



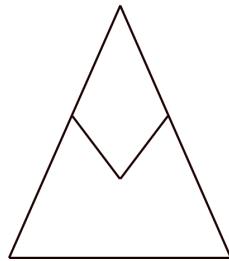
BERN UNIVERSITY OF APPLIED SCIENCES

BACHELOR THESIS  
INFORMATIK

---

Alpinist Tracker & Alerting System  
**ATAS**

---



*Autor:*  
Martin SCHMIDLI

*Dozent:*  
Mohamed MOKDAD

*Experte:*  
Daniel VOISARD

Bern, 31. Dezember 2017

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>6</b>
1.1 Vorarbeiten . . . . .	6
1.2 Aufgabenstellung . . . . .	6
1.3 Rahmenbedingungen . . . . .	7
1.4 Technologien . . . . .	7
1.5 Anwendungsfälle . . . . .	7
<b>2 Systembeschreibung - Grob</b>	<b>9</b>
2.1 Grobe Systemarchitektur . . . . .	9
2.2 Aufbau . . . . .	10
<b>3 Systembeschreibung - Detail</b>	<b>13</b>
3.1 Tracker . . . . .	15
3.2 Gateway . . . . .	16
3.3 Broker . . . . .	17
<b>4 Technologien</b>	<b>19</b>
4.1 LoraWan . . . . .	19
<b>5 Prototyp</b>	<b>23</b>
5.1 Ablauf Vorgehen . . . . .	23
5.2 Begründung neuer Prototyp . . . . .	25
5.3 Ablauf Prototyping . . . . .	26
5.4 Evaluation Hardware . . . . .	27
5.5 Zusammenbau Hardware . . . . .	28
5.6 Software Module . . . . .	32
5.7 Finalisierung . . . . .	33
<b>6 Testing</b>	<b>34</b>
6.1 Ablauf Vorgehen . . . . .	34
6.2 Definition . . . . .	37
6.3 Standort . . . . .	37
6.4 Messungen . . . . .	37
6.5 Auswertung . . . . .	37

6.6 Software . . . . .	39
------------------------	----

# Management Summary

# Motivation

# Danksagung

# Kapitel 1

## Einleitung

Stellen Sie sich ein kleines mobiles Gerät vor, nachfolgend Tracker genannt, welches Skifahrern, Wanderer usw. abgegeben werden kann. Das Gerät sendet die Position der Person an einen Empfänger, nachfolgend Gateway genannt. Der Gateway wird bei der Talstation oder im nächsten Bergdorf montiert. Der Gateway sendet die empfangenen Daten der Tracker an eine zentrale Stelle irgendwo im Internet. Die Administratoren des Systems, beispielsweise die Rega, können schlussendlich über eine Webseite die aktuelle Position der Personen in den Bergen mitverfolgen und überwachen.

### 1.1 Vorarbeiten

Das Projekt 2 ist eine Projektarbeit welche im Laufe des Informatikstudiums absolviert werden muss. Diese Arbeit habe ich im vorhergehenden Semester (Frühlingssemester 2017) durchgeführt. Ziel dieses Projektes, war es das in der Einleitung beschriebene System technisch umzusetzen.

Diese Bachelor Thesis baut auf den Erkenntnissen und Resultaten aus dem Projekt 2 auf.

### 1.2 Aufgabenstellung

Ziel der Bachelorarbeit soll es sein, dass während dem Projekt2 aufgebaute System intensiv zu testen und zu verbessern. Die Arbeit umfasst die folgenden Aufgaben.

- Der Tracker soll in realer Umgebung getestet werden bspw. während einer Wanderrung. Die Messdaten werden erfasst und analysiert.
- Der Tracker Prototyp soll verbessert werden. Die dazu erforderlichen Massnahmen werden während der Bachelorarbeit spezifiziert und umgesetzt. Am Ende der Arbeit soll eine klare Aussage gemacht werden können ob das erdachte System praxistauglich ist

- Die während dieser Arbeit geleisteten Arbeiten sollen in einem Bericht dokumentiert werden.

### 1.3 Rahmenbedingungen

Die Rahmenbedingungen dieses Projektes wurden gemeinsam mit dem Betreuer definiert

- Die Bachelorarbeit baut auf der Arbeits des Projekt 2 auf. Die erstellte Hard und Software wird weiterverwendet.
- The Things Network dient als Plattform für die Kommunikation unter den Komponenten.
- Als Programmiersprache soll C/C++ verwendet werden
- Auf den Einsatz eines Betriebssystemes auf dem Tracker Node soll verzichtet werden

### 1.4 Technologien

Sie als Leser denken nun vielleicht: "Das ist überflüssig. Ich habe doch mein Handy für so etwas? Schreib doch eine App!". Diese Aussage mag auf die touristischen Wander- und Skigebiete zutreffen. Der Empfang ist meistens wunderbar. Verlassen wir diese 'sicheren' Orte und bewegen uns aber in höheren Lagen, wird der Empfang mit dem Handy immer schlechter oder existiert gar nicht.

Aus diesem Grund mussten für dieses Projekt andere Technologien gefunden werden. In diesem Projekt werden folgende Technologien eingesetzt: MQTT, LoRa, LoRaWAN

Die verwendeten Technologien und der Grund für deren Einsatz werden in einem späteren Abschnitt genauer erklärt.

### 1.5 Anwendungsfälle

Dieses Abschnitt soll Ihnen erläutern, wie das ATAS System genutzt werden kann.

- Wenn in den Bergen eine Lawine ausgelöst wird, kann deren Position mit den Positionen der Tracker verglichen werden. Befindet sich ein Tracker in der Gefahrenzone, kann die Rettungsmannschaft sofort reagieren und ausrücken. Dieser Prozess läuft sofort ab und kann dabei die Überlebenschance der Opfer erhöhen.
- Wurden Personen von der Lawine begraben und der Tracker hat überlebt, könnte das Gerät weiterhin die Position senden. Kombiniert mit modernen Lawinensuchgeräten können die Einsatzkräfte gezielter nach Überlebenden suchen. Wenn keine

Übertragung mehr möglich ist, wissen die Überwacher zumindest den letzten Aufenthaltsort. Das ATAS System hat nicht das Ziel Lawinensuchgeräte zu ersetzen, es ist eher als Ergänzung zu verstehen.

- Bewegt sich Tracker auf eine Gefahrenzone zu, beispielsweise ein Gebiet mit erhöhter Steinschlaggefahr, könnte die Person frühzeitig davor gewarnt werden.
- Ist es zu einem Unfall gekommen, kann der Alpinist mittels Tracker ein Notsignal absetzen. Dazu muss der Benutzer nur auf den Notfallknopf drücken.

# Kapitel 2

## Systembeschreibung - Grob

Um die nachfolgenden Kapitel zu verstehen, ist es sehr wichtig, sich mit der im Projekt 2 erstellten Systemarchitektur und die Terminologie vertraut zu machen. Die Architektur wurde im Vergleich zum Projekt 2 um einige Komponenten ergänzt.

### 2.1 Grobe Systemarchitektur

Dieser Abschnitt bietet Ihnen einen groben Überblick über die Benutzer, die Komponenten und deren Beziehung innerhalb des ATAS Systems. Alle Einheiten werden auf den nachfolgenden Seite detailliert beschrieben.

## 2.2 Aufbau

### 2.2.1 Diagramm

Grobes Schema des ATAS Systems. Auf der nachfolgenden Seiten sind die einzelnen Komponenten im Detail beschrieben.

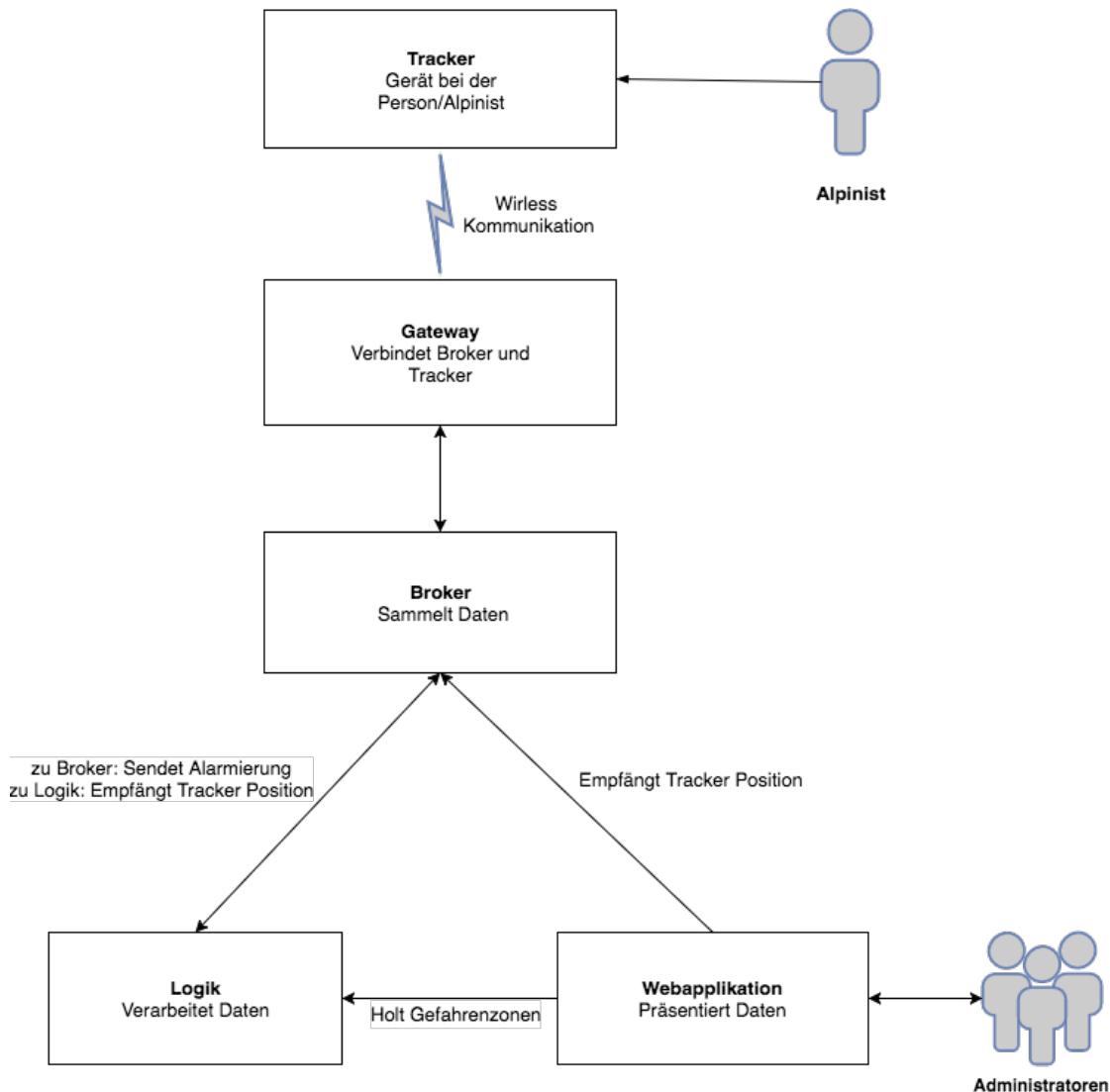


Abbildung 2.1: Atas Schema Grob

## 2.2.2 Benutzer

Die nachfolgenden Gruppen wurden als Benutzer des Systems identifiziert.

### Alpinist

Alpinist wird als Generalisierung für Personen, welche sich in den Bergen aufhalten verwendet. Dazu gehören bspw. Wanderer, Skifahrer, Bergbauern usw.

### Überwacher

Rettungsdienste wie bspw. die Rega oder die AirGlacier. Spitäler oder die lokalen Tourismusbehörden.

## 2.2.3 Komponenten

Das ATAS System besteht aus den nachfolgenden Komponenten. Komponenten sind als Hardware und Software zu verstehen.

### Tracker

Stellen Sie sich ein kleines mobiles Gerät vor, nachfolgend genannt Tracker. Der Alpinist trägt den Tracker bei sich bspw. in einem Rucksack.

Der Tracker verfügt über ein Display, Taste, GPS Modul, Kommunikationsmodul (Lora) und ein Lautsprecher.

- Taste: Mit dem Druck auf den Knopf können die Überwacher über eine Notsituation aufmerksam gemacht werden. Bspw. Wenn der Alpinist einen Unfall hatte und nun bewegungsunfähig ist.
- Lautsprecher: Über den Lautsprecher kann der Alpinist über eine Gefahrenquelle mit einem akustischen Signal aufmerksam gemacht werden. Bspw. Wenn sich der Alpinist in einem Bereich am Berg mit erhöhter Gefahr für Lawinen aufhält. Der Lautsprecher dient zur Information, **dass ein Problem besteht**.
- Display: Über ein Display können mehr Informationen zum Tracker und den Gefahrenzonen angezeigt werden. Das Display dient zur Information, **was für ein Problem besteht**.
- GPS Modul: Der Tracker verfügt über ein GPS Modul. Mit dem GPS Modul kann die Position des Tracker auf der Erde ermittelt werden.
- Lora Modul: Mittels Lora Modul können Daten an einen Empfänger, nachfolgend **Atas-Gateway** genannt, gesendet werden.

## **Gateway**

Ein Gerät welches mit den Trackern bidirektional kommuniziert. Der Gateway wird an einem Ort nahe den Bergen installiert. Als mögliche Orte kommt ein Dorf im Tal oder eine Talstation in Frage.

Der Gateway senden die empfangenen Daten an eine zentrale Datenmanager, nachfolgend **Broker** genannt.

## **Broker**

Sammelt und speichert Daten. Der Broker kommuniziert mit den Gateways. Zusätzlich bietet der Broker ein Interface zum Abfragen und senden von Daten.

## **Webapplikation**

Die Notfalldienste resp. Überwacher nutzen eine Webapp zum interagieren mit dem ATAS System. Die Webseite bietet folgende Funktionalitäten

- Zeigt die Position der Tracker auf einer Karte an
- Mittels der Webapp können Gefahrenzonen hinterlegt werden.
- Zeigt an ob sich ein Tracker in einer Gefahrenzone aufhält.

## **Business Logik**

Die BusinessLogik umfasst Software welche berechnet, ob sich ein Tracker in einer Gefahrenzone aufhält. Wenn sich der Tracker in einer Gefahrenzone aufhält, senden das System eine Alarm an den Tracker. Das Signal gelangt via Broker und dem Gateway zum Tracker.

## Kapitel 3

# Systembeschreibung - Detail

Die vorhergegangene grobe Architekturbeschreibung gibt einen guten ersten Überblick über das System. Technische Details zu den verwendeten Technologien oder den genauen Fluss der Kommunikation ist nicht ersichtlich. In diesem Abschnitt soll nun intensiv auf diese Thematik eingegangen werden.

Auf der nachfolgenden Seite finden Sie ein Schema welches einen genauen Überblick über das System liefert.

Die Komponenten werden in 2 Kategorien eingeteilt. Komponenten mit einem blauen Kästchen wurden im Rahmen dieser Arbeit selbstständig programmiert. Gelbe Kästchen markieren einen bestehenden Service welcher für den Aufbau des Systems implementiert wurde.

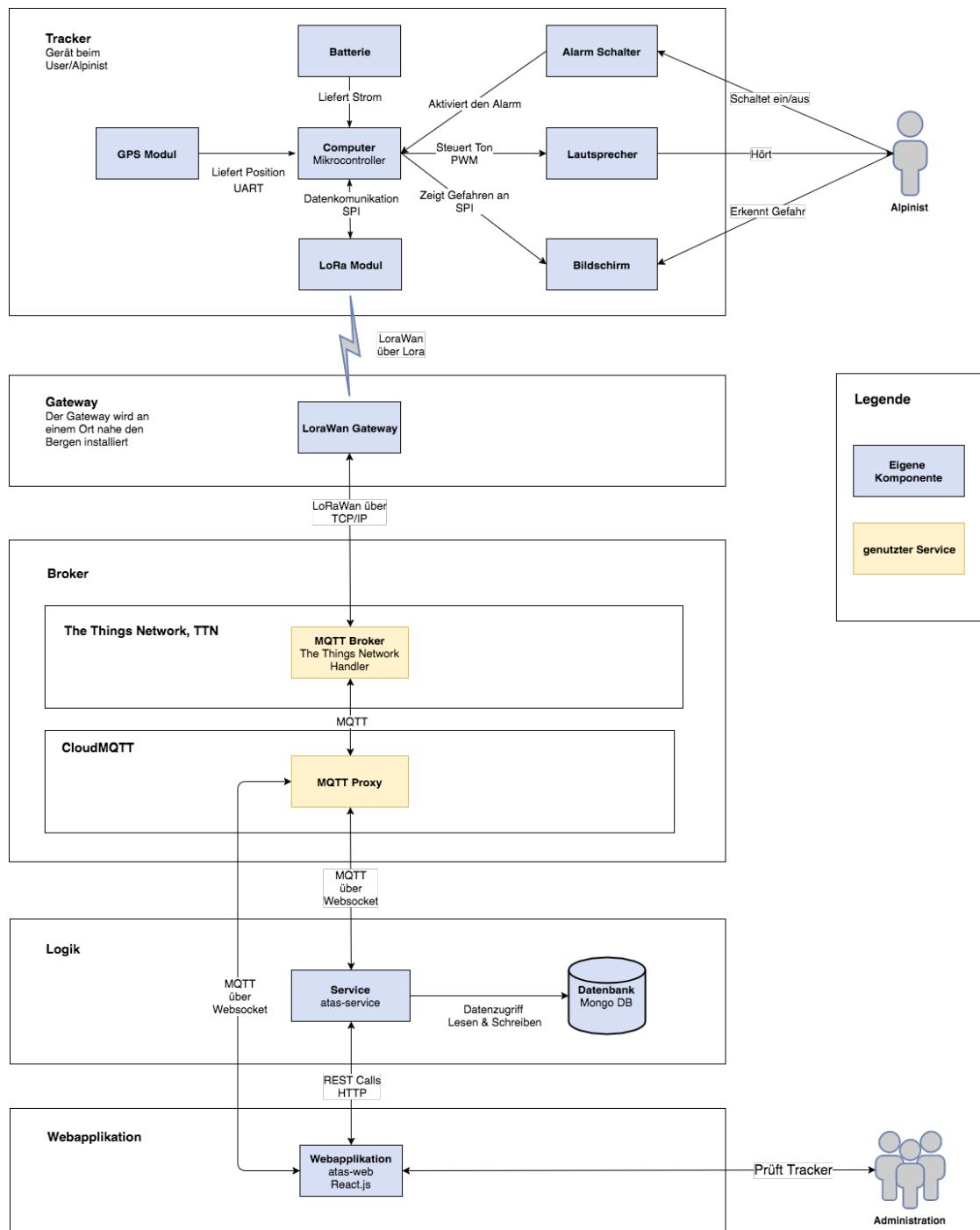


Abbildung 3.1: Systemübersicht Detail

### 3.1 Tracker

Wie in der untenstehenden Grafik ersichtlich, besteht der Tracker aus mehreren Komponenten/Modulen. Welche Komponenten verbaut und wie diese untereinander kommunizieren, wird im Kapitel 'Prototyp' genauer behandelt. Der Tracker besteht aus einem GPS Modul, einer Batterie, einem Notfallknopf, einem Lautsprecher, einem Display und einem Lora Modul.

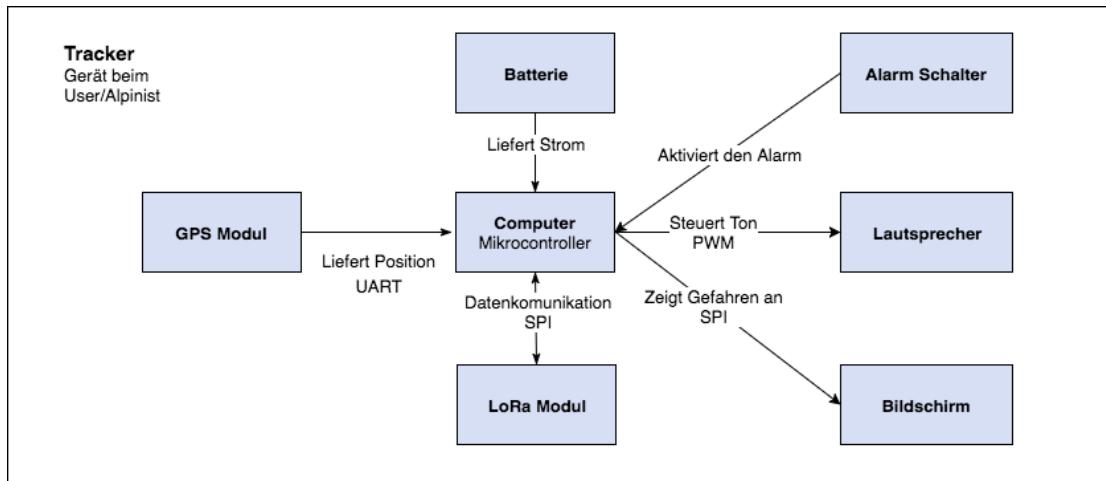


Abbildung 3.2: Tracker Detail

Die Komponenten haben die folgenden Funktionen und Aufgaben

**Mikocontroller:** Steuert die Komponenten

**GPS Modul:** Ermittlung der Position des Trackers via GPS

**Notfallknopf:** Aktiviert den Notfallmodus

**Lora Modul:** Übermittelt die Positionsdaten per Lora und ob ein Notfall vorliegt.  
Empfang von Gefahrenzoneninformationen per Lora

**Display:** Informiert den Alpinist über mögliche Gefahrenquellen

**Lautsprecher:** Informiert den Alpinisten über mögliche Gefahrenquellen mittels Signalton

## 3.2 Gateway

Der Gateway verbindet die Tracker mit dem TTN Netzwerk. Er sendet Daten zu den Trackern (Downlink) und empfängt Daten von den Trackern (Uplink). Was genau das TTN für eine Rolle übernimmt, wird später im Detail beschrieben.

### 3.2.1 Diagramm

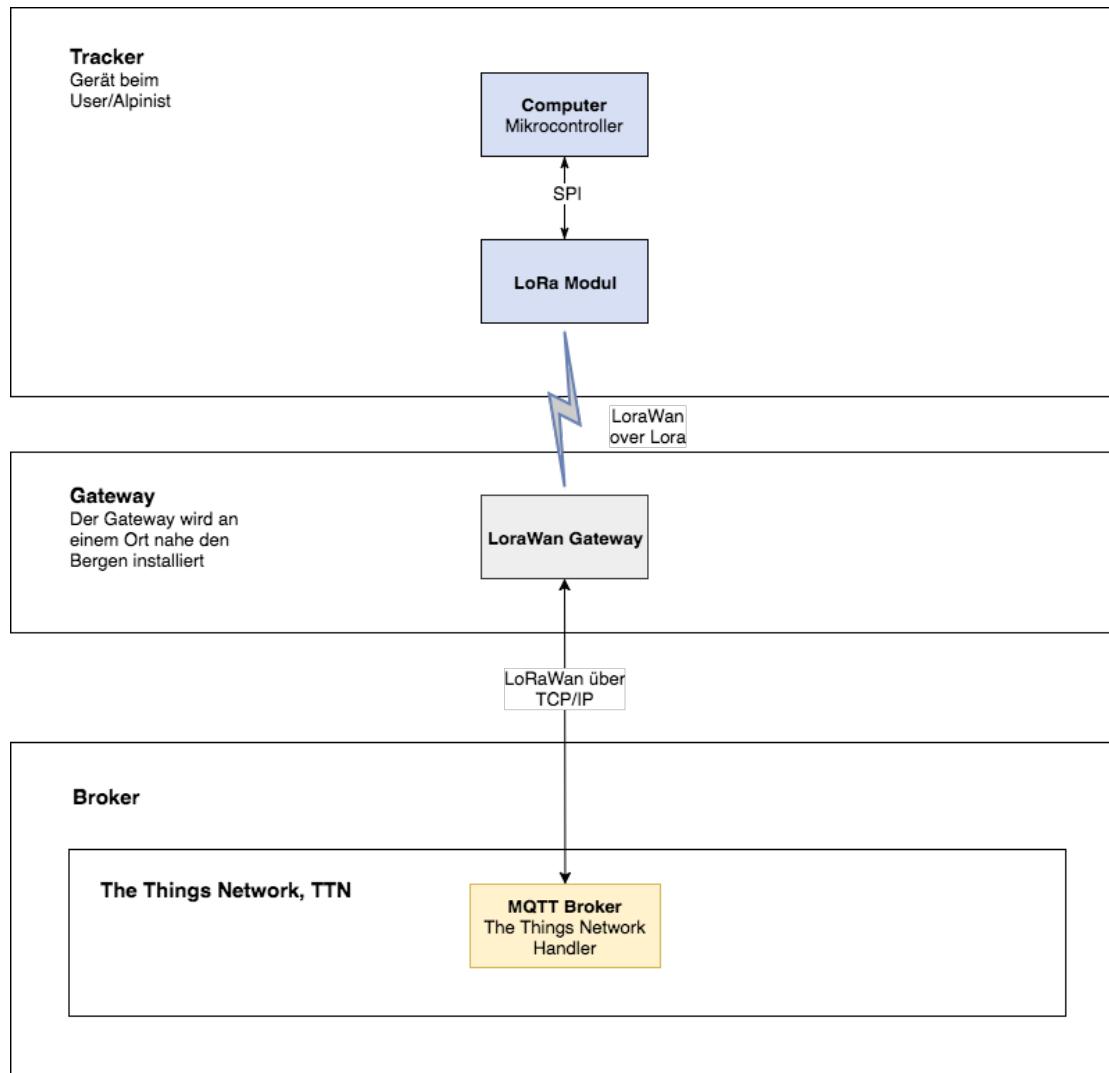


Abbildung 3.3: Gateway Detail

### 3.3 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.

#### 3.3.1 The Things Network (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 it. TTN provies a MQTT Broker to access the data more easily.

#### Diagramm

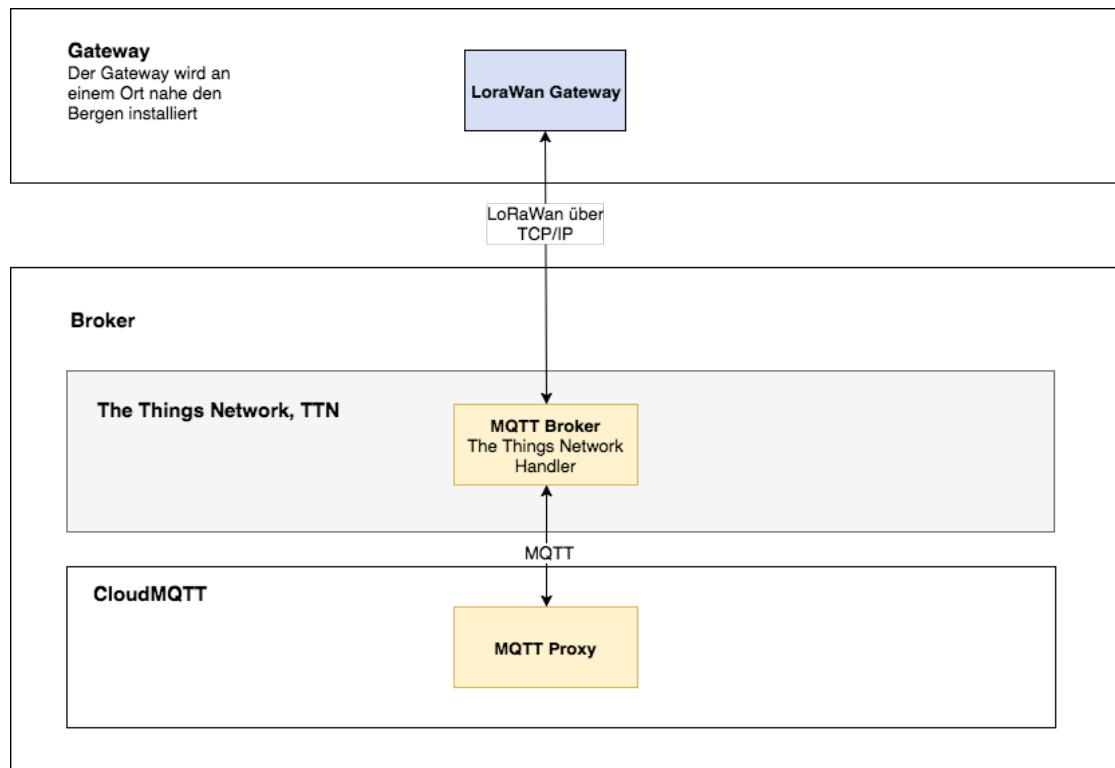


Abbildung 3.4: TTN Detail

### 3.3.2 CloudMQTT

Der MQTT Broker des TTN Netzwerkes bietet nur die Möglichkeiten die Daten via MQTT abzurufen. Ein Zugriff via MQTT über Websockets wird nicht unterstützt. Im Internet fand ich recht schnell den Service 'Cloud' MQTT. CloudMQTT spiegelt den TTN MQTT Broker, bietet aber zusätzlich ein Websocket Interface. Atas-Web und Atas-Service können nun via CloudMQTT Daten an die Tracker senden und empfangen.

Alle Daten welche zu CloudMQTT publiziert werden, werden nun auch automatisch zum TTN MQTT Broker publiziert (Published). Alle MQTT Topics welche vom CloudMQTT abonniert werden, werden auch vom TTN MQTT Broker abonniert (Subscribe).

#### Diagramm

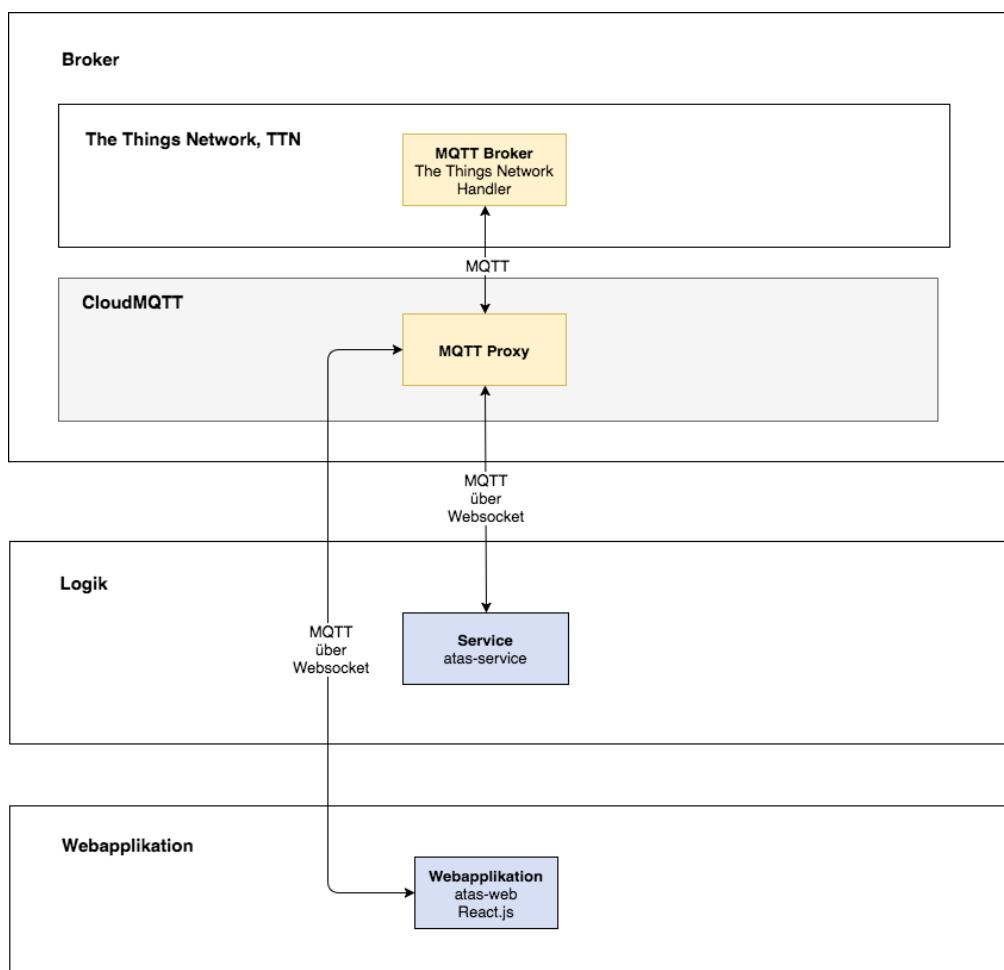


Abbildung 3.5: TTN Detail

# Kapitel 4

## Technologien

### 4.1 LoraWan

Um das Projekt durchzuführen, musste ich mir mehr Wissen zum Thema LoraWan aneignen. Das im Projekt 2 aufgebaute System ist lauffähig, verwendet aber gewisse Standardeinstellungen welche ich einfach übernommen habe. Das Wissen, ob und wie man das System optimieren könnte ist nicht vorhanden.

#### 4.1.1 Begriffserklärung

Zunächst ist es nötig einige Begriffe aus dem LoraWan Umfeld zu verstehen

Begriff	Erklärung
Endgerät	LoraWan Endgerät. Atas-Node ist ein LoraWan Endgerät.
AppEUI	Eindeutige ID einer Applikation. 64Bit lang. Wird genutzt um ein Endgerät einer Applikation zuzuweisen.
DevEUI	Eindeutige ID eines Endgerätes.
DevAddr	Logische Geräte Adresse, analog einer IP Adresse in einem IP Netzwerk.
NetSKey	Wird für die Verschlüsselung der Kommunikation zwischen Endgerät und Netzwerkanbieter bspw. TTN verwendet. Mittels Schlüssel kann die Datenintegrität sichergestellt werden d.H. Manipulierte Nachrichten werden erkannt.
AppSKey	Mittels AppSKey werden die Nutzdaten separat nochmals verschlüsselt. Nur der Nutzer des Netzwerks kann die Daten lesen das Transportnetzwerk bspw. TTN kann diese nicht einsehen.[?]
Downlink	Kommunikation vom Gateway zu Endgerät
Uplink	Kommunikation vom Endgerät zum Gateway

### 4.1.2 Endgerät Verbindungsmethoden

OTAA (Over-The-Air-Activation) und ABP (Activation By Personalization) sind Verbindungsmethoden für LoraWan Endgeräte in einem LoraWan Netzwerk.[2]. Die verwendete Methode definiert also wie sich ein Endgerät mit dem LoraWan Netzwerk verbindet.

Die beiden Methoden bieten verschiedene Vor und Nachteile.

#### ABP

ABP ist das simplere aber **weniger sichere** Verfahren. Die Adresse DevAdr sowie die Schlüssel NetSKey und des AppSKey werden für jedes Endgerät **einmalig** generiert und fix auf dem Endgerät hinterlegt. Gelingt es einem Angreifer sich Zugang zum Endgerät zu verschaffen, kann er die Schlüssel stellen, ein zweites Gerät am Netzwerk registrieren, die Identität des Originals annehmen und damit die Daten verfälschen.

Das Gerät muss sich nicht am Netzwerk anmelden. Das Endgerät kann direkt Daten senden. Empfängt ein Gateway die Daten, werden die Keys geprüft und die Kommunikation entsprechend an oder abgelehnt.

#### OTAA

Das Gerät muss sich am Netzwerk anmelden. Dieser Vorgang wird auch Join Prozedur genannt. Die DevAddr sowie die Keys NetSKey und AppSKey werden bei jeder Aktivierung des Gerätes **neu generiert** und an das Endgerät übertragen. Dieses Verfahren ist sicherer. Da es sich hier um eine bidirektionale Kommunikation handelt, müssen die Komponenten (Gateway & Endgerät) Downlinks unterstützen.

Das Installieren (Deployment) von Endgeräten wird vereinfacht. Das generieren vom AppSKey und NetSKey pro Endgerät entfällt.

Die nachfolgende Grafik zeigt auf, wie der Kommunikationsaufbau via OTAA abläuft [2].

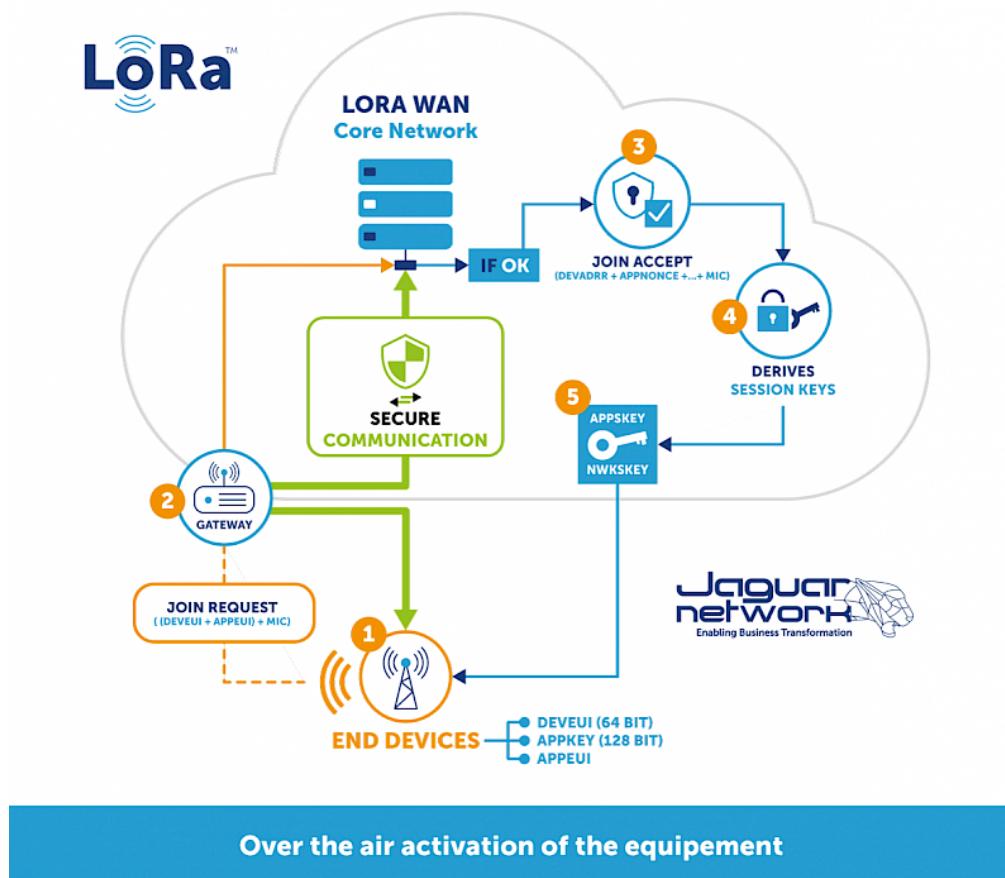


Abbildung 4.1: OTAA Ablauf

1. Gerät sendet einen Join Request.
2. Die Gateways empfangen die Anfrage
3. Das LoRaWan Netzwerk bspw. TTN prüft die Angaben
4. Session Keys werden generiert
5. Keys werden an das Endgerät gesendet und für die zukünftige Kommunikation verwendet

## Schlussfolgerung

Obschon die Sicherheit während diesem Projekt nicht im Vordergrund steht, ist es sicher zukunftsorientierter direkt OTAA einzusetzen. Systeme welche zum Schutz von Personen eingesetzt werden, müssen so sicher wie nur möglich konzeptioniert sein.

Sowohl Endgerät als auch Gateway unterstützen Downlinks, damit gibt es keine technische Hürde der den Einsatz von OTAA verhindern würde. Aus diesem Grund habe ich während dem Projekt die Verbindungsmethodik von ABP auf OTAA umgestellt und konnte damit die Sicherheit des Systems verbessern.

### 4.1.3 Frame Counters

Ein Endgerät besitzt zwei Zähler (FCntUp, FCntDown). Die Zähler werden bei einer Downlinkmessage (FCntDown) resp. Uplinkmessage (FCntUp) erhöht (+1).

Wird eine Reply Attacke durchgeführt d.H. ein Paket nochmals gesendet, wird dieses Paket vom System verworfen. Dies geschieht deshalb, weil das System bereits eine Nachricht mit dem gleichen Framecounter erhalten hat.

Diese zusätzliche Information wird mir beim Testen der Übertragung überaus nützlich sein. Anhand der Nummerierung, kann sehr schnell erkannt werden, ob ein Paket nicht sauber empfangen werden konnte. Bspw. Erhalten wir die Pakete 4,5,7 → Paket 6 wurde nicht korrekt übertragen.

### 4.1.4 Adaptive Data Rate (ADR)

Ist Adaptive Data Rate aktiv (ADR), entscheidet das LoraWan Endgerät selbständig ob und wie die Kommunikation optimiert werden soll. Gemäss TTN [3] sollte ADR nur für statische Nodes oder für mobile Nodes zur Erkennung von Stops (keine Bewegung) verwendet werden. Gemäss TTN ist ADR also nicht unbedingt für die Atas-Tracker geeignet. Die Tracker sind ja meistens in Bewegung bspw. durch Laufen, Ski fahren usw. Dennoch wurde mein Interesse geweckt und ich wollte genauer wissen warum dieses Feature nicht empfohlen wird.

Sobald ein LoraWan Endgerät dem Netzwerk mitteilt, dass es ADR verwenden möchte, beginnt das TTN Daten aufzuzeichnen. Die letzten 20 Messdaten die der Node gesendet hat werden analysiert. Es folgt ein Beispiel von der TTN Webseite:

# Kapitel 5

# Prototyp

Das Bachelorarbeit wird in 2 Hauptaufgaben aufgeteilt. **Testing** und das erstellen eines zweiten Prototyps nachfolgend genannt **Prototyp 2**. Dieses Kapitel beschreibt, wie ich beim Aufbau des Prototypen 2 vorgegangen bin.

## 5.1 Ablauf Vorgehen

Der im Projekt 2 erstellte Prototyp wurde mit sehr simplen vorgefertigten Elektronikkomponenten umgesetzt. Ziel dieser Arbeit war es in möglichst kurzer Zeit einen Prototypen aufzubauen. Während dieser Arbeit sollen nun neue Komponenten evaluiert und ein zweiter Prototyp erstellt werden. Der Ablauf dieser Phasen sieht wie folgt aus:

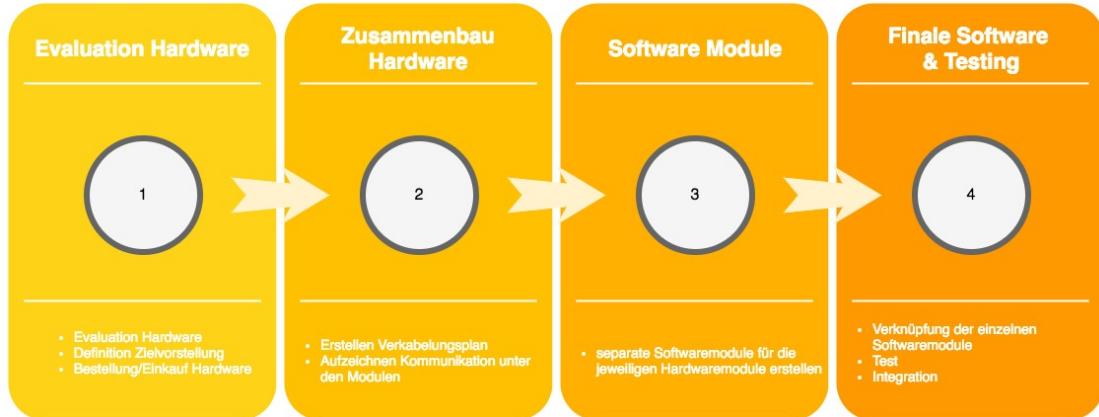


Abbildung 5.1: Ablauf für den Aufbau des zweiten Prototyps

Die einzelnen Schritte werden auf den kommenden Seiten detailliert erklärt.

### **5.1.1 Evaluation Hardware**

Auf Grundlage der bestehenden Prototyps soll neue Hardware evaluiert werden. Der Funktionsumfang des Prototyps soll gleichwertig bleiben. Sind die Komponenten erst definiert, wird das Material bestellt.

### **5.1.2 Zusammenbau Hardware**

Die Komponenten werden zusammengebaut und untereinander verbunden. Zur besseren Übersicht wird ein Schema der Elektronik erstellt.

### **5.1.3 Software Module**

Der bestehende Sourcecode des Prototyps 1 kann leider nicht eins zu eins übernommen werden. Die Software ist sehr stark für die Raspberry Pi Umgebung angepasst und ist daher nicht mit dem ESP32 Umfeld kompatibel. Die Software muss portiert werden. Die Logik der Software kann übernommen werden.

Pro Hardware Modul wird ein separates Software Modul erstellt.

### **5.1.4 Finalisierung**

Die Erstellten Softwaremodule werden miteinander verknüpft und die Software fertiggestellt. Abschließend werden einige Testszenarien durchgeführt.

## 5.2 Begründung neuer Prototyp

Es gibt diverse Gründe für diesen Schritt

1. Auf dem Prototyp 1 wird ein vollwertiges Betriebssystem (Raspian, Linux Derivat) eingesetzt. Einige Funktionen bspw. das Lesen der GPS Daten werden vom System sehr vereinfacht. Dies ermöglichte mir einen raschen und unkomplizierten Aufbau der Software. Obschon ein solches 'High Level' OS eine grosse Hilfe in der Entwicklung darstellt, ist es in der Industrialisierung der Lösung eher hinderlich. Der Einsatz eines OS hat diverse Nachteile
  - Vom OS selbst benötigen wir für den Betrieb der Atas-Node Software nur einen Bruchteil der Funktionalität. Speicher, Memory und Systemperformance wird für Systemprozesse verschwendet.
  - Durch die hohe Auslastung des Systems werden stärkere Prozessoren benötigt. Der Energieverbrauch der Lösung steigt.
  - Mehr Software bedeutet Zwangsläufig mehr Fehlerquellen. Obschon das eingesetzte System (Raspian) als Stabil gilt, kann es aufgrund der Softwarekomplexität zu Fehlern kommen.
  - Durch den Einsatz eines OS reduzieren wir die Sicherheit des ganzen Systems. Je mehr Interfaces ein System anbietet, desto anfälliger wird es für den unerlaubten Zugriff.
2. Der Prototyp 1 bietet viele Anschlussmöglichkeiten. Bspw. für ein Display, USB, oder eine RJ45 Buchse. Funktionen die wiederum welche Entwicklung stark vereinfachen, aber den Energieverbrauch erhöhen und die Grösse des Systems unnötig vergrößert.
3. Die Verbauten Komponenten sind nicht für eine Industrialisierte Lösung geeignet. Man könnte sogar von einem 'Bastelpunkt' sprechen. Die Komponenten hindern das Projekt daran eine professionelle Form anzunehmen.

### 5.3 Ablauf Prototyping

Aus meiner Sicht bildet während der Arbeit zu konstruierende Prototyp 2 nur einen Zwischenstufe bis zu Finalen Version. Eine Visualisierung meiner Vorstellung:



Abbildung 5.2: Prototyping Ablauf

Mein Ziel ist es mit dem Umstieg auf anderen Komponenten, den Grundstein für den Prototypen 3 zu legen. Der Prototyp 3 ist nicht Bestandteil dieser Arbeit. Prototyp 3 bildet die finale Prototyp Version. Die Einzelnen Komponenten vom Prototyp 2 werden auf einen eigens designeten PCB installiert. Durch diesen Schritt würde sich das System in Sachen Platzbedarf nochmals massiv verbessern.

## 5.4 Evaluation Hardware

Für den Aufbau des zweiten Prototypen wurde die nachfolgende Hardware beschafft.

Gerätetyp	Model	Bild
Entwicklungsboard + Mikrocontroller	Espressif ESP-WROOM-32 Core Dev Kit	TODO
GPS Module	ublox Neo 6M	TODO
Lora Modul	RFM96	TODO
Speaker	Piezo Speaker	TODO
Display	eink Display Waveshare 1.54 Zoll 200x 200	TODO
Schalter	gewöhnlicher 4 Pin Schalter	TODO

## 5.5 Zusammenbau Hardware

### 5.5.1 Schema

Die Kommunikation unter den Komponenten wurde im Laufe der Projektarbeit für mich immer unübersichtlicher. Um mir die Arbeit zu erleichtern, habe ich ein Schema erstellt. Vor der Arbeit hatte ich noch nie ein Schema gezeichnet und musste mich zuerst einige Stunden in die Materie einarbeiten. Auf anraten einiger Kollegen, mit Erfahrung im Elektronik Umfeld, sowie der frei verfügbaren Lernressourcen und Vorlagen, fiel meine Wahl auf das Programm **Eagle** der Firma Autodesk [4]. Dank der guten Dokumentation und den Lernvideos auf diversen Plattformen bspw. Youtube kam ich sehr schnell vorwärts.

Ich begann damit Vorlagen der Elektronischen Bausteine/Chips finden. Die Vorlagen zu Lora, GPS, ESP, Schalter und Piezo Lautsprecher konnte ich teilweise vom Hersteller oder von der Community herunterladen und in Eagle importieren. Für das Display musste ich eine bestehende Vorlage eines Displays anpassen. Ich habe dabei die Größe und die Pinbelegung verändert, damit das Element so ähnlich wie nur möglich wie das verwendete Display (Waveshare eInk 200x200) aussieht.

Das Schema ist auf der nachfolgenden Seite abgebildet.

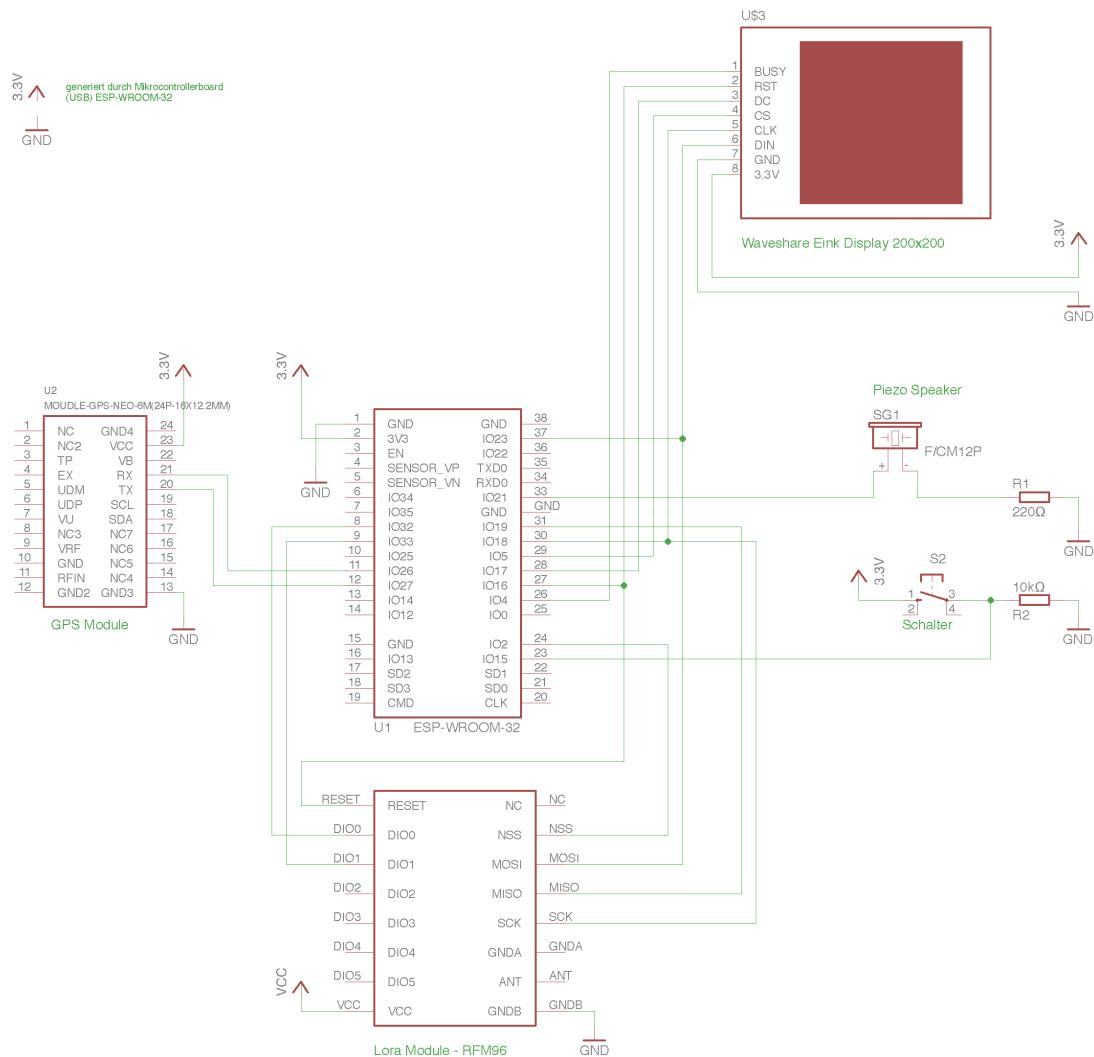


Abbildung 5.3: Prototyp Schema

### 5.5.2 Mikrocontroller - Pinbelegung

Die nachfolgende Pinbelegung wurde verwendet um den ESP32 mit den übrigen Komponenten zu verbinden. Zu meiner Überraschung sind die Pins nicht fix durch den ESP32 festgelegt. Die Pins können meist für jegliche Funktion bspw. PWM, SPI verwendet werden. Durch diese Freiheit wird die Verkabelung massiv vereinfacht und die Anzahl der möglichen Fehlerquellen wird reduziert.

MCU Pin	Modul	Funktion
IO02	Lora	SPI ChipSelect (CS)
IO04	Lora + Display	SPI Busy
IO05	Display	SPI ChipSelect(CS)
IO15	Switch	-
IO16	Lora + Display	SPI Reset
IO17	Display	DataCommand (DC)
IO18	Lora + Display	SPI Clock (CLK)
IO19	Lora + Display	Master In Slave Out (MISO)
IO21	Piezo Speaker	-
IO23	Lora + Display	SPI Master Out Slave In (MOSI)
IO26	GPS	UART RX
IO27	GPS	UART TX
IO32	Lora	DIO0
IO33	Lora	DIO1

### 5.5.3 Zusammenbau

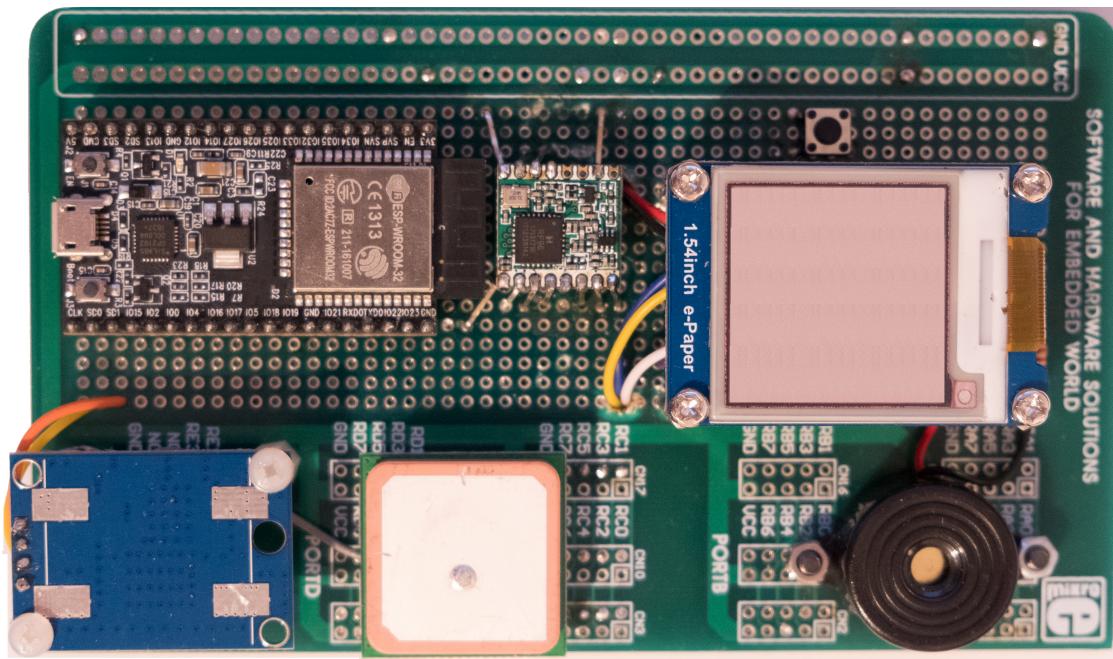


Abbildung 5.4: Prototyp 2

## 5.6 Software Module

### 5.6.1 Entwicklungsumgebung

Durch den Umstieg vom Raspberry Pi auf den Mikrocontroller ESP32 hat sich im Bereich der Entwicklungsumgebung einiges verändert. Die ESP32 Plattform unterstützt, Stand heute (16.11.2017), zwei Entwicklungsumgebungen. Das wäre einerseits ESP-IDF (IoT Development Framework) sowie Arduino[5]. Die beiden Umgebungen sind miteinander kombinierbar. So kann in der IDF Umfeld die Arduino Unterstützung als Komponente importiert werden. Bestehende für Arduino konzipierte Bibliotheken können dadurch ebenfalls verwendet werden. Ich habe mich in Absprache meines Betreuers für die ESP-IDF Umgebung entschieden und mein System entsprechend vorbereitet. Die Installation der Entwicklungsumgebung ist nicht schwer, bedingt aber einige manuelle Schritte. Die Umgebung besteht aus einer Reihe von Tools (Toolchain) zum Kompilieren, Fehlersuche und dem Flashen des Mikrocontrollers.

Ich verzichte an dieser Stelle auf eine detaillierte Auflistung aller Installationsschritte. Die Installation wurde gemäss der Anleitung auf der Webseite der Firma Espressif durchgeführt [6].

Als Hostsystem wurde ein macOS System verwendet. Damit das System den ESP korrekt erkennen konnte, musste ein zusätzlicher Treiber installiert werden [7].

## 5.7 Finalisierung

# Kapitel 6

## Testing

### 6.1 Ablauf Vorgehen

Ein Teil des Systemaufbaus aus dem vorgängigen Projekt soll getestet werden. In diesem Kapitel wird genau beschrieben, wie beim Testing des Systems vorgegangen wird.

In der nächsten Abbildung sind die geplanten Schritte aufgeführt.

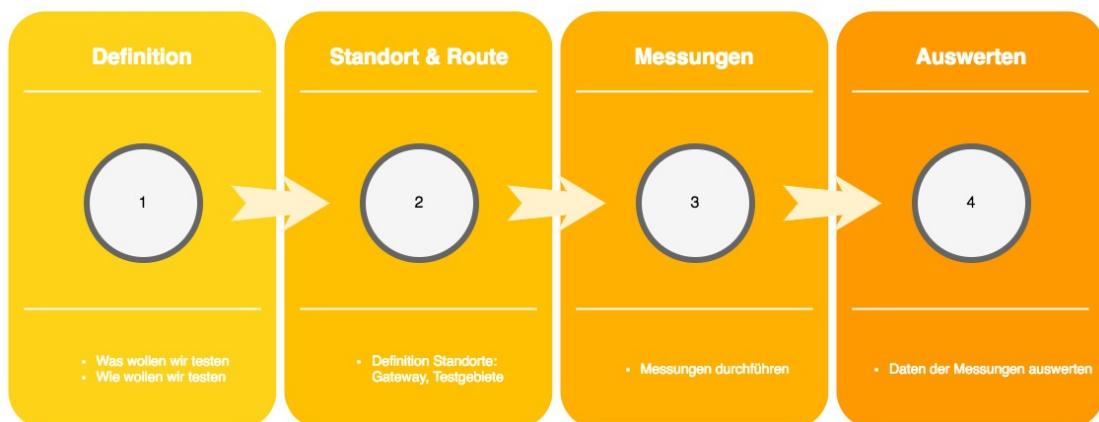


Abbildung 6.1: Ablauf Testing

Die einzelnen Schritte werden auf den kommenden Seiten detailliert erklärt.

### 6.1.1 Definition

Es muss definiert werden

- welche Komponenten geprüft werden sollen

Während der Arbeit soll die Kommunikation zwischen Atas-Node und Atas-Gateway getestet werden.

- welche Aspekte der Komponente sollen betrachtet werden

Der Fokus liegt auf der **Zuverlässigkeit** der Übertragung. Mich interessiert, wie sicher es ist, dass eine Meldung ihr Ziel erreicht.

- mit welchen Mitteln resp. Messinstrumenten gemessen wird

Das TTN Netzwerk bietet uns die Möglichkeit mittels Dashboard einzusehen, wann welche Pakete auf dem Gateway eintreffen. So können wir prüfen ob die Verbindung zwischen Gateway und Tracker funktioniert.

The screenshot shows a table titled 'Messages' with the following data:

payload	time	frame	RSSI	frequency
48 65 6C 6C 6F 2C 20 77 6F 72 6C 21	22:53:43	3	-96	867.10000
48 65 6C 6C 6F 2C 20 77 6F 72 6C 21	22:52:37	2	-94	868.50000
48 65 6C 6C 6F 2C 20 77 6F 72 6C 21	22:51:30	1	-93	868.30000
48 65 6C 6C 6F 2C 20 77 6F 72 6C 21	22:50:24	0	-97	868.10000

Abbildung 6.2: TTN Messages

- welche Daten sollen untersucht werden

Für die Messungen sind mehrere Daten von Interesse. Dazu gehören

- Signalstärke
- Wie viele Pakete gehen verloren
- Distanz der Übertragung
- Sichtverhältnisse zwischen Sender und Empfänger

### Schlussfolgerung

Abgeleitet aus der vorhergehenden Definitionen ergibt sich die folgende Aufgabenstellung im Bereich Testing. Es muss eine Testumgebung aufgebaut und die Verbindung getestet werden. Um von den bestehenden TTN Gateways unabhängig zu sein, wird ein **eigener Gateway** aufgebaut. Anschließend werden mehrere Wanderungen auf verschiedenen Routen durchgeführt. Die Verbindung wird aufgebaut und mittels TTN Dashboard geprüft.

### **6.1.2 Standort**

Ausgehend von den vorangegangen Definitionen muss folgendes definiert werden.

- Es muss ein Ort gefunden werden, wo der Gateway platziert werden kann.

Der Standort kann frei gewählt werden. Einzige Voraussetzung ist ein Zugang zum Internet. Der Gateway empfängt die Daten der Endgeräte und muss diese dem TTN Netzwerk über das Internet weiterleiten können. Eine erhöhte Position ist von Vorteil.

- Für die Tests sollten mind. 3 Routen/Wanderungen definiert werden. Dies garantiert eine vernünftige Anzahl an Messwerten.
- Auf dem Pfad selbst werden mind. 5 Orte definiert wo die Verbindungstests durchgeführt werden.

### **6.1.3 Messungen**

Die vorgängig definierten Routen werden abgelaufen und die Messungen werden an den definierten Standorten durchgeführt.

### **6.1.4 Auswertung**

Die erhobenen Daten werden analysiert und kategorisiert. Messungen sollen in 3 Kategorien eingeteilt werden.

1. Kein Sichtkontakt zwischen Empfänger und Sender.
2. Wenig Sichtkontakt resp. teilweise blockiert. Bspw. durch Bewaldung oder Steinformationen
3. Direkter Sichtkontakt

## **6.2 Definition**

## **6.3 Standort**

## **6.4 Messungen**

## **6.5 Auswertung**

### **6.5.1 Installation Software**

Die Installation der Gateway Software wird von TTN bereitgestellt und von der Community aktiv verwaltet und verbessert. Die TTN Community Gruppe aus Zürich hat ein sehr simples Installationsskript zusammengestellt um die Gateway Software automatisiert zu installieren.

Vorgehen auf dem Gateway

1. Paketdefinitionen geupdated

Code: apt-get update apt-get upgrade

```
sudo dpkg-reconfigure tzdata // Europe/Zurich
```

```
sudo raspi-config // Enable SPI
```

```
git clone https://github.com/ttn-zh/ic880a-gateway.git /ic880a-gateway cd /ic880a-gateway
```

```
change EUI Source to Wireless
```

```
sudo ./install.sh spi
```

# Selbständigkeitserklärung

Ich bestätige, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Textstellen, die nicht von mir stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen.

Ort, Datum: Bern, 15.01.2018

Namen Vornamen: Martin Schmidli

Unterschriften: .....

# Anhang

## 6.6 Software

Es wird kein Code direkt an dieses Dokument angehängt. Zur Verwaltung des Sourcecodes wurde die Plattform Github verwendet. Jegliche Commits vor dem 16.09.2017 gehören zur Projekt 2 Arbeit. Commits nach diesem Datum wurden im Zuge der Bachelor Thesis erstellt. Folgend Sie den Links um den Sourcecode einzusehen.

### 6.6.1 Atas-Webapp

<https://github.com/schmm2/atas-webapp>

### 6.6.2 Atas-Service

<https://github.com/schmm2/atas-service>

### 6.6.3 Atas-Node

Software welche auf dem 1. Prototypen eingesetzt wird

<https://github.com/schmm2/atas-node>

### 6.6.4 Atas-Node2

Software welche auf dem 2. Prototypen eingesetzt wird

<https://github.com/ATAS-Group/atas-node2>

# Abbildungsverzeichnis

2.1	Atas Schema Grob	10
3.1	Systemübersicht Detail	14
3.2	Tracker Detail	15
3.3	Gateway Detail	16
3.4	TTN Detail	17
3.5	TTN Detail	18
4.1	OTAA Ablauf	21
5.1	Flowchart Prototyp 2	23
5.2	Prototyping Ablauf	26
5.3	Prototyp2 Schema	29
5.4	Prototyp 2	31
6.1	Flowchart Testing	34
6.2	TTN Messages	35

# Literaturverzeichnis

- [1] <https://www.thethingsnetwork.org/wiki/LoRaWAN/Security> 27.07.2017
- [2] <https://www.jaguar-network.com/en/news/lorawan-in-a-nutshell-2-internet-of-things-iot>,  
27.07.2017
- [3] <https://www.thethingsnetwork.org/wiki/LoRaWAN/ADR> 27.07.2017
- [4] <https://www.autodesk.com/products/eagle> 27.07.2017
- [5] <http://iot-bits.com/documentation/esp32-tutorial-and-example-programs/>  
20.12.2017
- [6] <https://dl.espressif.com/doc/esp-idf/latest/get-started/index.html>  
27.12.2017
- [7] <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers> 27.12.2017