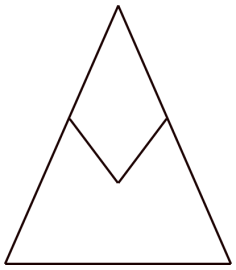
Bern University of Applied Sciences

Project 2

Aplinist Tracker & Alerting System
ATAS



Author:
Martin Schmidli

Teacher:
Mohamed Mokdad

Bern, June 22, 2017

Contents

Chapter 1

Project description

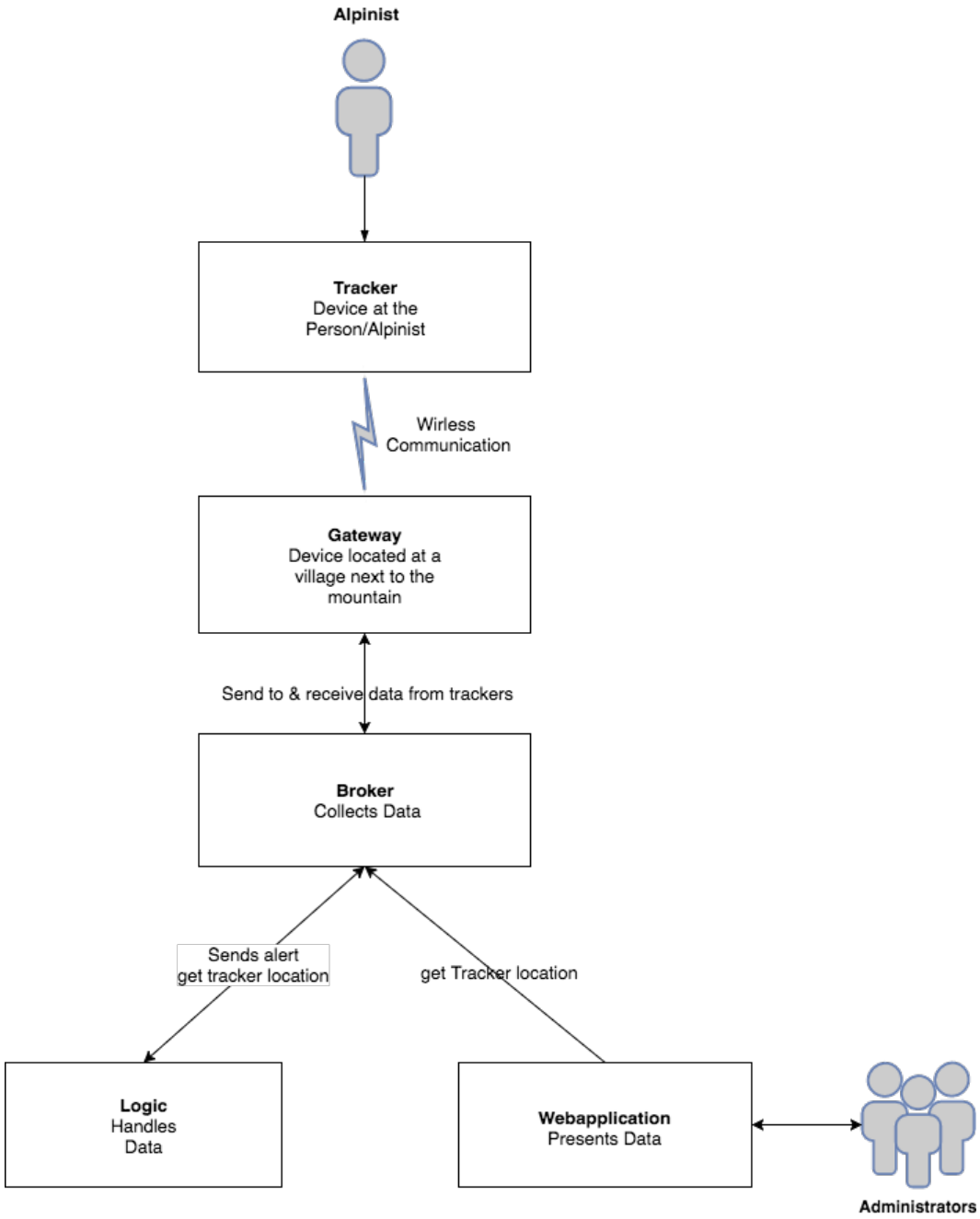
1.1 Introduction

The Alpinist Tracking and Alerting system short ATAS has two main functionalities.

1. Track peobles location which are located on a mountain for example skiers, hikers... short alpinists.
2. Alarm a rescue team if the user had an accident

1.2 Systemarchitecture, Abstract

This section will provide an abstract overview which kind of components are used in the ATAS System.
The sytem will be devided into different modules. All the modules and users of the system are explained on the next page.



1.2.1 System Users

Alpinist

Generalization: Skier, hiker or any other person in the mountains.

Administrators

The administrators of the system are peoble working for a tourism region, hospital or rescue teams like the Rega.

1.2.2 System Modules

Tracker

Imagine a small mobile device, following called **tracker**. The Tracker can be carried by an **alpinists**.

The tracker device is equipped with an **alarm-button** and a **speaker**. With the alarm-button, the alpinist can inform the administrators about an accident. For example if the aplinist broke its knee.

The speaker is used to inform the alpinist, with an alarming sound, that he's situated inside a dangerzone and he would better leave this zone immediately. A dangerzone is a place with a high possibility for an accident, for example, an area with a high possibility for an avalanche.

The tracker is sending its GPS position to a receiver station, following called **gateway**.

Gateway

A device which is able to talk to the trackers. The gateway can be installed in a village/ building in the valley next to the mountains.

All the gateway are sending the received locations from the trackers to a central datamanager called **broker**.

Broker

The broker collects data and stores them. The Broker provides an Interface where we can grab all the data and use them in a webapplication.

Webapplication

The administrators can use a **webapplication** to

- Monitor the actual position of the persons on the mountains.
- Define dangerzones

Logic

Software / service which calculates if a tracker is inside a dangerzone. If the tracker is inside a dangerzone it will send an alarm to the tracker. The alermsignal is relayed over the Broker.

1.3 Usecase

So why is the ATAS system useful? The ATAS system can be used for:

- If an avalanche is triggered in the mountains, its position can be compared to the position of the trackers. If at the time of the incident, a tracker was in this area, the rescue teams can be sent to do a rescue operation. This immediate action will give the victims important seconds and may save lifes.
- If peoble were in the avalanche and the tracker survived, the device could keep sending data and support the rescue team in their mission bye pointing them to the persons. If no transmission is possible through snow, the rescue team has at least his last GPS location, this might reduce the amount of time to find the person. ATAS is not planned as a replacement for Avalanche transceivers, it should be seen as a supplement.
- If a tracker is moving towards a dangerzone, for example an area with rockfall, the user can be informed with the tracking device.
- If an alpinist had an accident in the mountains, the person can send a rescue signal with the tracker device by triggering the alarm-button.

1.4 Technologies

At first glance this hole system must look very laborious to a user. Why should a person have to take another device with him on his hikes? He could easily use his smartphone and an App to send his location. This statement is true if we consider only the touristic ski areas of switzerland. In most ski areas the phone reception is wonderful. If we leave those ”safer” areas for higher altitude, the reception gets worse or disappears completely.

This is the reason why for this project another technologies had to be found.

For this project the following technologies have been used:
MQTT, LoRA, LoRaWAN

You should get yourself familiar with these technolgies to fully understand the following chapters.

1.5 Demarcation

This section defines which kind of tasks are to do during this project

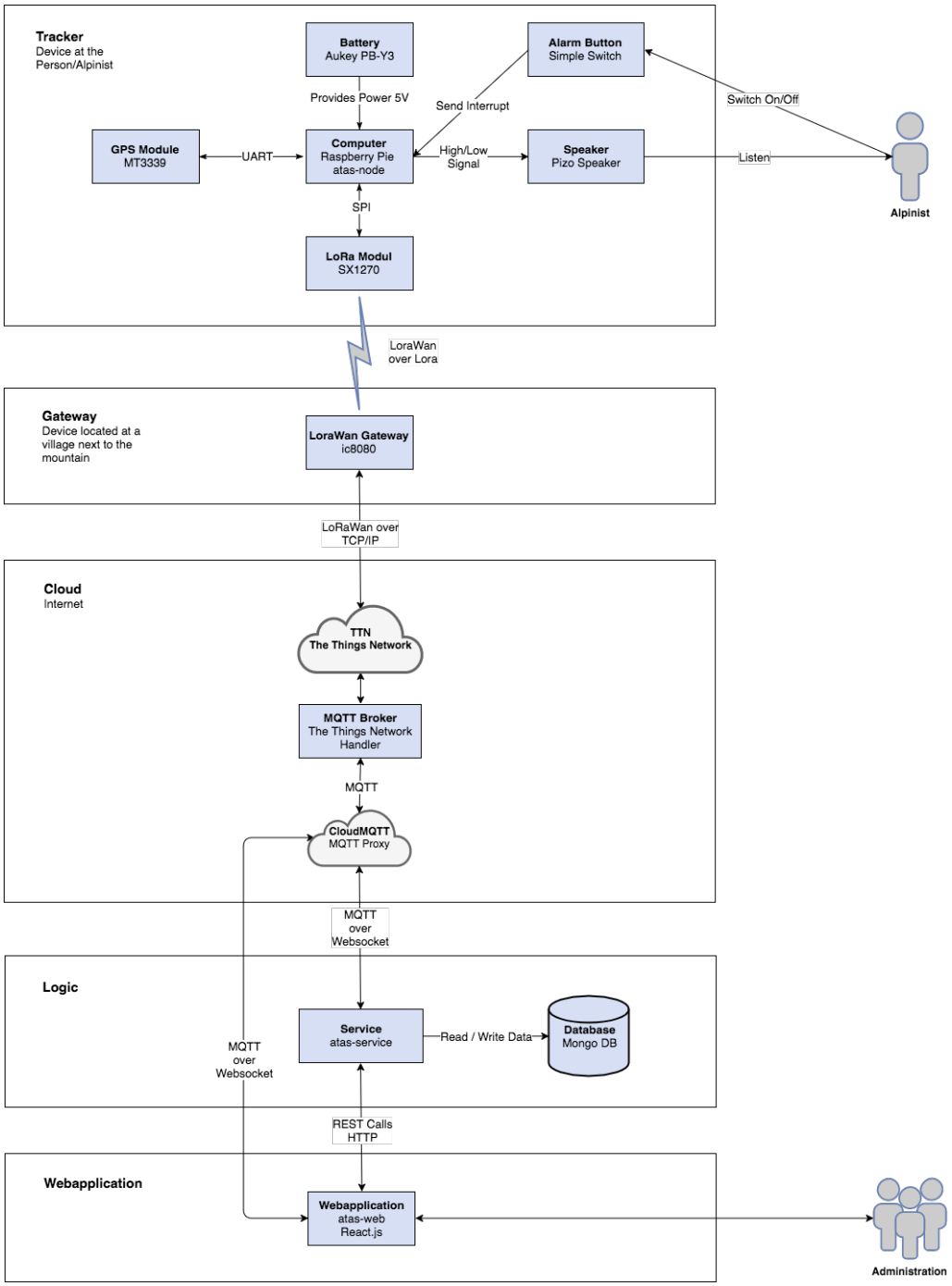
- Build a tracking device
  - The user should be able to trigger a manual alarm and inform the rescue team of an accident
  - Tracks the position of the user and sends the data to a data collector
- Build a webpage, which enables the administrators to
  - view the location of the trackers
  - manage dangerzones
- Build a software / service which controls the position of the trackers. If a tracker dwells inside a dangerzone the tracker has to be alerted
- Connect all the modules
- Setup a broker system

Chapter 2

Systemarchitecture, Detail

This chapter will provide a detailed overview how the ATAS System was implemented. It will be more technical and show in detail how the modules of ATAS interact with each other.

2.1 Diagramm


Tracker
Device at the Person/Alpinist
Battery
Aukey PB-Y3
Alarm Button
Simple Switch
Provides Power 5V
Send Interrupt
Switch On/Off
GPS Module
MT3339
UART
Computer
Raspberry Pie
atas-node
High/Low Signal
Speaker
Pizo Speaker
Listen
Alpinist
SPI
LoRa Modul
SX1270
LoraWan over Lora
Gateway
Device located at a village next to the mountain
LoraWan Gateway
ic8080
LoRaWan over TCP/IP
Cloud
Internet
TTN
The Things Network
MQTT Broker
The Things Network Handler
MQTT
CloudMQTT
MQTT Proxy
MQTT over Websocket
Logic
Service
atas-service
Read / Write Data
Database
Mongo DB
MQTT over Websocket
REST Calls HTTP
Webapplication
Webapplication
atas-web
React.js
Administration

2.2 Tracker

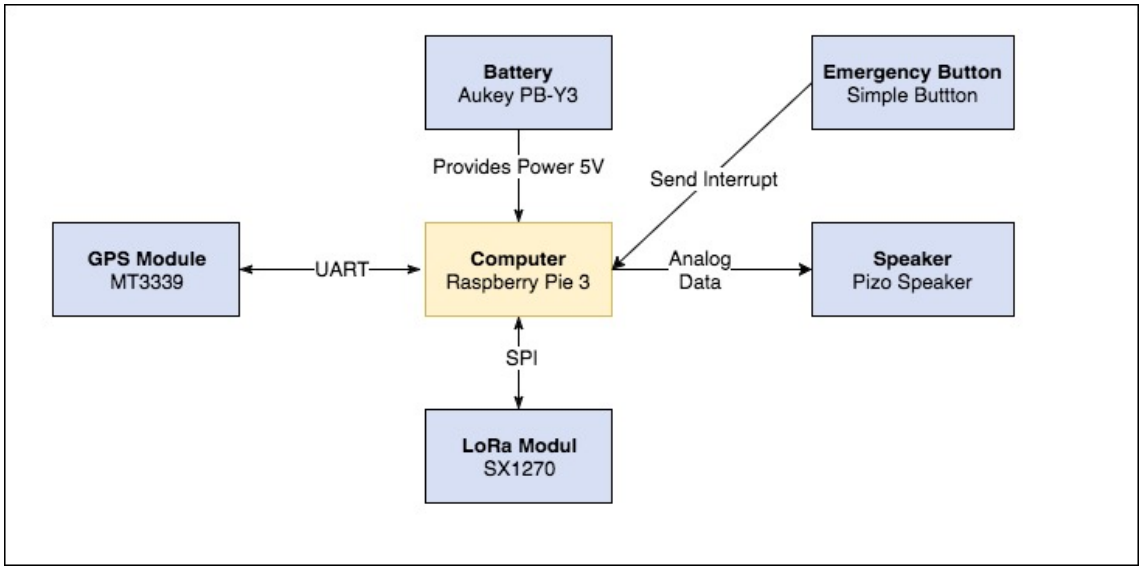
The tracker itself contains differnet components.

2.2.1 Computer

As my knowledge for hardware close porgramming (c, c++ ...) is failry limited, I decided to use a hardware plattform, which handles most of the system tasks for me. The computer had to be small and doesn't use to much energy. I decided to use a Raspberry Pie 3 single board computer.

Diagramm

The Raspberry Pie is connected to other devices



Tasks

The Raspberry Pie computer will be used to

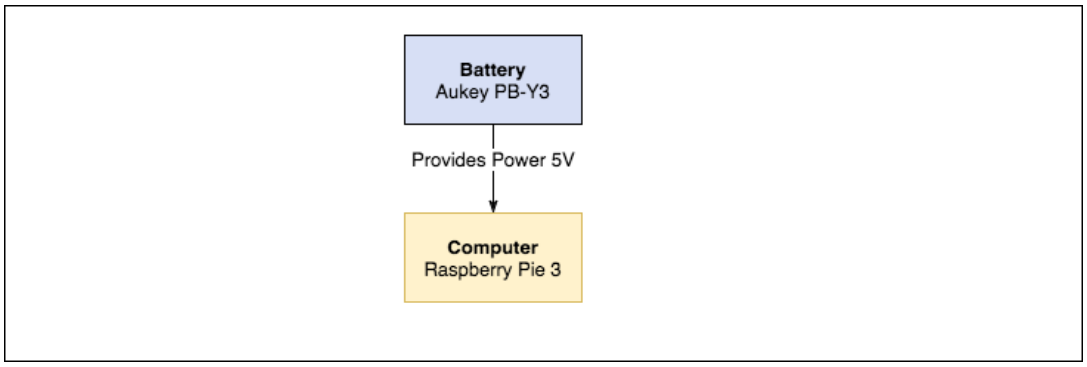
- Recieve GPS data from the GPS module via UART
- Recieve and send data to the Lora Module via SPI
- Listen for signals from the alert-button
- Enables or disables the speaker

2.2.2 Battery

The tracker needs to mobile so we can put it in our backpack. To power the tracker I attached a battery pack over USB.

Type: Aukey PB-Y3 Battery Pack
Power: 30000mAh
Physical Interface: Attached to the Raspberry Pie over Micro USB.

Diagramm



2.2.3 GPS Module

The GPS Module provides the GPS data to the tracker.

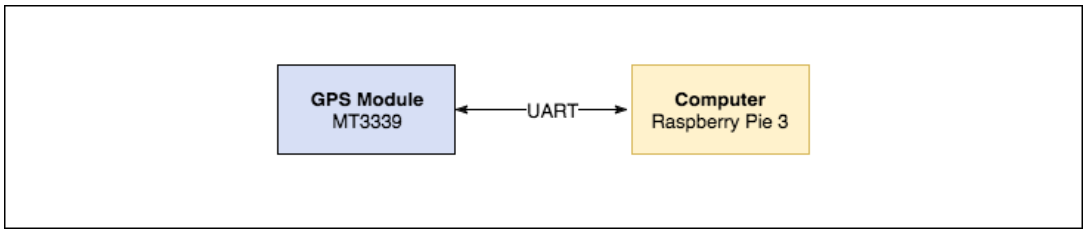
Type: MT3339
Interface: Acessable via UART.

Tasks

The GPS module will be used to

- Get the GPS Location(Longitude, Latitude) of the Tracker
- Sends GPS data to the Raspberry Pie via UART

Diagramm

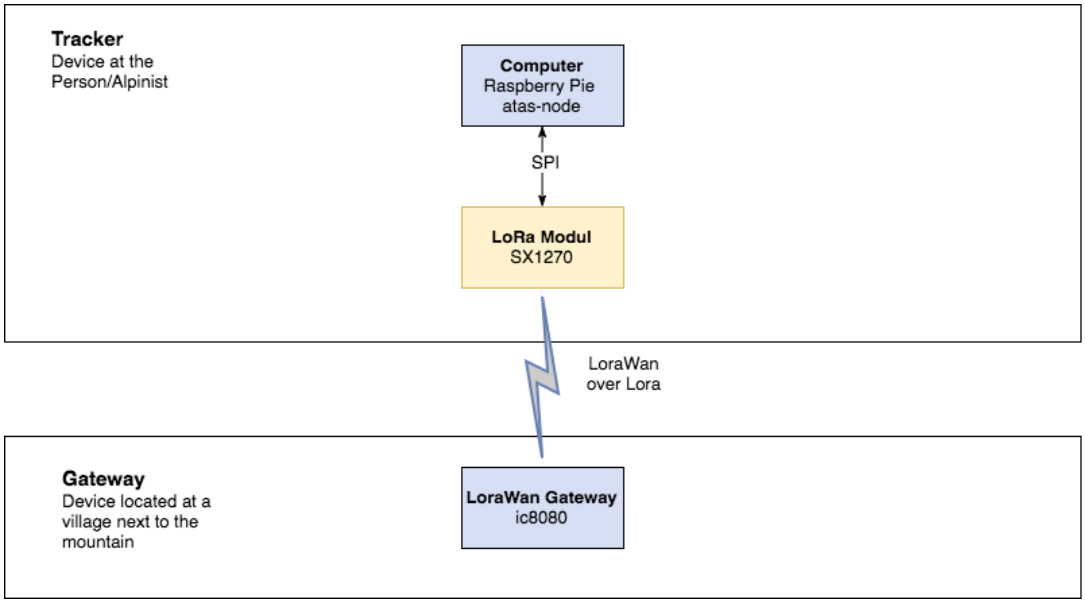


2.2.4 LoraWan Modul

The LoraWan Modul establishes the wireless connection to the Gateway.

Type: Lora Module Semtech SX1276
Interface: Acessable via SPI

Diagramm



Tasks

The Lora module will be used to

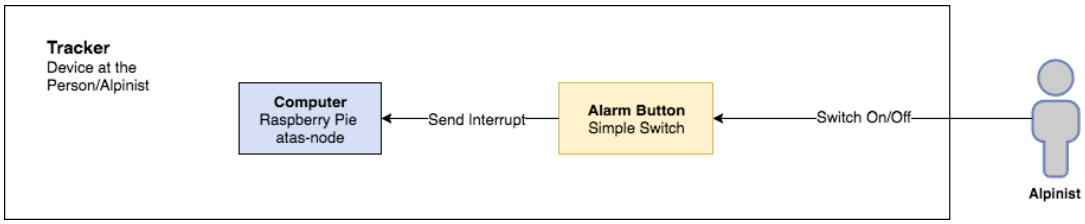
- Receive and transmit data to LoRaWan gateway.
- Receive and transmit data to the Raspberry Pie.

2.2.5 Alarm Button

Simple Switch.

- Set input on Raspberry Pie GPIO Pin on Low (Alert) or High (no Alert)
- Gives the Alpinist the possibility to activateand deactivate the alarm

Diagramm



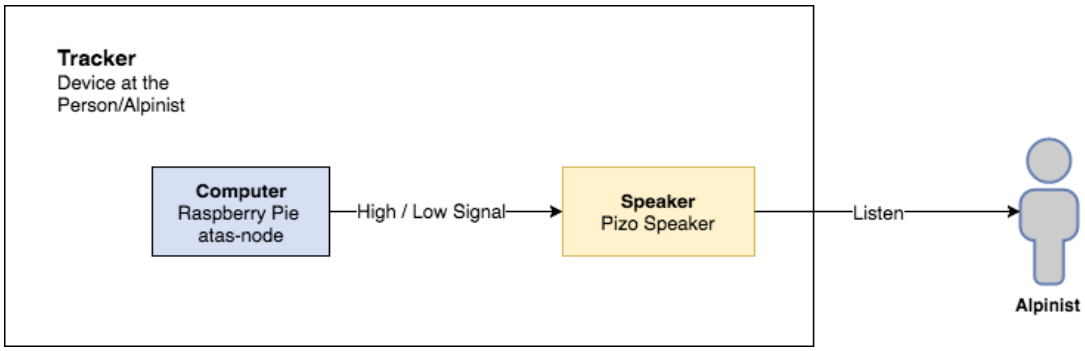
2.2.6 Speaker

Simple Piezo Speaker

Tasks

- Gets enabled by the Raspberry Pie
- Generates an alamring signal, which indicates, that the tracker is inside a dangerzone.

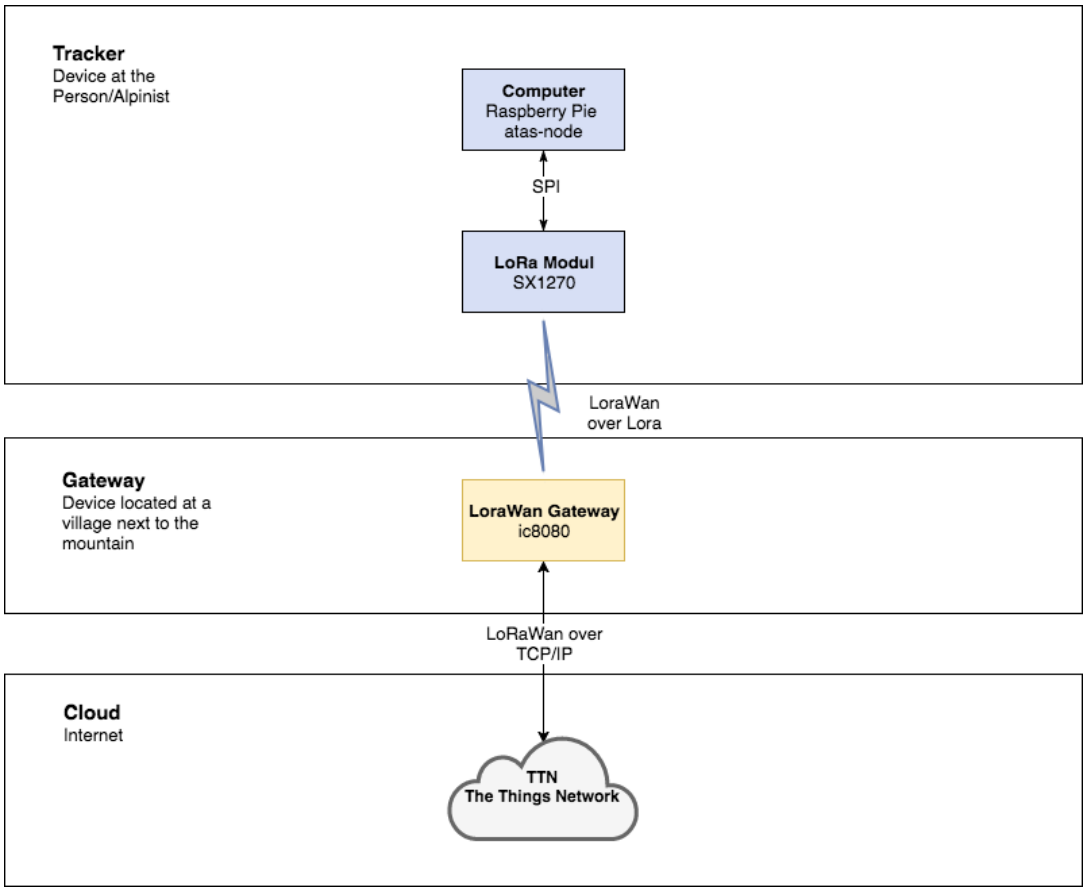
Diagramm



2.3 Gateway

The gateway relays data between the trackers and The Things Network (TTN). What kind of fucntionality TTN is providing is explained later in this docuent.

Diagramm



Tasks

The Lora Gateway will be used to

- Receive and transmit data to the Cloud Network The Things Network (TTN) over LoRaWan via TCP/IP
- Sends and receive datat to the Trackers over LoRaWan via Lora

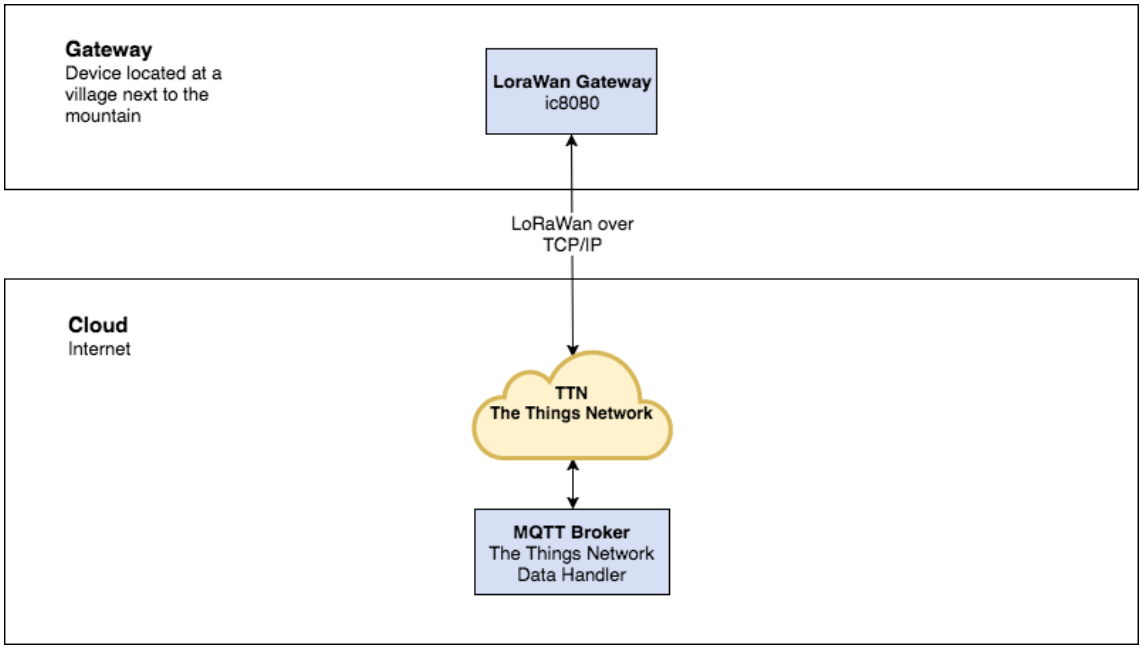
2.4 Broker

The broker has been split up in subcomponents. The broker service hasn't been developed by myself. I use two platforms the Things Network (TTN) and CloudMQTT.

2.4.1 The Things Network short TTN

The Things Network short TTN is an internet platform whichs allows us to connect LoRaWan devices. TTN will send (Downlink) and receive (Uplink) data from the trackers via gateway. TTN will collect and store all the data which the gateways send to them. TTN provies a MQTT Broker to access the data more easily.

Diagramm



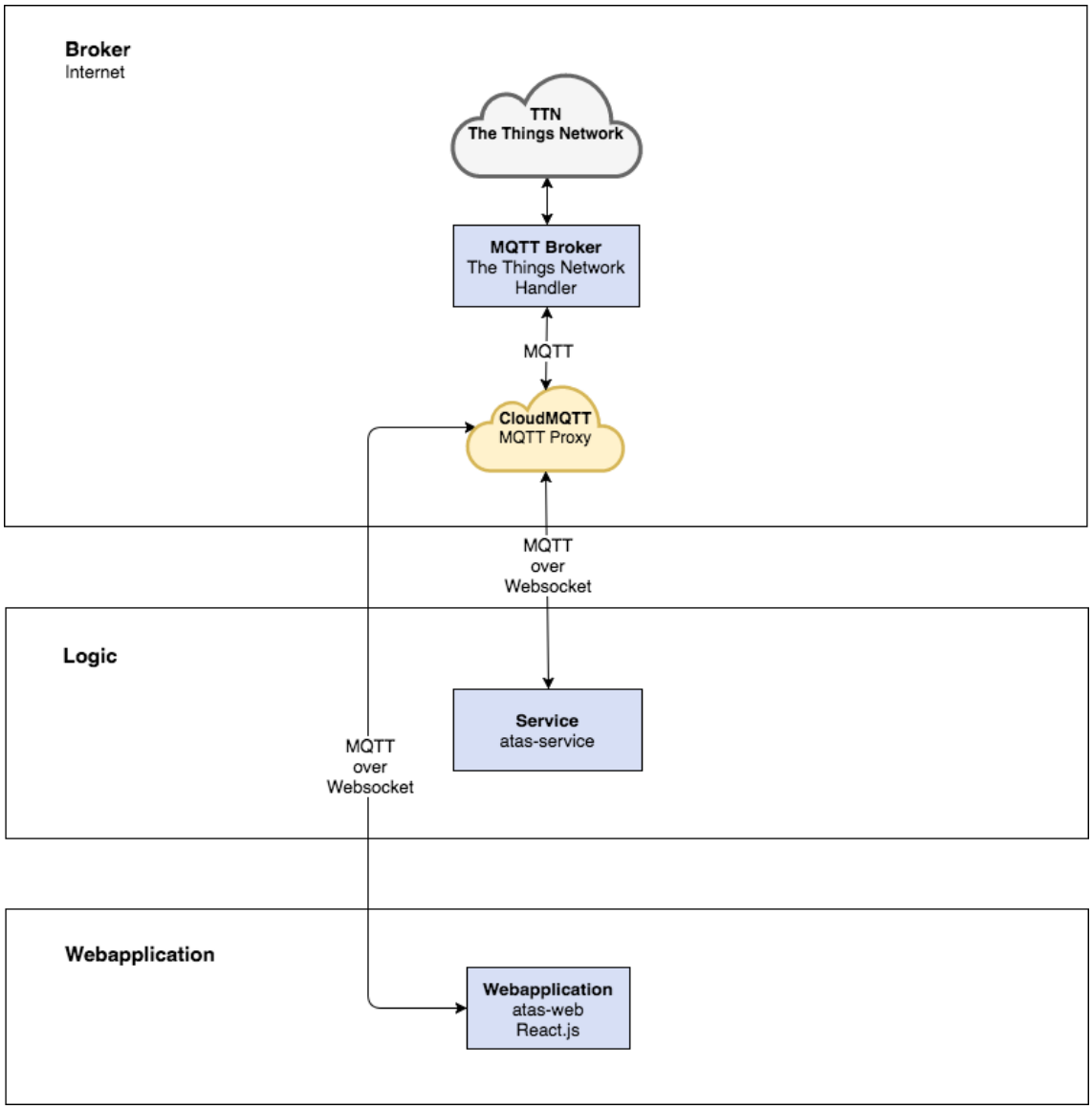
Tasks

- Manage trackers
- Manage gateways
- Provides MQTT Broker
- Send and Receive data from the trackers via gateway

2.4.2 CloudMQTT

To this date, the TTN MQTT Broker does not provide the possibility to access the data via websocket Interface. To bring MQTT to a Web Application, another service was needed. I decided to setup CloudMQTT. CloudMQTT mirrors the TTN MQTT Broker (Topics) and provides an Websocket Interface.

Diagramm



Tasks

- Subscibes and pusblishes to all Topics on the TTN MQTT Broker. It is mirroring the TTN MQTT Broker.
- Provides Websocket Interfaces so we can connect from our applications via MQTT over Websocket.

2.5 Software

2.5.1 Atas-Service

Atas-Service calculates if a tracker is inside a dangerzone. The service is comparing the GPS location of the tracker with the geographical area of the dangerzone. If the tracker is inside a dangerzone it will send an alarm to the tracker. The alarmsignal is relayed over the Broker.

Tasks

- Provide a Rest API to Access Data: Dangerzones
- Calculates Data example: Is a User in a danger zone...
- Publish data to the MQTT Broker
- Read/write to the Database

Database

Mongo DB Database.

Tasks

- Stores Data: Dangerzones

2.5.2 Atas-Web

React.js Webapplication

Tasks

- MQTT communication with the Cloud MQTT Broker (Tracker, GPS...)
- Rest communication with Atas-Service to Fetch Dangerzones
- Shows Tracker on a Map

2.5.3 Atas-Node

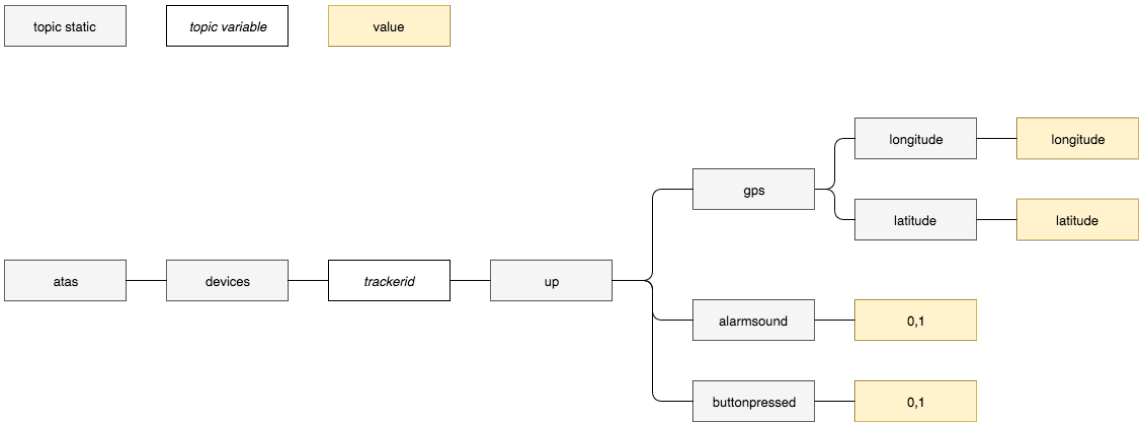
Chapter 3

Data

3.1 TTN MQTT

The following structure

3.1.1 Topics



Chapter 4

Attachments

4.1 Code

All code written for this projects hase been uploaded to github. No code will be attached to this documentation. Follow the links provided to access the code.

4.1.1 Atas-Webapp

`https://github.com/schmm2/atas-webapp`

4.1.2 Atas-Node

`https://github.com/schmm2/atas-node`

4.1.3 Atas-Service

`https://github.com/schmm2/atas-service`