UNIVERSITY OF THESSALY

Co-funded by the Erasmus+ Programme of the European Union

Real-Time GPS Tracking System using MQTT and Expo

Firstname : Tsantamirindra

Lastname : RANDRIANARISOA

Course : Wireless Sensor Networks and Internet of Things

Erasmus Student from Madagascar 2025-2026

Table of contents

Introduction

Wireless Sensor Networks (WSNs) and the Internet of Things (IoT) play a crucial role in modern connected systems by enabling devices to collect, transmit, and process data over the internet. These technologies are widely used in applications such as smart cities, environmental monitoring, healthcare, and real-time tracking systems.

As part of my research on IoT Cloud Services, I aimed to understand how sensor data can be transmitted, managed, and visualized using cloud-based platforms. To put this into practice, I implemented a real-time GPS tracking system without using physical hardware, relying instead on free and accessible tools.

This project simulates a GPS-enabled device that publishes location data to an IoT cloud platform using the MQTT protocol. A mobile application built with Expo then subscribes to this data and visualizes the trajectory in real time on a map. Through this implementation, the project demonstrates how IoT cloud services can be used effectively for data communication and visualization, even in a resource-constrained environment.

1. System overview

The system is designed to simulate a real-time GPS tracking application using cloud-based IoT services. It consists of three main components:

1.1. GPS data simulator

Since physical hardware was not used, a software-based GPS simulator generates location data along a predefined route. This component:

- Creates latitude and longitude coordinates using linear interpolation.
- Packages the data with a device ID and timestamp in JSON format.
- Publishes the data to a cloud MQTT broker at regular intervals.

1.2. IoT Cloud Platform

HiveMQ Cloud serves as the central hub for message communication between the GPS simulator and the mobile application. Its main responsibilities include:

- Receiving published GPS data from the simulator.
- Maintaining real-time message delivery with QoS (Quality of Service) guarantees.
- Forwarding location updates to all subscribed clients efficiently.

To set up the cloud platform for this project:

- A HiveMQ Cloud instance was created with a unique broker URL and port.
- A username and password were configured for secure authentication.
- TLS encryption was enabled to ensure secure transmission of data.
- A specific MQTT topic (iot/device001/location) was created to organize messages from the simulated device.

1.3. Mobile Application

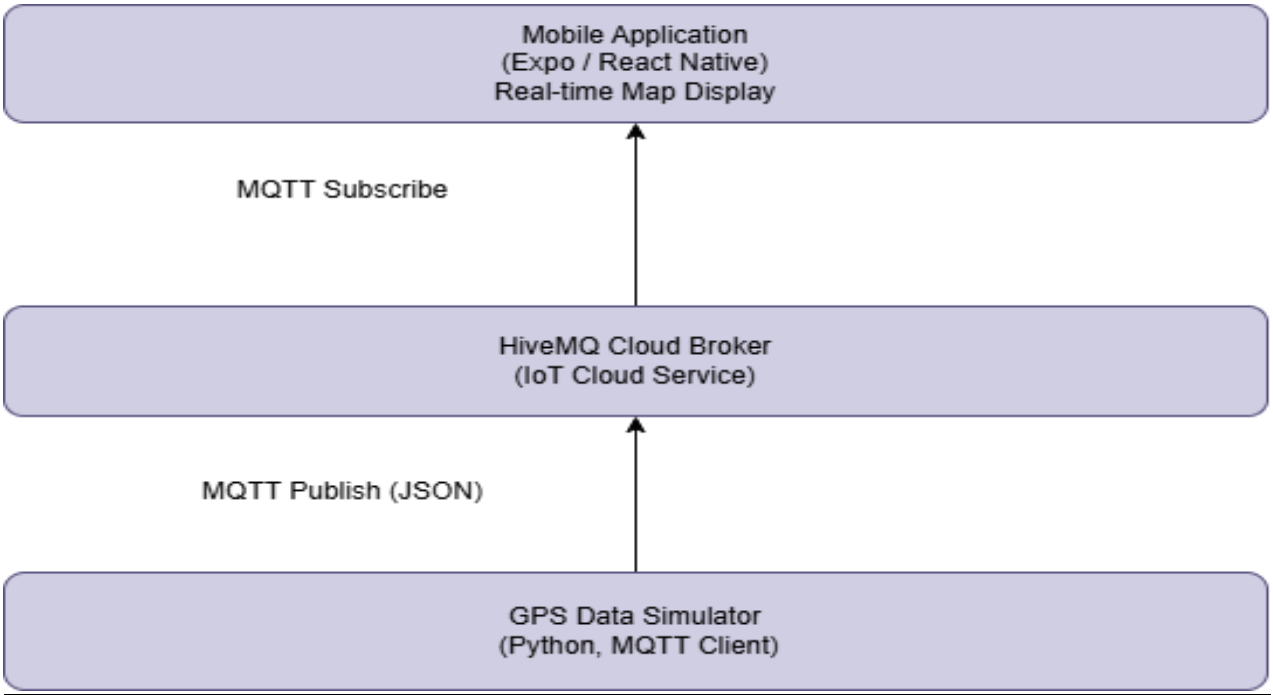
The mobile app is built using Expo and subscribes to the MQTT topic to visualize the GPS trajectory. Its features include:

- Connecting securely to the HiveMQ broker using TLS and authentication.
- Receiving location updates in real time.
- Rendering the trajectory on an interactive map, allowing users to track movement along the predefined path.

2. Architecture system

The system uses a cloud-based IoT architecture built on the MQTT publish/subscribe model to enable real-time GPS tracking. It is organized into three layers: a simulated device layer that generates location data, an IoT cloud service layer that handles secure message delivery, and an application layer responsible for data visualization.

Location data is generated by a Python-based simulator and transmitted via HiveMQ Cloud to a mobile application developed with Expo and React Native. The cloud layer ensures secure, reliable, and real-time communication, while the mobile application subscribes to the MQTT topic and displays the trajectory dynamically on an interactive map.



Here is a brief summary of the technologies used :

GPS data simulation	Python 3.x with jupyter notebook environment
IoT Cloud Platform (MQTT Broker)	HiveMQ Cloud
Mobile application	React Native Expo

3. User Guide

This guide explains how to install, configure, and run the GPS Tracking System locally.

3.1. Requirements

To run the project successfully, the following tools and environments are required.

General	Git (to clone the repository)
For mobile Application	Android phone
For Expo code	Node.js, Yarn, Expo CLI
GPS Data Simulator	Python 3.x, Jupyter Notebook

3.2. Execution steps

Step1. Clone the GitHub repository :

```
git clone https://github.com/Tsanta04/gps-tracker.git
```

Step2. Navigate to the project directory :

```
cd gps-tracker
```

Step3. Verify the project structure :

```
gps-tracker/
├── APK/              # Android APK file of the mobile application
├── AppCode/          # Expo (React Native) mobile application source code
├── Documentation/    # Project documentation and reports
└── SimulationCode/   # Python-based GPS data simulator (Jupyter Notebook)
```

Step4. Run the code of Data Simulation

- Navigate to the SimulationCode directory and launch Jupyter Notebook :

```
cd SimulationCode

jupyter notebook
```

- Select Send.ipynb to be opened to the notebook :

```
Send.ipynb
```

- When you open the Jupyter Notebook, you will see the code. Make sure that the connection settings are defined in the notebook variables:

```
- Broker: `e792f3c61e8648d2bd1822c6d1910301.s1.eu.hivemq.cloud`

- Port: `8883` (TLS)

- Topic: `iot/device001/location`

- Device ID: `device001`
```

- Run all cells including the dependency:

```
!pip install paho-mqtt
```

Step5. Option1 - Use Mobile application

- Download the APK file from the APK folder in the repository.
- Install the application on your Android device.
- Launch the app to connect to the cloud service.
- The GPS trajectory will be displayed in real time on the map.

Option2 – Run the React native Expo Code

- Navigate to the application directory:

```
cd AppCode
```

- Install dependencies with yarn :

```
yarn install
```

- Start the Expo Development Server:

```
npx expo start -c
```

- Run the app on the phone using the Expo Go app by scanning the QR code displayed in the terminal. The GPS trajectory will be displayed in real time on the map.

Conclusion

This project demonstrates the design and implementation of a real-time GPS tracking system using cloud-based IoT services without relying on physical hardware. By integrating a software-based GPS simulator, an MQTT cloud broker (HiveMQ Cloud), and a mobile application built with Expo and React Native, the system effectively illustrates the key principles of Wireless Sensor Networks and the Internet of Things, including data generation, secure transmission, and real-time visualization.

Leveraging the MQTT publish/subscribe model ensures efficient, scalable, and low-latency communication between the simulated device and the mobile client. The cloud layer enables reliable message delivery with security features such as authentication and TLS encryption, while the mobile application provides an interactive interface to visualize device trajectories dynamically in real time.

Despite the limitations of free cloud services and the absence of physical sensors, the project achieves realistic behavior comparable to real-world tracking systems. It highlights the practicality of IoT cloud platforms for rapid prototyping, experimentation, and learning. Moreover, the system can serve as a foundation for future enhancements, including data storage, analytics, multi-device support, or integration with actual GPS hardware, expanding its potential for research and real-world applications.