based application)

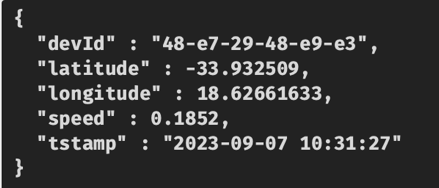
# Design

## Architecture

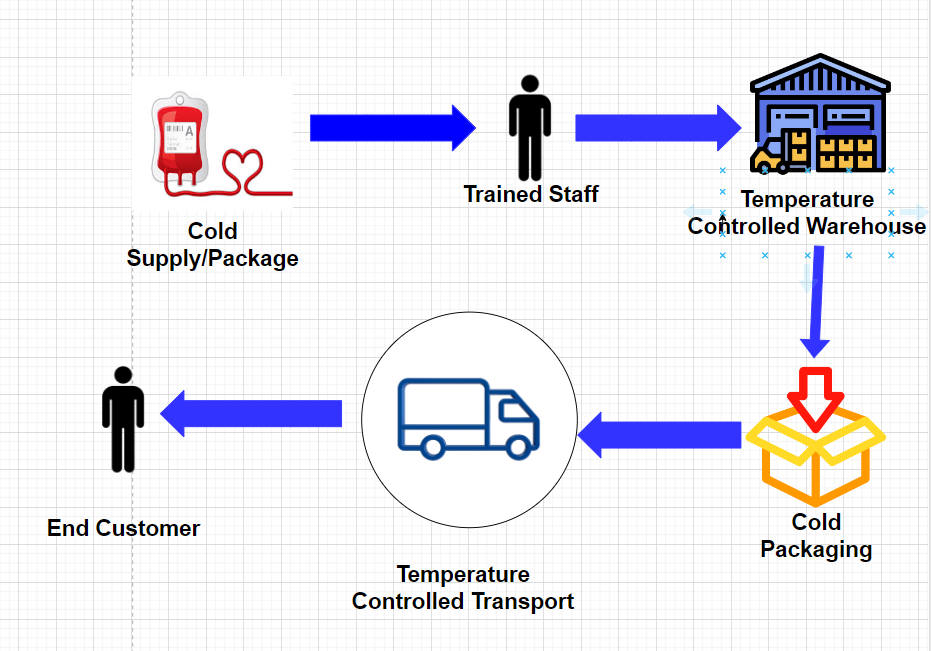
Proposed architecture



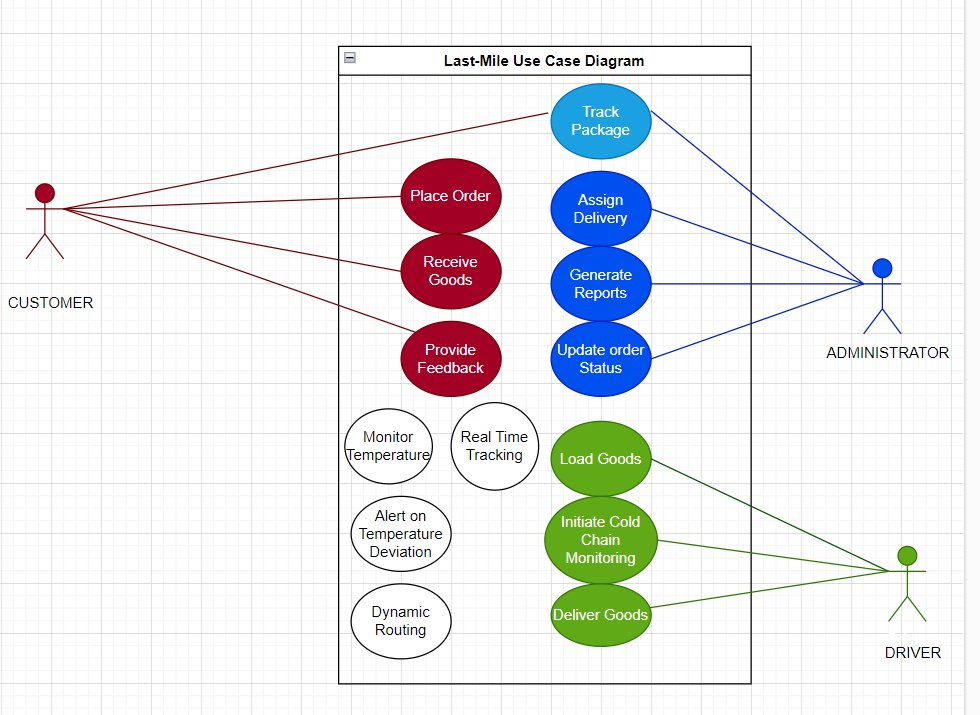
## Data Model



## Process Model / Activity Diagrams



Use case Diagram.



## Wireframes

# Implementation

## Smart Container

Github repository: https://github.com/dany-meyer/uwc\_tests/tree/main/Smart\_Container\_Arduino

## Driver App

Github repository: https://github.com/CS-UWC/Last-Mile/tree/master/driver\_cargo

## Admin App

Github repository: https://github.com/CS-UWC/Last-Mile/tree/master/cargo\_admin

## Dashboard

# Test

## Modul Test

**NEO-6M/7M GPS Module**

The GPS module is rather inconsistent in picking the satellite signals to generate the coordinates. Often time the GPS module must be exposed to the sky to get satellite signals this may be a suitable solution in highly flexible testing environments but becomes an issue when the module is closed inside the container.

The power supply to the module does not have many inconsistencies but does require a stable power supply as it has its own battery.

**Temperature/Humidity**  
The above modules only require a steady connection to the correct pins do not have any other inconsistencies associated with them.

## Integration Test

## Final Acceptance Test

Screen Shots

## Sources to learn from the Project

1. Create Nodejs
2. Programming the micro controller: https://github.com/dany-meyer/uwc\_tests/tree/main/Smart\_Container\_Arduino
3. Create Database
4. Create Flutter App