LoRaWAN-Konzept der „5 FÜR SÜDWESTFALEN“

Im Auftrag von

Gefördert durch:

Bundesministerium
für Wohnen, Stadtentwicklung
und Bauwesen

aufgrund eines Beschlusses
des Deutschen Bundestages

Versionshistorie

Version	Datum	Beschreibung	Freigabe
1.0	04.04.2024	Erstfassung	Stadt Bad Berleburg

1 Inhaltsverzeichnis

2 Einleitung

Im Rahmen des Bundesmodellprojekts „Smart Cities: 5 für Südwestfalen“ gehen die fünf Pionierkommunen Arnsberg, Bad Berleburg, Menden, Olpe und Soest voran, um eine ganze Region smarter zu machen.

Kernelement des Projektes ist der Aufbau einer gemeinsamen Urbanen Datenplattform (UDSP). Die Plattform wird für zahlreiche Use Cases genutzt, dazu gehören neben Open Data vor allem Anwendungen im Bereich Internet of Things (IoT). Als wichtigste Basistechnologie für die vielen verschiedenen, IoT-basierten Anwendungsfälle hat sich LoRaWAN herauskristallisiert.

Dieses Dokument behandelt die technischen Grundlagen zu LoRaWAN, den Aufbau und Betrieb eines LoRaWAN-Netzes mit dem Schwerpunkt kommunaler Anforderungen und die Integration mit der UDSP. Neben technischen Aspekten werden Inhalte zur Beschaffung und zum Projektmanagement vermittelt.

Aufgrund der gemeinsamen Nutzung der Plattform durch verschiedene, voneinander unabhängige Organisationen spielt Kooperation eine große Rolle. Daher definiert dieses Dokument auch Standards und Best Practise Beispiele für Produkte und Prozesse, um einen reibungslosen Betrieb der Plattform unter Koexistenz vieler unterschiedlicher und unabhängiger Anwendungsfälle zu gewährleisten.

Die Inhalte des Dokuments sind im Wesentlichen durch die Expertise der Firma Hypertegrity AG, die gleichzeitig Architekt der UDSP ist, und deren Netzwerk an weiteren externen Experten entstanden. Ebenso sind Erfahrungen aus den Pionierkommunen und aus Gesprächen mit anderen MPSC-Kommunen eingeflossen.

 Diese und jede neuere Version wird auf der Homepage der Stadt Bad Berleburg (https://www.bad-berleburg.de/Leben/Nachhaltig-und-Smart/Smart-City-BLB/) veröffentlicht. Die Weiterentwicklung erfolgt durch das Smart City Team der Stadt Bad Berleburg. Anregungen und Verbesserungsvorschläge richten Sie bitte schriftlich an S.Willerscheid@Bad-Berleburg.de oder M.Ladda@Bad-Berleburg.de .

3 Allgemeine Informationen zum LoRa-Konzept

Smart City ist nicht ohne IoT-Integration denkbar. Der urbane Datenraum umfasst neben IoT auch Bereiche wie Mobilität, Health, GIS usw., jedoch ist IoT ein essenzieller Bestandteil, um den oder die digitalen Zwilling(e) einer Stadt aufzubauen. Dieses Dokument beschreibt die technischen und organisatorischen Grundlagen für die Planung, den Aufbau und den Betrieb einer IoT-Infrastruktur und die Integration mit einer urbanen Datenplattform. Aufgrund der großen Verbreitung, der Standardisierung und der vergleichsweise geringen Kosten liegt der IoT-Schwerpunkt auf der LoRaWAN-Technologie. Bei der urbanen Datenplattform wird davon ausgegangen, dass sie das Kontextmanagement mittels der FIWARE-Standards implementiert.

Es gibt eine Vielzahl von möglichen Übertragungstechnologien:

- Mobilfunkbasiert → LTE-M / LTE / 5G / Narrow-Band-IoT (NB-IoT)
- LoRa → LoRaWAN / Helium (Kombination LoRa und Kryptowährung Helium)
- WiFi / WLAN
- Gebäudeautomation → BACnet / Zigbee / Z-Wave / KNX / DALI / Matter
- Smart Meter → M-Bus / wM-Bus
- Industrie → Modbus / OPC-UA
- Nahbereich → Bluetooth / NFC

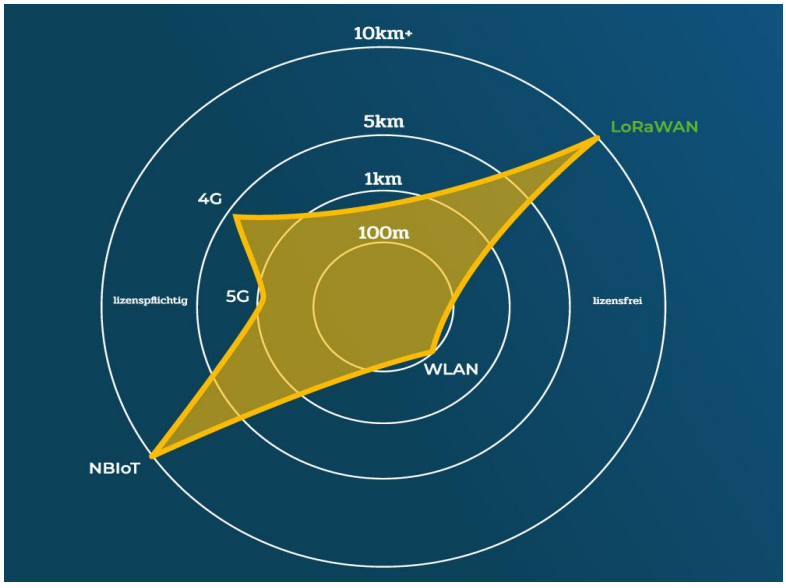


Abbildung 1 - Vergleich der Funktechnologien[1]

[1] *Quelle: https://alpha-omega-technology.de/smartes-facilitymanagement-mit-lorawan-energieeffizient-und-lizenzfrei*

LoRaWAN bietet eine einzigartige Kombination aus kostenfrei nutzbaren Funkfrequenzen, hoher Reichweite (bei geringer Datenmenge) und einem geringen Energieverbrauch. Diese Kombination ist für viele Anwendungsfälle der Smart City sehr gut geeignet.

Anwendungsfälle wie Parkraummanagement, Umweltüberwachung, Raumklimaüberwachung (CO2, Temperatur, Luftfeuchte), einfache Personenstromzähler oder sicherheitsunkritische Fluss-Pegelüberwachung benötigen nur geringe Datenmengen bei Übertragungsperioden im Minutenbereich. Somit kann ein hinreichend akkurates Bild bei der Datenerhebung erstellt werden.

3.1 IoT, LoRaWAN & Urban Data Space Platform

Dieses Konzept zeigt auf, wie Datensouveränität und Interoperabilität in der Praxis umgesetzt werden können. Hierfür wird der Datenraum Urban-IoT via LoRaWAN erschlossen.

Das Konzept beginnt mit einer Gesamtübersicht über die relevanten Komponenten einer LoRaWAN- und Smart City Integrations-Architektur. Anschließend wird mit der technischen Einführung das Fundament gelegt. Es folgen Empfehlungen für den Aufbau eines Netzes auf Basis praktischer Erfahrungen und es werden die Prozesse und Datenflüsse im Detail dargestellt. Dabei behandeln wir u.a. den Umgang mit den Smart Data Models. Danach zeigen wir auf, wie ein Citizen-Science-Ansatz zur Nutzung der LoRa-Infrastruktur durch die Bürger der Kommunen funktionieren kann. Anhand eines LoRa-Adapters in Node-RED wird aufgezeigt, wie die praktische Anwendung zur Integration mit der Plattform funktioniert.

3.2 Datenplattform Standards

(Meta-)Daten aus Fachverfahren, IoT-Daten sowie Daten aus anderen Datenräumen - wie z.B. dem Datenraum Mobilität - bieten die Möglichkeiten, das Leben und den Lebensraum lebenswerter zu gestalten, den Bürgern mehr Transparenz zu schaffen und durch den bedarfsgerechten Einsatz von Ressourcen Geld zu sparen und die Umwelt zu schonen. Als Grundlage müssen die Daten in einem standardisierten Datenformat zugänglich gemacht werden. Die Anforderungen an das sogenannte Kontext-Informations-Management wurden vom Netzwerk der Open & Agile Smart Cities (OASC) (https://oascities.org/list-of-cities/) in den Minimum Interoperability Mechanismen (MIMs) im MIM1 definiert:

> "Die Verwaltung von Kontextinformationen gewährleistet einen umfassenden und integrierten Zugang, die Nutzung, die gemeinsame Nutzung und die Verwaltung von Daten über verschiedene Lösungen und Zwecke hinweg. Es verwaltet die Kontextinformationen, die von Internet-of-Things-Geräten (IoT) und anderen öffentlichen und privaten Datenquellen stammen, und bietet übergreifende Kontextdaten und den Zugriff über eine einheitliche Schnittstelle."

Für MIM1 hat die FIWARE-Foundation das NGSI-Protokoll durch die ETSI standardisieren lassen. Außerdem hat die FIWARE die Referenzimplementierung des Kontext-Brokers (Orion) als Open-Source-Lösung zur Verfügung gestellt. Somit ist die Möglichkeit gegeben, Kontext-Informationen über eine standardisierte Schnittstelle bereitzustellen. Hier setzt dann MIM2 ein, um die Interoperabilität zu gewährleisten. MIM2 fordert:

“Smart Data Models für interoperable und replizierbare intelligente Lösungen in mehreren Sektoren, beginnend mit Smart Cities, aber auch für Smart Agrifood, Smart Utilities, Smart Industry usw.

Harmonisierte Darstellungsformate und Semantiken, die von Anwendungen sowohl für die Nutzung als auch für die Veröffentlichung von Daten verwendet werden.

Smarte Datenmodelle für interoperable und reproduzierbare smarte Lösungen in verschiedenen Sektoren, beginnend mit smarten Städten, aber auch für smarte Agrar- und Ernährungswirtschaft, smarte Versorgungsunternehmen, smarte Industrie, usw.”

Hier hat die FIWARE-Foundation mit mehr als 800 lizenzfrei nutzbaren Datenmodellen der smartdatamodels.org einen wichtigen Grundstein zur weltweiten Interoperabilität gelegt.

Eine FIWARE-Plattform wie die Urban Data Space Platform (UDSP) nach der Referenzarchitektur der DIN SPEC 91357 kann somit auf den Standards aufbauen und einen interoperablen Urbanen Datenraum (MIM1 & MIM2) schaffen.
Wichtig: Ein Ziel der Plattform ist es, die Daten-Basis für den/die Digitalen Zwilling/e zu schaffen. Der Digitale Zwilling ist eine detaillierte Darstellung eines physischen Objekts oder Systems – z.B. eines Parkplatzes –mit dem Zweck, dessen Zustand zu erfassen. Aus diesem Grund konzentriert sich das Datenmodell auf die Darstellung des Parkplatzes anstelle des "Parksensors" selbst. Die Informationen über den Zustand des Parkplatzes stammen vom Sensor, aber die spezifischen Merkmale des Sensors (wie Art und Hersteller) werden in diesem Prozess abstrahiert. Dies führt dazu, dass wir eine standardisierte, sensor-unabhängige Darstellung des Parkplatzes erhalten, was die Grundlage für unseren Digitalen Zwilling bildet. Die Reduzierung der Abhängigkeit von Sensortyp und Sensorhersteller trägt dazu bei, dass das Datenmodell und somit der Digitale Zwilling flexibel und anpassungsfähig bleiben.

Kurz: Die Urban Data Space Platform schafft als City Integration Layer (oder auch zentrale Datendrehscheibe) Datenverfügbarkeit, Einheitlichkeit und Interoperabilität - unabhängig von der verwendeten Basis-Technologie. Dabei befinden sich die Daten immer im Hoheitsgebiet der Städte, Kommunen oder Dörfer. Die Datensouveränität - als wichtige Säule der digitalen Souveränität - ist daher sichergestellt.

3.3 Komponentenübersicht

Das folgende Bild zeigt den Gesamtüberblick der beteiligten Komponenten:

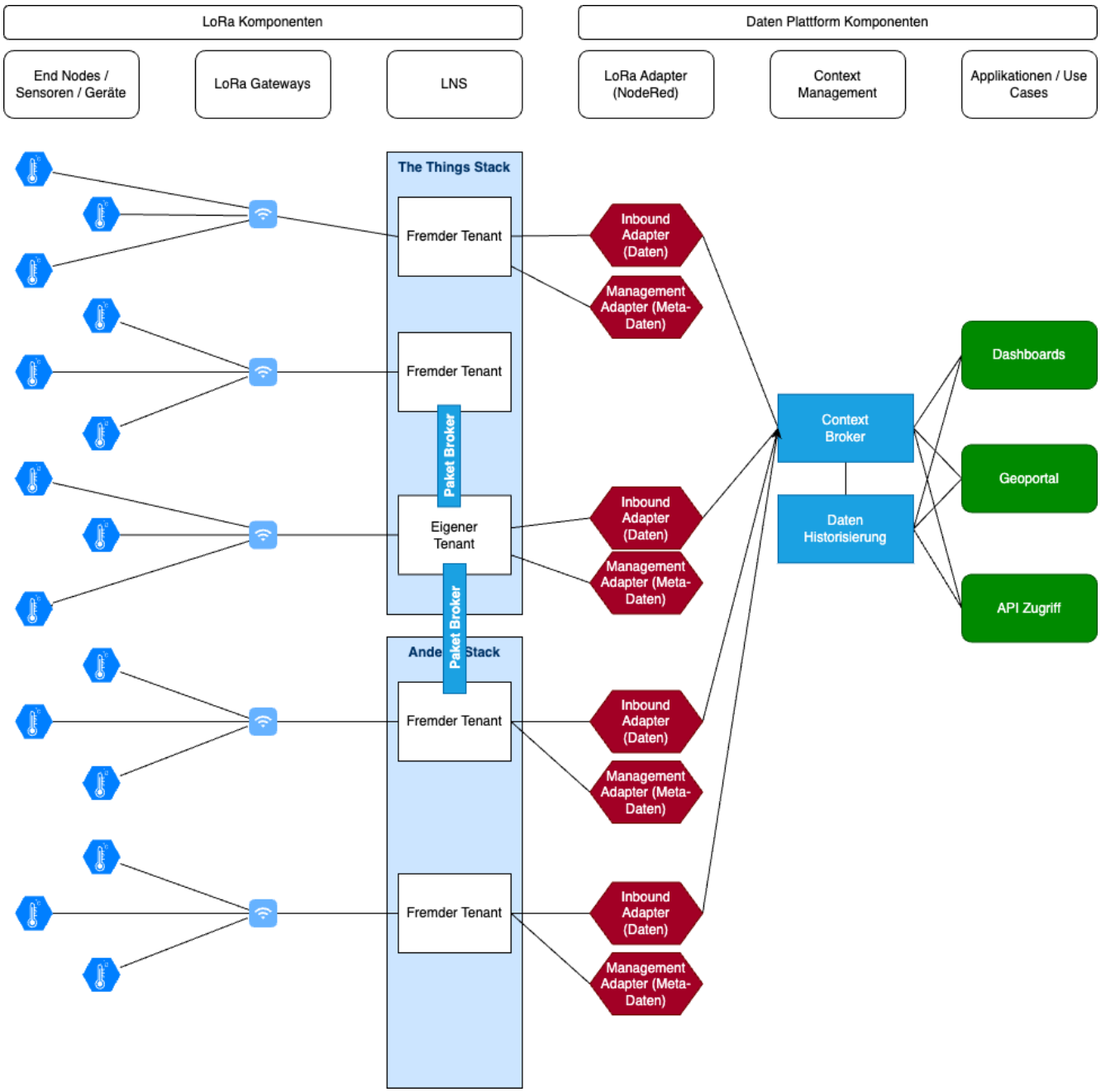


Abbildung 2 - Gesamtüberblick Komponenten[2]

[2] Eigene Darstellung

3.4 Begriffserklärung der Komponenten

Im Folgenden werden die LoRa-Komponenten und die Komponenten der Plattform aus der Abbildung kurz erläutert. Details folgen in weiteren Kapiteln.

3.4.1 Sensor

Ein Sensor erfasst Messwerte und überträgt diese verschlüsselt via LoRa an alle erreichbaren Gateways.

3.4.2 Gateway

Das Gateway empfängt die Daten vom Sensor und leitet diese an den LoRaWAN Network Server (LNS) weiter.

3.4.3 LoRaWAN Network Server (LNS)

Der LNS implementiert das LoRaWAN-Protokoll. Er entschlüsselt die Daten des Sensors (und verschlüsselt (im Falle einer bidirektionalen Kommunikation mit dem Sensor) auch die Daten. Der LNS bietet in der Regel auch einen MQTT-Server. Am MQTT-Server kann sich ein MQTT-Client (wie er z.B. im Node-RED zur Verfügung steht) registrieren und somit Messwerte der Sensoren aus dem LoRaWAN bekommen. Weitere Bestandteile des LNS sind Paket Broker und Payload Decoder.

3.4.4 Packet Broker

Ein Packet Broker dient dazu, LoRaWAN-Daten mit anderen Mandanten (Tenants) im TTI auszutauschen. Ein Packet Broker bietet sogar die Möglichkeit Daten mit anderen Stacks (wie z.B. Chirpstack) auszutauschen. Ein Packet Broker ist auch für Chirpstack verfügbar – allerdings muss diese Chirpstack-Installation diesen dann implementieren und weitere Bedingungen erfüllen (z.B. eine eigene NetID).

3.4.5 Payload Decoder

Der Payload-Decoder decodiert die von den Sensoren gesendeten Daten. Die Daten werden in der Regel in einem binären Format (sensorherstellerabhängig) übertragen und müssen von einem Payload-Decoder interpretiert werden, um nutzbare Informationen daraus zu gewinnen. Als Ergebnis der Dekodierung ist das JSON-Format üblich. Der Payload-Decoder wird in der Regel vom Sensorhersteller bereitgestellt. Sollte dies nicht der Fall sein, muss der Hersteller mindestens eine Payload-Spezifikation bereitstellen.

Der sensortyp-spezifische Payload-Decoder arbeitet in der Regel im LNS. Der Payload-Decoder ist entweder im LNS als Vorauswahl verfügbar (der Sensor ist dem LNS bekannt) – oder muss installiert werden. Sollte der Sensor-Hersteller keinen Decoder für das LNS zur Verfügung stellen, muss ein eigener Decoder entwickelt werden. Auch im LoRa Adapter kann ein Payload-Decoder integriert werden. Hier sind die Lizenzbedingungen des Payload-Decoders des Sensorherstellers zu beachten – hier ist generell Wert auf eine Open Source-Lizenz des Decoders zu legen.

3.4.6 LoRa Adapter (Node-RED)

Der LoRa Adapter wandelt die proprietären nutzbaren Informationen, die das Resultat des Payload Decoders sind, in FIWARE-Konforme Datenmodelle um und sendet diese an den Context Broker. Der Begriff "Inbound" bedeutet in dem Zusammenhang, dass Daten durch den Adapter "hineinkommen" in die Plattform.

3.4.7 Management Adapter (Metadaten)

Über den (optionalen) Management Adapter können Informationen zu den Sensoren aus dem LNS abgerufen werden (z.B. Name, Beschreibung oder auch Attribute am Beispiel von TTS).

3.4.8 Context Broker

Der Context-Broker speichert die Context-Informationen und gibt sie an Consumer (z.B. Dashboards, Geoportale, APIs – die in Applikationen / Use Cases genutzt werden) weiter. Der Context Broker bietet auch einen Subscription-Mechanismus, über den sich externe Systeme über Änderungen informieren lassen können. Der Context Broker hält den aktuellen Stand (z.B. „Parkplatz frei“ oder „Temperatur = 16 Grad Celsius“). Die Daten können auch historisiert werden. Neben dem Orion Context Broker der FIWARE-Foundation gibt es weitere FIWARE-kompatible Broker wie Stellio oder Scorpio.

3.4.9 Historisierung

Man unterscheidet bei den Messwerten zwischen aktuellen Daten und historischen Daten. Der Context-Broker behandelt die aktuellen und alle historischen Daten unterschiedlich, indem er sie getrennt speichert. Die Behandlung der historischen Daten wird als „Historisierung“ oder als „temporale Datenhaltung“ bezeichnet. Messwerte werden in Zeitreihendatenbanken (Time Series Databases) gehalten. Hieraus lassen sich Dashboards generieren, die den Verlauf über die Zeit darstellen.

4 Technische Einführung

Inhalt dieses Kapitels sind Detailinformationen zu Auswahl, Aufbau und Betrieb aller Komponenten, welche für ein funktionierendes LoRaWAN-Netzwerk erforderlich sind. Dabei soll dieses Netzwerk die Basis für die Umsetzung von Smart City Use-Cases bilden.

LoRa ist ein Kommunikationsprotokoll. Es ermöglicht die Übertragung kleiner Datenpakete über große Distanzen mittels Geräten, welche sich in der Regel durch ihren niedrigen Energiebedarf auszeichnen.

Neben den technischen Ansätzen zur Erfüllung der Anforderungen soll dieses Konzept auch die technischen Grundlagen vermitteln.

Es gibt drei wichtige Anforderungen an Übertragungstechnologien:

- Nutzung von LowPower-Technologien

- hohe Reichweite
- große Bandbreite zur Übertragung möglichst großer Datenmengen

Allerdings sind nicht alle Anforderungen gleichzeitig erreichbar. Die LoRa-Technologie ermöglicht neben dem Einsatz von LowPower-Nodes eine relativ große Reichweite und erzielt dabei durch die spezielle Chirp-Modulation eine gute Gebäudedurchdringung, dies geht jedoch auf Kosten der verfügbaren Bandbreite. Der Begriff „Node“ ist erforderlich, da sich der Begriff Sensor im eigentlichen Sinne ausschließlich auf die Bestandteile beschränkt, welche die tatsächliche Messung realisieren, nicht jedoch die Verarbeitung und Kommunikation dieser Messwerte. Im alltäglichen Sprachgebrauch werden Node und Sensor meist für die Gesamteinheit aus Sensor(en) und Verarbeitungs-/Sendeeinrichtung verwendet, beispielsweise bietet ein CO2-„Sensor“ in einem Gehäuse die Funktionen zum Messen von CO2-Gehalt und Lufttemperatur sowie die LoRa-Sendeeinrichtung.

Zur besseren Verdeutlichung der Zusammenhänge werden in diesem Konzept Grafiken von The Things Industries genutzt.

4.1 LoRa und LoRaWAN

Im bildlichen Sinne ist LoRa die Definition unseres Alphabetes, unseres Zeichenvorrates. LoRaWAN dagegen ist die Struktur des Schreibblattes, mit Linien oder Kästchen sowie die Sprache, in welcher die Buchstaben zur Bildung von Worten und Sätzen unter Beachtung der richtigen Rechtschreibung und Grammatik genutzt werden. Erst durch diese Definition ergibt sich ein verständlicher Text, den alle Personen, die dieser Sprache mächtig sind, lesen und interpretieren können.

Die LoRa-Modulation ist der physikalische Layer für das LoRaWAN-Protokoll. Es sind jedoch auf Basis dieser Modulation weitere, proprietäre Protokollumsetzungen möglich. Ein Beispiel dafür ist das Protokoll Mioty, welches auf der Basis von LoRa arbeitet, jedoch nicht das LoRaWAN-Protokoll unterstützt. Es stehen 8 Kanäle mit einer Übertragungsrate zwischen 292 Bit/s - 50 kBit/s zur Verfügung. Damit ordnet sich diese Technologie knapp unterhalb der von früher bekannten analogen Datenübertragung über das analoge Telefonnetz ein. In diesem Zusammenhang wird auch oft von einem 0G-Netz gesprochen, das andere Ende der Entwicklung sehen wir aktuell bei 5G bzw. den ersten Ansätzen für 6G.

Folgende Frequenzbänder werden für LoRa genutzt, Gateways sind jeweils werksseitig auf eines dieser Bänder ausgelegt:

Region	Frequenz
Europa	433 MHZ 868 MHz
Asien	430 MHZ
Amerika	915 MHz
weltweit	2,4 GHz

An dieser Übersicht ist zu erkennen, dass mit dem neuen Frequenzband 2,4 GHz erstmalig eine global einheitliche Funkfrequenz zur Verfügung steht. Für dieses Band werden aktuell erste Gateways bei den Herstellern verfügbar. Für kommunale Anforderungen in Deutschland ist

allerdings nicht davon auszugehen, dass ein globales Funkband erforderlich ist. Daher ist die Empfehlung, sich durch die Nutzung des in Europa etablierten 868MHz SRD-Bandes den Zugriff auf ein gut sortiertes Geräteportfolio zu sichern. Für das neue weltweite Band werden perspektivisch immer mehr Geräte verfügbar werden, Kosten und Umfang sind aktuell noch nicht klar definierbar.

4.1.1 Regulatorische Vorgaben

Die verfügbaren Funkbänder werden auf europäischer Ebene reguliert, in Deutschland obliegt der Bundesnetzagentur (BNetzA) die Überwachung der Einhaltung dieser Regeln.

Durch die regulatorischen Vorgaben wird in Europa hauptsächlich das 868 MHz-Band für LoRa genutzt. Dieser Frequenzbereich wird aber auch durch zahlreiche andere IOT-Anwendungen genutzt, dies bedeutet, es kann zu Störungen durch andere, fremde Anwendungen kommen. Dementsprechend sind passende Maßnahmen zu planen und umzusetzen. (vgl. 4.5.1)

Hier sind folgende Regeln zu beachten:

Neben einer maximalen Sendeleistung für die Nodes von 25 mW ist ein maximaler Arbeitszyklus (Duty-Cycle) je Node auf der Luftschnittstelle vorgegeben. Dieser beträgt 1%, damit darf ein Node maximal 36 Sekunden innerhalb einer Stunde senden.

Durch die Auswahl des sogenannten Spreizfaktors kann die Bitrate und Reichweite direkt beeinflusst werden. Dabei bedeuten lange Reichweiten kleinere Bitraten, was nichts Anderes bedeutet, als dass man sich eine große Reichweite auf Kosten der Datenmenge erkauft, welche übertragen werden kann.

Spreizfaktor	Bitrate
LoRa: SF12 / 125 kHz	250 bit/s
LoRa: SF11 / 125 kHz	440 bit/s
LoRa: SF10 / 125 kHz	980 bit/s
LoRa: SF 9 / 125 kHz	1760 bit/s
LoRa: SF 8 / 125 kHz	3125 bit/s
LoRa: SF 7 / 125 kHz	5470 bit/s

In LoRa ist ADR, Adaptive Data Rate, implementiert. Dieser Mechanismus ermittelt die Signalstärke der Nodes und erlaubt damit eine automatische Anpassung auf den kleinstmöglichen Spreizfaktor und damit eine Senkung des Energieverbrauches der Geräte und eine Erhöhung der Netzkapazität. Damit kann der genutzte Spreizfaktor durch die Positionierung der Sensoren und Gateways beeinflusst werden.

So ergibt sich die Empfehlung, die Gateways so zu platzieren, dass sich die Menge der Sensoren innerhalb eines Bereiches befindet, welcher eine Übertragung der Daten in den unteren Spreizfaktoren bis SF9 erlaubt. Gibt es dann besondere Störeinflüsse, welche die Übermittlung behindern, ist eine ausreichende Reserve bis SF12 vorhanden. Außerdem arbeiten die Sensoren umso energiesparender, je kleiner der verwendete Spreizfaktor ist.

Der Spreizfaktor in LoRa ist ein Maß für die Prozessgewinn-Spreizung, die in der LoRa-Modulationstechnologie verwendet wird. Er gibt an, wie viele Chirp-Signale pro Bit übertragen werden. Ein höherer Spreizfaktor bedeutet, dass mehr Chirps pro Bit verwendet werden, was zu einer größeren Verarbeitungsgewinnung führt, aber auch zu einer niedrigeren Datenrate.

In LoRaWAN reicht der Spreizfaktor von SF7 bis SF12. SF7 hat die höchste Datenrate, aber die geringste Reichweite, während SF12 die niedrigste Datenrate, aber die größte Reichweite hat.

Man kann den Spreizfaktor in den Metadaten der empfangenen LoRaWAN-Nachrichten sehen, die von den meisten LNS bereitgestellt werden.

4.1.2 Lizenzrechtliche Vorgaben

LoRa ist eine Entwicklung der Firma Semtech, welche auch die Rechte hält. Lizenzgebühren werden in der Regel bereits bei Herstellern der entsprechenden Funkchips erhoben, welche diese Kosten dann auf den Chippreis umlegen (durchschnittliche LoRa-Chip-Preise liegen im Bereich weniger US-Dollar). Jeder Node beinhaltet mindestens einen dieser Chips, Gateways können mehrere beinhalten. Weitere Kosten entstehen durch die Nutzung der Funktechnologie nicht.

Aufbauend auf der LoRa-Modulation definiert LoRaWAN als ein Funknetzwerkprotokoll die Details, wie alle Komponenten des Netzwerks miteinander agieren. Im Ergebnis entsteht ein Funknetzwerk. LoRaWAN wird, wie auch LoRa durch die LoRa-Alliance spezifiziert und weiterentwickelt. Durch die Nutzung der Funkschnittstelle ist auch dieses Netzwerk ein Best-Effort-Netzwerk, typischerweise wird eine Übertragungsqualität von ca. 96% erreicht. Best Effort bedeutet, dass alle Beteiligten an der Kommunikation ihr Bestes tun, um die Daten von A nach B zu übertragen, es jedoch keine Garantie dafür geben kann. Als Zugriffsverfahren wird Aloha verwendet, ein Verfahren ohne Kanalabtastung.

Die Datenpakete können dabei mit oder ohne Confirmation vom Node zum LNS gesendet werden. Dabei ist jedoch zu beachten, dass die Bestätigung aller eingehenden Nachrichten durch den LNS jeweils einen Downlink (vom LNS an den Sensor) darstellt, welcher in verschiedenen Netzwerken unterschiedlich reglementiert ist.

Die Daten werden innerhalb des Netzwerkes grundverschlüsselt. Eine bessere Übersicht dazu findet sich im Abschnitt 4.3 untenSicherheit in LoRaWAN-Netzen.

4.1.3 Sensoreinbindung

Für die Einbindung von Nodes in das Netzwerk bestehen zwei unterschiedliche Methoden, OTAA und ABP. Aktuell wird OTAA - Over The Air Activation - als Stand der Technik eingesetzt. Dabei wird auf der Basis initialer Schlüssel, welche sowohl im Node (hier Sensor und Gateway) als auch im LNS hinterlegt sind, neues Schlüsselmaterial generiert, auf dessen Basis dann die Kommunikation abgesichert wird. Das neue Schlüsselmaterial ist dann lediglich dem Node und dem LNS bekannt.

Eine ältere Art und Weise ist ABP - Activation by Personalisation. Hier wird die Kommunikation auf Basis von synchronen Schlüsseln abgesichert. Diese Methode gilt als weniger sicher als OTAA, der Einsatz ist jedoch für definierte Anwendungsfälle wie z.B. Mappingdienste (Visualisierung mit geografischem Kontext) sinnvoll. Es existieren im Markt Sensoren, welche zwischen beiden

Varianten umkonfiguriert werden können (z.B. mittels Software des Herstellers und NFC) bzw. solche Geräte, welche lediglich fest eine der beiden Varianten unterstützen.

4.1.4 Vor- und Nachteile

Ziel von LoRaWAN ist es, eine Grundlage für offene Ökosysteme ohne Vendor-Lock-In auf Basis von Open Source zu bieten. Wo liegen die Vor- und Nachteile dieser Technik?

Vorteile:

- Geringer Energiebedarf
- Hohe Reichweite und Robustheit
- Asymmetrische Bandbreite: Die Datenübertragungsbandbreite vom Sensor aus ist höher als im Downlink
- Verlust von einzelnen Daten ist akzeptabel

Nachteile:

- geringe Datenrate
- Best Effort – d.h. Garantien einer 100% gesicherten Datenübertragung sind nicht möglich
- Einschränkungen bei Downlinks
    - Downlinks verbrauchen mehr Energie als Uplinks
    - Downlink-Bandbreite ist geringer als die Uplink-Bandbreite
    - Bestätigte Nachrichten: Die Notwendigkeit von Bestätigungen kann den Verkehr und den Energieverbrauch erhöhen
    - Kommunikationsfenster: Nachrichten können nur in bestimmten Zeitschlitzen empfangen werden

Auf Grund dieser Eigenschaften eignet sich LoRaWAN besonders gut für IOT-Anwendungen, die Kommunikation in Sensornetzen, die Fernparametrierung von Geräten oder die Nutzung als Backupkanal, sollte der Hauptkommunikationskanal temporär nicht zur Verfügung stehen.

Weniger geeignet ist diese Technik für Anwendungen, welche auf die Verfügbarkeit von OTA-Firmwareupdates oder kontinuierliche Datenübertragung (Video/ Audio) angewiesen sind. Man kann z.B. via „Confirmed Data“ eine Wiederholung der Datenübertragung erwirken, aber auch das ist keine Garantie für eine 100% gesicherte Übertragung, daher sind kritische Anwendungsfälle (Hochwasserwarnsysteme etc.) eher nicht (ausschließlich) mit LoRaWAN umsetzbar.

Auch der Einsatz innerhalb der kritischen Infrastruktur stellt besondere Anforderungen an die Netztopologie und die nachgelagerte IT-Infrastruktur, ist jedoch nicht ausgeschlossen. Dabei wird der Umfang der kritischen Infrastruktur durch den Gesetzgeber wie folgt definiert. “Kritische Infrastrukturen (KRITIS) sind Organisationen oder Einrichtungen mit wichtiger Bedeutung für das staatliche Gemeinwesen, bei deren Ausfall oder Beeinträchtigung nachhaltig wirkende

Versorgungsengpässe, erhebliche Störungen der öffentlichen Sicherheit oder andere dramatische Folgen eintreten würden."[3]

4.2 Aufbau eines LoRaWAN-Netzwerkes

Typischerweise setzt sich ein LoRaWAN-Netzwerk aus diversen Elementen zusammen. Es beinhaltet Sensoren im Feld, die durch die Lora Luftschnittstelle mit Datenkonzentratoren, auch als Gateways bekannt, kommunizieren. Diese Gateways wiederum sind über die Wide Area Network-Schnittstelle (WAN; Internet) mit einem LoRa Netzwerk Server (LNS) verbunden.

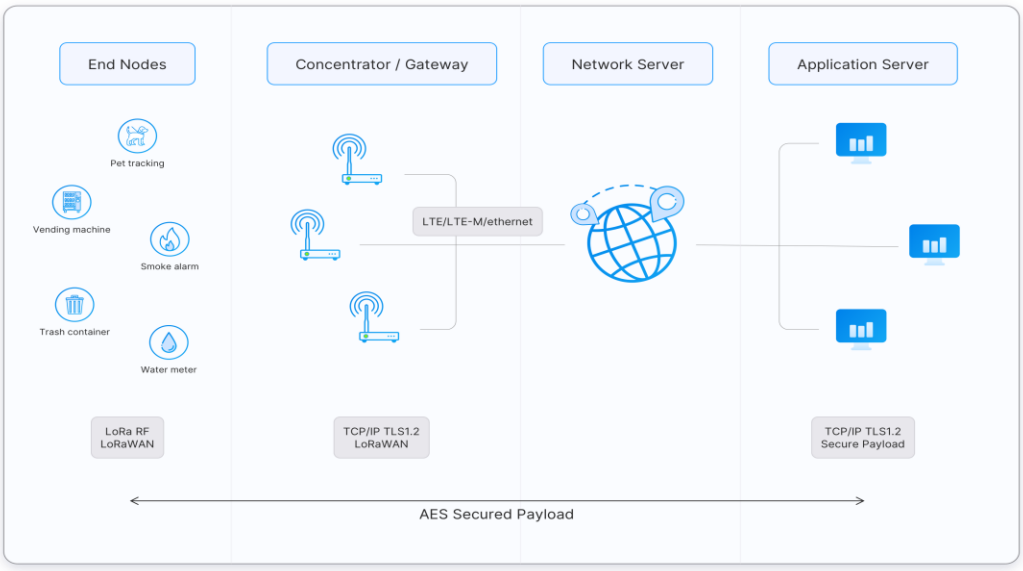


Abbildung 3 - Typische Ausprägung einer LoRa-Netzwerk-Architektur[4]

Der LNS ist Teil eines Softwarestacks, bestehend aus verschiedenen Komponenten. Hier wird neben der Verwaltung der Gateways (Gateway Server) auch das Schlüsselmaterial der Nodes verwaltet (Identity Server) und die Verbindung der einzelnen Nodes mit dem Netzwerk gemanaged (Join Server). Den Abschluss bildet der Application Server, welcher die Daten an definierten Schnittstellen für die nachgelagerten Backendsysteme zur Verfügung stellt.

[3] Quelle: https://www.bbk.bund.de/DE/Themen/Kritische-Infrastrukturen/kritische-infrastrukturen_node.html

[4] Quelle: *https://www.thethingsnetwork.org/docs/lorawan/architecture/architecture.png*

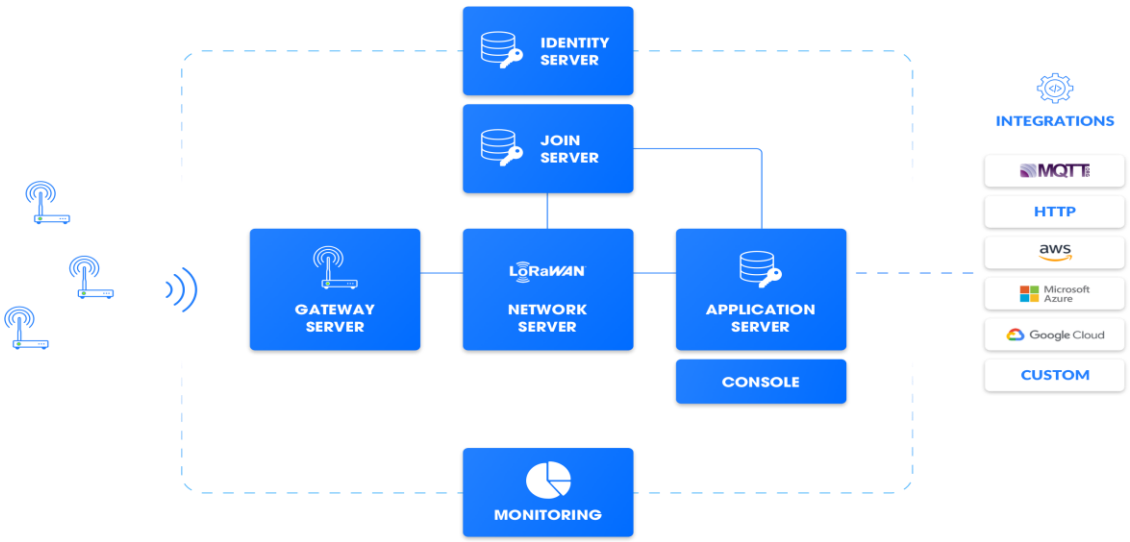


Abbildung 4 - Aufbau TTS[5]

4.2.1 Node

Als Node bezeichnet man das Gerät, welches in Verbindung mit einem Sensor Messwerte aufnehmen kann und technisch in der Lage ist, diese Daten unter Nutzung des entsprechenden Kommunikationsprotokolls an das Netzwerk zu übertragen. Dabei ist der Sensor für die Aufnahme der Messwerte und der Node an sich für die Bereitstellung der Kommunikationswerte verantwortlich. Meist wird die gesamte Einheit aus Sensor- und Kommunikationseinheit als Node bezeichnet, da sie sich in der Regel in einem Gehäuse befindet. Eine weitere Art von Nodes besitzt als Aktor die Möglichkeit, Steuerbefehle zu empfangen und auszuführen.

Es gibt verschiedene Geräteklassen, die sich hinsichtlich ihrer Kommunikationsmöglichkeiten und ihres Energieverbrauchs unterscheiden. Die drei Hauptklassen sind Klasse A, B und C.

- Klasse A (bidirektionale Endgeräte):
    - Kommunikation: Sendet Daten unregelmäßig und empfängt kurz darauf Antworten.
    - Energieverbrauch: Sehr energieeffizient, da die meiste Zeit im Schlafmodus.
    - Anwendungsbereiche: Sensoren, Tracker und andere batteriebetriebene Geräte.
- Klasse B (bidirektionale Endgeräte mit geplanten Empfangsfenstern)
    - Kommunikation: Zusätzlich zu den Funktionen der Klasse A verfügen diese Geräte über geplante Empfangsfenster.

[5] *Quelle: https://www.thethingsindustries.com/media/pagedown-uploads/e0cd1c32-27c8-4291-a220-fcbc903994a0.png*

  - Energieverbrauch: Höher als Klasse A, da regelmäßige Empfangsfenster Energie benötigen.
  - Anwendungsbereiche: Geräte, die regelmäßige Datenupdates benötigen.
  - Hinweis: Diese Geräte sind praktisch selten.
- Klasse C (bidirektionale Endgeräte mit ständigem Empfang)
  - Kommunikation: Ständige Kommunikationsbereitschaft, außer beim Senden.
  - Energieverbrauch: Am höchsten, da das Gerät ständig aktiv ist.
  - Anwendungsbereiche: Netzstrombetriebene Geräte wie industrielle Steuerungssysteme oder smarte Straßenbeleuchtung.

Jede Klasse erfüllt spezifische Anforderungen und ist für unterschiedliche Anwendungsfälle geeignet, wobei der Energieverbrauch und die Kommunikationsmöglichkeiten die Hauptunterscheidungsmerkmale sind.

4.2.2 Gateway

Gateways sind Datenkonzentratoren mit verschiedenen Schnittstellen. Diese Geräte empfangen auf der Luftschnittstelle über ihre Antenne Datenpakete im definierten Frequenzbereich. Alle Daten, welche hier empfangen werden, werden 1:1 an den LNS weitergeleitet. Dafür existieren verschiedene Protokolle. Am gängigsten ist der Einsatz des Protokolls UDP, einem minimalen, verbindungslosen Netzwerkprotokoll. In der letzten Zeit werden aber auch immer mehr Geräte verfügbar, welche MQTT oder Basic Station anstelle des Protokolls UDP nutzen.

Basic Station ist ein neues Protokoll. Es bietet gegenüber der bisherigen Gateway Anbindung mittels UDP-Protokoll folgende Vorteile:

- zentralisiertes Update- und Konfigurationsmanagement
- TLS und tokenbasierte Authentifizierung
- zentrale Verwaltung der Kanalbelegung (Channel Plan)
- keine Abhängigkeit vom lokalen Zeitmanagement

Allerdings schränkt Basic Station auch den Gateway Betreiber ein. So kann hier zum Beispiel kein Multi Paket Forwarding (Verteilen der eingehenden Datenpakete an mehrere LNS) mehr erfolgen, da man nicht mehr manuell mehrere LNS-Serveradressen eintragen kann. Hinweis: Dies scheint abhängig vom Basic Station Anbieter zu sein. Laut des Anbieters Kerlink ist mit deren Protokoll trotzdem ein Multi Packet Forwarding möglich.

4.2.3 LoRa-Netzwerk-Server

Der LoRa-Netzwerk-Server (LNS) definiert das zu nutzende Netzwerk. Hier bestehen die Möglichkeiten des Einsatzes eines privaten Servers, den eines Dienstleisters oder eines Shared Service bis hin zum Community-Server freier, offener Netzwerke. Die Charakteristika privat, Shared oder Public sind dabei unabhängig von der Betriebsart, wo und wie diese Server betrieben

werden. Sie beziehen sich auf den Charakter des Funknetzes und der Betreiber ist sehr wohl in der Lage, auch ein privates Netzwerk öffentlich frei zugänglich zu machen.

4.2.3.1 Privater Server

Als privaten LNS betrachtet man einen Netzwerkserver, welcher ein geschlossenes LoRaWAN-Netzwerk zur Verfügung stellt. Dies bedeutet, nur der Betreiber des Servers ist in der Lage, neue Geräte (Nodes und auch Gateways) in dieses Netzwerk zu integrieren. Dies hat jedoch keine Auswirkung auf die Übertragungssicherheit bzw. das Erreichen der Schutzziele wie Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität der Datenpakete. Hierfür ist ausschlaggebend, dass mit der Nutzung der Funkfrequenz 868MHz ein Shared Medium genutzt wird, welches direkten Einfluss auf diese Ziele, insbesondere die Verfügbarkeit hat.

Das Ziel Verfügbarkeit kann jedoch durch den Eigenbetrieb des LNS und der Gateways für alle Bereiche mit Ausnahme der Luftschnittstelle deutlich besser adressiert werden, da man hier durch die Schaffung von Redundanzen sowie die Umsetzung von Notfall-Szenarien selbst direkt Einfluss nehmen kann. Für die Erreichung der anderen Ziele sind im LoRaWAN-Protokoll bereits Mechanismen umgesetzt.

4.2.3.2 Shared-Service

Das Ziel der besseren Verfügbarkeit kann auch bei der Nutzung eines Dienstleisters den eigenen Anforderungen entsprechend adressiert werden, hier ersetzt man jedoch die eigenen Aufwände und Ressourcen durch finanzielle Aufwände, welche für höhere Service Level Agreements (SLA) ebenfalls steigen. Auch hier besteht grundsätzlich die Option des Betriebes als rein privates oder als Shared Netz. In der Variante Shared Netzwerk kommt zusätzlich der Vorteil zum Tragen, dass alle im Netzwerk registrierten Gateways gemeinsam zur Netzabdeckung beitragen. Sprechen wir beispielsweise von einem Shared Netzwerk der Partnerstädte eines Verbundes, so kann damit ein gemeinsames Netzwerk über alle Stadtgebiete ausgerollt werden. Tracker-Nodes (Sensoren zur Erfassung der aktuellen Position, GPS-Tracking) funktionieren damit in den Territorien aller Verbandskommunen.

4.2.3.3 Community-Server

Diese Variante eignet sich hervorragend für den Aufbau von POCs oder MVPs. Ein bedeutender Vorteil ist die Nutzbarkeit bereits bestehender Netzabdeckung und damit die Ausdehnung des Netzwerkes. Hier gleicht das Communitynetzwerk einem Shared Netzwerk, nur, dass die Community eine wesentlich größere Ausdehnung hat und deutlich mehr Gateways registriert sind. Ein weiterer Vorteil ist die Kostenseite, da das Communitynetzwerk meist kostenfrei zur Verfügung steht. Daraus ergeben sich aber auch eine Reihe von Spielregeln bei der Nutzung dieser Netze, welche den fairen Umgang miteinander definieren. Diese sollten eingehalten werden, möchte man seinen Platz in der Community nicht verlieren.

Weitere Informationen zu diesem Thema sowie Möglichkeiten zum zeitgleichen Einsatz unterschiedlicher Varianten finden sich im Abschnitt 2.4 Variantenvergleich private vs. freie Netze

4.2.4 Geografische Kriterien

Für die Planung des Netzwerkes sind verschiedene Aspekte relevant. Zunächst ist es wichtig, das Gebiet zu definieren, in welchem die Sensoren zukünftig arbeiten sollen. Sind diese Bereiche definiert, so sind in diesen Gebieten mögliche Objekte zu identifizieren, welche als Standort für ein Gateway genutzt werden können. Die dabei relevanten Aspekte sind neben der Bereitstellung der Spannungsversorgung der mögliche Montageort der Antenne sowie die Möglichkeit der Netzanbindung.

Um den Standort ist eine typische Netzabdeckung zu erwarten, welche auf Grund der topografischen Gegebenheit der Umgebung variiert. Aus diesem Grund gilt die Empfehlung, die erwartete Netzabdeckung mittels Mapping zu validieren. Wie ein Standort gefunden und hinsichtlich seiner Eignung geprüft werden kann, wird im Abschnitt 3, Empfehlungen für Aufbau und technische Ausstattung lokaler LoRa-Netzwerke beschrieben.

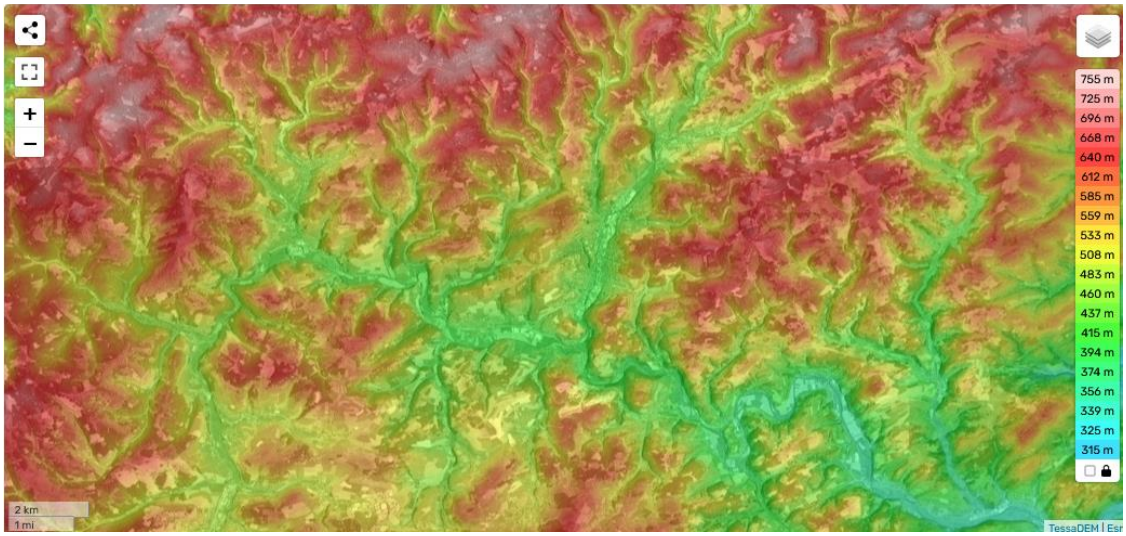


Abbildung 5 - Topografische Lage von Bad Berleburg und Umgebung[6]

Indem man auf einer Karte die Standorte der Sensoren und die erwartete Reichweite der Gateways darstellt, kann man die Gebiete ermitteln, in denen eine zuverlässige Verbindung zwischen den Sensoren und Gateways zu erwarten ist.

4.2.5 Technische Kriterien

Um eine zuverlässige Datenübertragung der Datenpakete sicherzustellen, sind einige Aspekte zu berücksichtigen. Ein wichtiger Aspekt ist, dass die Gateways so positioniert werden müssen, dass die Sensoren nicht in die hohen, zweistelligen Spreizbereiche wechseln müssen, um ihre Daten übertragen zu können. Da bei jeder Datenübertragung eines Sensors zum Gateway der

[6] *Quelle: https://de-de.topographic-map.com/map-dgskl/Bad-Berleburg/?zoom=12&center=51.04744%2C8.38054&base=5*

verwendete Spreizfaktor ermittelt und vom LNS an die Nachricht angehängt wird, kann dies mit Mitteln des Monitorings erfolgen.

Diese Maßnahme zählt bereits zur Verbesserung der Datenübertragungsqualität, welche im Abschnitt 4.5.1 ausführlich beschrieben werden.

4.3 Sicherheit in LoRaWAN-Netzwerken

LoRaWAN verwendet eine Reihe von Sicherheitsmaßnahmen zum Schutz des Netzes und der übertragenen Daten.

Eines der wichtigsten Sicherheitsmerkmale von LoRaWAN ist der Einsatz von Verschlüsselung zum Schutz von Daten auf der Luftschnittstelle. LoRaWAN verwendet AES-128-Verschlüsselung, um die zwischen Geräten und dem Netzwerk übertragenen Daten zu schützen. Neben dieser Verschlüsselung verwendet LoRaWAN eine Reihe weiterer Sicherheitsmaßnahmen zum Schutz des Netzwerks, darunter Geräteauthentifizierung und sicherer Schlüsselaustausch. Diese Maßnahmen gewährleisten, dass nur autorisierte Geräte auf das Netzwerk zugreifen können und dass die Kommunikation zwischen Geräten und dem Netzwerk sicher ist.

LoRaWAN umfasst auch Mechanismen zur Erkennung und Entschärfung von Angriffen auf das Netz, wie z. B. Denial-of-Service-Angriffe. In LoRaWAN ist der Message Integrity Code (MIC) ein Wert, der zur Überprüfung der Integrität einer Nachricht verwendet wird. Er wird vom Absender der Nachricht berechnet und in den Nachrichtenkopf aufgenommen. Der MIC wird verwendet, um sicherzustellen, dass die Nachricht während der Übertragung nicht manipuliert oder verändert wurde.

Die MIC wird mithilfe eines Algorithmus berechnet, der die Nutzlast der Nachricht, den Netzwerksitzungsschlüssel des Geräts und einige andere Werte berücksichtigt. Wenn die Nachricht empfangen wird, wird der MIC mit demselben Algorithmus und den Werten aus dem Nachrichtenkopf neu errechnet. Wenn die berechnete MIC mit der in der Nachricht enthaltenen übereinstimmt, bedeutet dies, dass die Nachricht nicht verändert wurde und authentisch ist. Stimmen die MICs nicht überein, bedeutet dies, dass die Nachricht möglicherweise manipuliert wurde und der Empfänger die Nachricht verwerfen sollte. Die Verwendung der MIC ist ein wichtiger Teil der Sicherheitsmaßnahmen im LoRaWAN, da sie dazu beiträgt, die Integrität und Authentizität der über das Netz übertragenen Nachrichten zu gewährleisten.

Ein weiteres Sicherheitselement ist der “Frame Counter”, ein Feld im Nachrichtenkopf, das dazu dient, Wiederholungsangriffe (Replay Attacks) zu verhindern. Es handelt sich dabei um einen Zähler, der bei jeder von einem Gerät übertragenen Nachricht erhöht wird und im Nachrichtenkopf enthalten ist, um sicherzustellen, dass jede Nachricht eindeutig ist. Wenn ein Angreifer versucht, eine zuvor übermittelte Nachricht erneut zu senden, kann das Netz den Angriff erkennen, indem es den Rahmenzähler in der empfangenen Nachricht mit dem erwarteten Wert vergleicht. Wenn der Frame Counter nicht dem erwarteten Wert entspricht, kann das Netz daraus schließen, dass es sich um einen Wiederholungsangriff handelt und die Nachricht ignorieren oder zurückweisen. Der Frame Counter ist ein wichtiges Sicherheitsmerkmal in LoRaWAN-Netzwerken, da er dazu beiträgt, Angreifer daran zu hindern, alte Nachrichten erneut zu senden, um sich unbefugten Zugang zu verschaffen oder das Netzwerk zu manipulieren.

Grundsätzlich hängt die Sicherheit der Kommunikation über LoRaWAN von der Vertraulichkeit der Schlüssel sowie von deren Austauschprozessen zwischen Lieferanten und Betreibern der Sensoren sowie der sauberen Implementierung und Konfiguration der Sicherheitsmechanismen der involvierten Systeme ab. Dies bedeutet u.a. den Einsatz von verschlüsselten Verbindungen vom LNS zu den Backendsystemen, ein Nutzer- und Rollenkonzept für den LNS sowie dessen konsequente Umsetzung. Anstelle der Nutzung lokaler Accounts können die Zugriffe des TTI-LNS mit in die zentrale Nutzerverwaltung einer UDSP eingebunden werden. TTI erlaubt die Nutzung externer auth. Provider wie z.B. Keycloak, Details finden sich hier: (https://www.thethingsindustries.com/docs/reference/components/identity-server/#authentication-providers)

4.4 Private vs. freie Netze

Alle Nodes, welche den LoRaWAN-Standard unterstützen, können sowohl in private als auch in freie Netze eingebunden werden. Gleiches gilt für die Lora-Gateways, wobei einige Betreiber privater Netzwerke weitere Restriktionen bis hin zum ausschließlichen Einsatz eigener Gateways definieren. Diese Vorgehensweise wird unter anderem durch den Betrieb der Gateways sowie deren Monitoring bzw. spezielle Konfigurationen des eingesetzten LNS begründet. Jede Einschränkung bedeutet jedoch auch einen kleinen Vendor-Lock-In und sollte genau hinterfragt werden.

Private Netzwerke müssen an dieser Stelle weiterhin in dedizierte und Shared Netze unterschieden werden.

Folgende Kriterien sollten beachtet werden:

- Anforderung hinsichtlich der geografischen Abdeckung des Netzwerkes
- Anforderung hinsichtlich der Verfügbarkeit des Netzwerkes
- eigene Kapazitäten zum Aufbau/ Wartung der Gateways
- eigene Kapazitäten zum Betrieb der Backend-Infrastruktur
- Einbeziehung bereits vorhandener Geräte in Partnernetzwerken (TTN/TTI)
- Einbeziehung der Bevölkerung - Citizen Science
- Vorgaben des Fördermittelgebers, sofern vorhanden
- Kosten und verfügbares Budget
- Vernetzung mit benachbarten Projekten/ Initiativen

4.4.1 Vorteile privates Netzwerk

Mit einem privaten Netzwerk hat der Betreiber alle Komponenten im eigenen Verantwortungsbereich und kann damit auf die folgenden Kriterien direkten Einfluss nehmen:

- Verfügbarkeit
- Performance
- Skalierung
- Supportmodell des SW-Entwicklers
- Eingesetzte Software

So kann z.B. frei zwischen den verfügbaren LNS-Stacks gewählt werden. Dabei sind aktuell folgende Produkte verfügbar:

- **The Things Stack** - oder auch kurz TTS, der Softwarestack hinter TTN/TTI, Open Source, Enterprise inkl. Herstellersupport verfügbar
- **ChirpStack** - freier Open Source, nur Communitysupport verfügbar, Produkt eines einzelnen Entwicklers
- **Loriot** - kommerzielles Produkt mit entsprechenden kostenpflichtigen Supportplänen

Diese Stacks können selbst betrieben oder durch einen Dienstleister betrieben werden. Die Netze verschiedener Dienstleister wie Zenner, MVV, Lechwerke setzen teilweise auf die o.g. Netzwerkserver.

Private Netzwerke benötigen regulär eine eigene Net-ID. Um diese zu erhalten, ist eine Mitgliedschaft in der LoRa-Alliance erforderlich, welche diese IDs an seine Mitglieder vergibt. Die Kosten dieser NetIDs werden in Abhängigkeit der Netzwerkgröße berechnet. Weitere Informationen zur Mitgliedschaft in der LoRa-Alliance, inkl. der entstehenden Kosten, finden sich auf der Webseite der Alliance. Einige Betreiber privater Netzwerke operieren auch im Testbereich der NetIDs, was jedoch einen Datenaustausch mit anderen Netzwerken unmöglich macht. (Packet Broker) Weitere Informationen zur NetID sind auf der Homepage der Lora-Alliance.

4.4.1.1 Dediziertes vs. Shared Netz

Anbieter privater Netze bieten oft einen eigenen Mandanten / eine dedizierte Stackinstanz innerhalb eines Shared Netzwerkes an. Dies bedeutet, die Daten werden analog freier Netze an die Stacks aller Kunden des Dienstleisters verteilt, der Stack mit dem passenden Schlüsselmaterial ist in der Lage die Daten zu entschlüsseln, während die anderen die Datenpakete verwerfen. Oft kommt auch ein zentraler LNS zum Einsatz und nur die Datenhaltung der Sensoren und Sensordaten wird in einzelne Mandanten strukturiert. In diesen Fällen arbeiten alle Kunden unter einer gemeinsamen NetID. Diese Konstrukte bieten lediglich den Vorteil, dass die Stamm- und Bewegungsdaten der Nodes durch die Mechanismen der Mandantenteilung des Systems separiert werden und damit Skaleneffekte für den Betreiber entstehen, welche dieser mehr oder weniger stark an seine Kunden weitergibt.

4.4.2 Vorteile freies Netzwerk

Ein freies Netz zeichnet sich durch einen offenen, kostenfreien Zugang für jeden aus. In der Regel handelt es sich dabei um Communitynetze. Diese Community betreibt die notwendige Infrastruktur und erlässt für die Nutzung dedizierte Regeln. Oft wird die Infrastruktur durch kommerzielle Anbieter zur Verfügung gestellt, welche aus verschiedenen Motivationen handeln.

So steht hinter der The Things Network-Community The Things Industries. Hier ist die Motivation klar, durch TTN einen niedrigschwelligen Einstieg in das Thema LoRa zu schaffen und gleichzeitig durch eine Integration der beiden Netze (TTN & TTI) einen einfachen Übergang aus dem Communitybereich in die kommerziellen Angebote zu offerieren. Gleichzeitig stellt die Netzkopplung ein wichtiges Argument pro TTI dar: Alle TTI-Datenpakete, welche durch TTN-Gateways empfangen werden, können via Packet Broker ins TTI übermittelt werden. Damit stellt die Menge der communitybetriebenen TTN-Gateways einen Großteil der weltweiten

Netzabdeckung auch für TTI sicher. Da es sich bei dieser Option um ein kostenpflichtiges Feature handelt, sollte im Vorfeld geprüft werden, ob auf die Dienste der Communitygateways zurückgegriffen werden soll und kann.

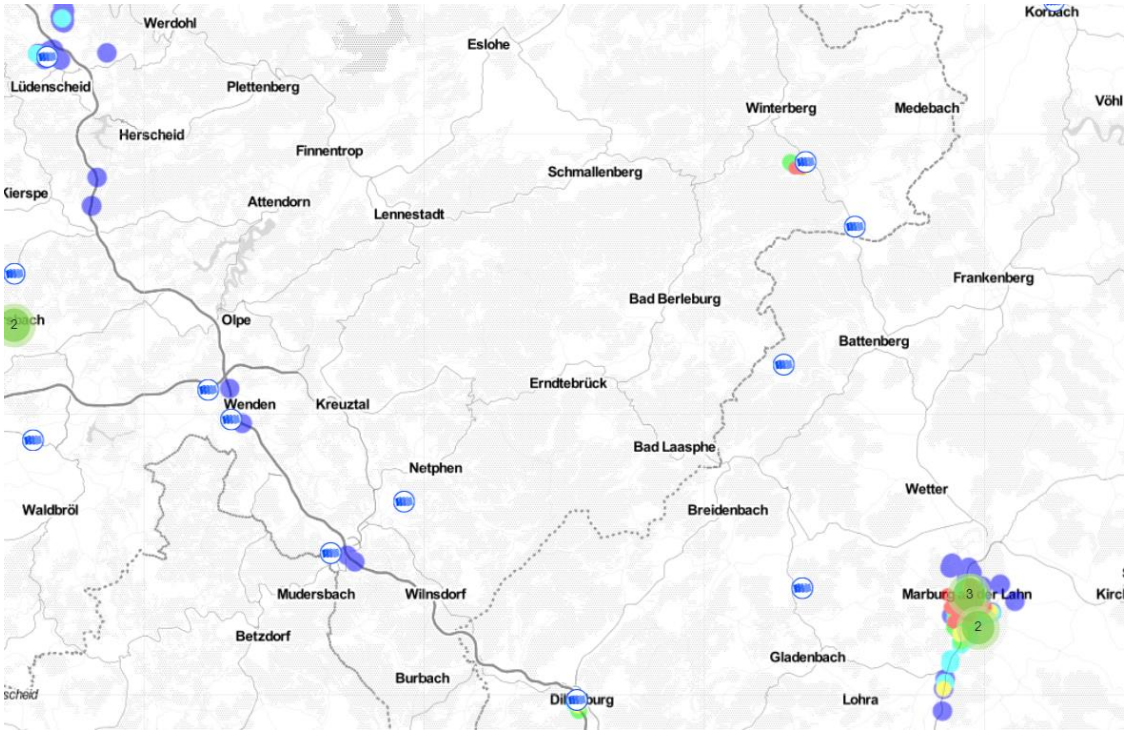


Abbildung 6 - Übersicht über TTN Gateways in einer Region (Stand 08.12.22)[7]

Bei Dienstleistungsangeboten für "eigene Netze" ist es daher wichtig immer zu prüfen, ob wirklich eine eigene NetID enthalten ist oder ob das Angebot lediglich eine eigene Instanz der Komponenten hinter einem, durch den Anbieter zentral betriebenen LNS beinhaltet. TTI ist definitiv die letztere Variante, der Anbieter geht aber auch offen mit dieser Tatsache um.

4.4.3 Kopplung privater und freier Netzwerke

Durch die gezielte Auswahl der zu nutzenden Netzwerke lassen sich Synergien nutzen. So sind Aktivitäten im Kontext des Citizen Science (vgl. 7) überwiegend im offenen Community Netzwerk TTN verortet, während kommunale IoT-Geräte auf Grund ihrer prozessualen Relevanz nicht selten in ein kommerzielles Netzwerk mit SLAs für die Rechenzentrum-Komponenten integriert werden.

Durch die mögliche Kopplung von TTN und TTI können LoRaWAN-Gateways, welche in dem einen Netzwerk angemeldet sind, auch Daten für das andere Netzwerk empfangen und an den Adressaten übermitteln. Damit ergeben sich starke Synergieeffekte: Die Kommune kann mit ihren Gateways und der damit verbundenen Netzabdeckung das bürgerliche Engagement im Stadtgebiet

[7] Quelle: *https://ttnmapper.org/heatmap/*

fördern, TTN-Gateways interessierter und engagierter Bürger können weiße Flecken auf der Abdeckungskarte füllen.

Eine Win-Win-Situation für beide Seiten.

Es gibt auch Möglichkeiten, weitere Netze anderer Betreiber zu koppeln. Ein Beispiel dafür ist der Packet Broker, hinter diesem steht jedoch auch ein kommerzielles Modell zur Kompensation der Aufwände, welche der anzubindende Netzbetreiber mit der Übermittlung der fremden Daten hat. In der Regel handelt es sich um einen sehr niedrigen Cent-Betrag je übertragener Nachricht.

4.5 Hinweise zum Umgang mit Funknetzen

Funknetze sind grundsätzlich Best-Effort-Netze. Wie bereits in einem der vorherigen Abschnitte beschrieben zeichnet ein solches Netzwerk aus, dass alle Beteiligten an der Kommunikation ihr Bestes tun, damit die Nachricht vom Sender bis zum Empfänger kommt - es jedoch dafür keine Garantie gibt. Bestes Beispiel sind Sonnenstürme. Hier wird die Erde durch massive elektromagnetische Stürme getroffen, welche ihren Ursprung in der Sonne haben. Bei extremen Ereignissen dieser Art wird nicht nur techn. Infrastruktur auf der Erde in Mitleidenschaft gezogen, es gibt auch regelmäßig Störungen der Funkübertragung von Daten.

Mit diesem Wissen obliegt es dem Betreiber des Funknetzes in Abhängigkeit der Anforderungen der realisierten Anwendungsfälle eine entsprechende Vorsorge zu betreiben. Wie das konkret aussehen kann, wird im folgenden Abschnitt Möglichkeiten zur Steigerung der Übertragungsqualität beschrieben.

Auch muss bei der Umsetzung der Anforderungen genau darauf geachtet werden, ob und wenn ja mit welchem Aufwand die erforderlichen Parameter erfüllt, werden können. Als Beispiel ist hier die Pegelmessung von Bächen und anderen kleinen Fließgewässern zu benennen. Hier ist die Zielstellung des Anwendungsfalls wichtigstes Kriterium für eine Möglichkeit der Umsetzung.

Es gibt zwei mögliche Zielstellungen für einen solchen Anwendungsfall.

- Ergänzung/ Vervollständigung der Messwerte der offiziellen Messstellen der Landesbehörden
- Alarmierung der Bevölkerung im Rahmen des Katastrophenschutzes - Verhinderung von “Gefahren für Leib und Leben” der Anwohner

Die Anforderungen generieren je nach Zielstellung sehr unterschiedliche Aufwände. Variante 1 ist dabei wesentlich einfacher und mit weniger Aufwand umzusetzen als Variante 2. Allein auf Grund möglicher Haftungsthemen wird Variante 2 zu zahlreichen Diskussionen führen - bei einer internen Realisierung mit den dann dafür Verantwortlichen Personen und bei Fremdvergabe in Form von entsprechenden Aufschlägen für die Schaffung techn. Redundanzen sowie Risikoabsicherung.

Ein weiteres Beispiel ist die Einführung der LoRa-Funktechnik in den Bereich der Rauchwarnmelder. Hier ersetzt die Funktechnologie nicht die Alarmierung im Brandfall. Die Alarmierung bei Gefahr für Leib und Leben, wird wie bisher über ein optisches und akustisches Signal direkt am Rauchwarnmelder realisiert. LoRaWAN ist als Funknetze eine Best-Effort-Technologie – die Übertragung kann also nicht zugesichert werden.

Was mit der Funktechnik abgedeckt wird, ist die jährliche Begehungs- und Prüfverpflichtung. Der Sensor überwacht selbst, dass er in seiner Halterung an der Decke sitzt, die Einbauvorschriften weiterhin erfüllt sind (Freiraum von 60cm rund um den Sensor), die Batterie in gutem Zustand ist und das Gerät funktioniert. Damit spart der Vermieter genau diese Aufwände - kommt jedoch mit der kontinuierlichen Information sofort ins nächste Thema. Die gesetzliche Verpflichtung des Vermieters fordert eine jährliche Begehung und dementsprechend dann die zeitnahe Reaktion auf Mängel. Werden jetzt sehr zeitnah Fehlermeldungen an den Vermieter kommuniziert, muss dieser genauso zeitnah reagieren. Was passiert im Brandfall mit Personenschaden, wenn dem Vermieter nachgewiesen werden kann, dass er von einem Defekt der Rauchmelder Kenntnis hatte?

All dies sind Themen, derer man sich bewusst sein muss und welche man im Rahmen der Definition der Anwendungsfälle dringend betrachten sollte.

4.5.1 Möglichkeiten zur Steigerung der Übertragungsqualität

Grundsätzlich muss der Betreiber des Netzwerkes und damit auch in der Regel der Operator der LoRa-Gateways diese Gateways monitoren. Ausschlaggebend für eine stabile Datenübertragung sind dabei folgende Aspekte:

- Auslastung der einzelnen Kanäle der Luftschnittstelle
- KPIs des Gateways (CPU-, Speicherauslastung)
- Stabilität und Latenzen der WAN-Anbindung des Gateways
- genutzte Spreizfaktoren, RSSi und SNR der Datenübertragung

Je nach Typ und Hersteller des Gateways verfügen diese über verschiedene Möglichkeiten des Monitorings. Einige stellen Daten für ein SNMP-Monitoring zur Verfügung, welches mit standardisierten Auswertungstools wie z.B. Zabbix ausgelesen werden kann. Andere Hersteller stellen diese Informationen proprietär z.B. über eine herstellerspezifische Cloud oder eigene (meist kostenpflichtige) LoRa-Netzwerkserver zur Verfügung. Auch die Performance der Gateways ist abhängig vom eingesetzten Modell sowie dem gewählten Hersteller.

Bei Problemen auf der Luftschnittstelle sind folgende Maßnahmen in Betracht zu ziehen:

- Nachverdichtung mit weiteren Gateways
- Optimierung der Sende-/ Empfangssituation durch Verbesserung der Antennenposition beim Sender und oder Empfänger
- Nutzung höherer Spreizfaktoren durch ADR (erfolgt i.d.R. selbstständig)

Das Monitoring der Luftschnittstelle kann entweder über Daten erfolgen, welche direkt aus dem Gateway bereitgestellt werden oder der LNS stellt entsprechende Informationen und Metriken bereit.

Mit den, im TTN/TTI verfügbaren Daten kann sowohl ein Monitoring aus Sicht eines einzelnen Nodes (Wie gut bekommt dieser Node seine Daten an den LNS übertragen) als auch aus Sicht der Gateways (wie viele Nodes senden regelmäßig an das Gateway und wie sieht die geografische Abdeckung aus) durchgeführt werden. Eine weitere Grundlage für eine Abdeckungsoptimierung ist eine Abdeckungskarte. Eine solche Karte kann entweder über eine, in der eigenen UDSP vorhandenen, GIS-Komponente oder durch die Nutzung frei verfügbarer Komponenten

(https://ttnmapper.org) implementiert werden. Hier wird visualisiert, von welchen Positionen eine Verbindung mit dem Netzwerk über welche Gateways mit welcher Übertragungsqualität erreicht wurde.

Zuletzt kann auch die Aktivierung der bestätigten Nachrichten ein Mittel sein, um die erfolgreiche Übertragung von Nachrichten zu überwachen. Hierbei sind jedoch die Regeln des Netzbetreibers zu beachten - so sind im TTN lediglich 10 Downlinks je Node und Tag (Nachrichten vom LNS an den Node) erlaubt. Jede Bestätigung ist dabei ein Downlink.

Mit dem Einsatz von bestätigten Nachrichten (eine Option, die entweder direkt in der Firmware des Nodes oder im Rahmen dessen Parametrierung aktiviert werden muss) wird die gesamte Kommunikationskette vom Node über das Gateway bis hin zum LNS abgesichert.

Es existieren jedoch auch weitere Handlungsoptionen im Kontext des LNS. Hier stehen Bemühungen zur Sicherstellung der Verfügbarkeit und der benötigten Performance im Scope der Betrachtungen. Da die Nodes 24/7 senden, sollte auch die Verfügbarkeit und Performance des LNS 24/7 abgesichert sein. Betreibt man den LNS On Premise, also selbst auf eigenen Servern oder in der Cloud, ist man selbst dafür verantwortlich. Wird der LNS durch einen Dienstleister betrieben, sollte ein entsprechendes SLA vereinbart werden, welches natürlich in diesem Fall auch kostenrelevant sein wird.

In der Regel lassen sich die Komponenten des LNS in einem HA-Cluster aufbauen, diese Redundanzen sichern die Verfügbarkeit und garantieren eine definierbare Performance. Völlig unabhängig vom möglichen Betriebsmodell müssen alle Komponenten automatisiert überwacht werden - im Fehlerfall muss eine Reaktion innerhalb definierter Fristen erfolgen, was geschultes Personal erfordert.

Zur Vervollständigung der Betrachtung gehört auch die Überlegung, an welchen Stellen Nachrichten temporär gesichert / gecached werden müssen, um einem möglichen Datenverlust vorzubeugen. Dabei ist auch die Kommunikationskette bis ins Backendsystem zu berücksichtigen. Dies kann z.B. über ein entsprechendes Monitoring erfolgen, welches die Löschung der temporär gesicherten Rohdaten erst nach erfolgreicher Verarbeitungsmeldung aus dem Backend erlaubt.

All diese Aufwände stehen natürlich unter dem Vorbehalt, dass sie sich durch die Anforderungen der Anwendungsfälle begründen lassen. Wer diese hohen Anforderungen einfordert, muss diese auch belegen und finanzieren können.

4.6 Verbreitete Irrtümer

Alle im Frequenzbereich 868MHz arbeitenden Nodes teilen sich die Luftschnittstelle. Dadurch ergibt sich aus der Nutzung einer speziellen Netzart (Sigfox, LoRa - privat oder Community, Mioty) keinerlei Vorteil. Sofern die Nodes den LoRaWAN-Standard unterstützen, wird auch jedes LoRaWAN-Gateway, welches sich im Sendebereich des Nodes befindet, die Nachricht empfangen und an den angebundenen LNS weiterleiten. Erst dort wird erkannt, dass diese Nachricht nicht beim richtigen Adressaten angekommen ist, die fremde Nachricht wird verworfen. Daraus ergibt sich:

- keine Erhöhung der Datensicherheit durch den Betrieb eines privaten Netzes, diese hängt ausschließlich von der Sicherstellung der Vertraulichkeit des Schlüsselmaterials der Nodes ab
- keine Absicherung der WAN-Anbindung der Gateways - da alle empfangenen Nachrichten zum LNS weitergeleitet werden, gehen alle Daten ggf. über die SIM-Karte des Gateways und belasten deren Datenvolumen.

Weiterhin ist es falsch, dass die NetID 00000 und 00001 lediglich für Testzwecke erlaubt sind. Diese beiden IDs sind eher als öffentlicher ID-Raum zu betrachten, jeder kann in diesem Bereich ein eigenes Netzwerk implementieren. Er wird jedoch niemals die Möglichkeit haben, über Mittel der Netzkopplung, wie sie die LoRa-Alliance vorsieht, Datenpakete aus anderen Netzwerken zu beziehen.

5 Empfehlungen für Aufbau und technische Ausstattung lokaler LoRa-Netzwerke

Beim Aufbau lokaler Netzwerke ist einiges zu beachten. Dies sind zum einen die geografischen und topologischen Gegebenheiten, existierende Optionen und Besonderheiten. So gilt beim Aufbau von Gateways, dass die Höhe der Antenne relevant für die Reichweite der zu erzielenden Abdeckung ist. Gehört einem aber nur das zweithöchste Gebäude in der Stadt, so muss man abwägen, ob dies nicht auch als Alternative ausreichend ist oder ob man die zusätzlichen Aufwände für ein kleines Stück mehr Reichweite in Kauf nehmen möchte. Neben solchen statischen Bedingungen spielen auch dynamische Parameter wie die Wetterverhältnisse eine Rolle, starker Schnee, starker Regen, hohe Luftfeuchtigkeit können die Abdeckung ebenfalls beeinflussen.

Es gibt verschiedenste Gateways unterschiedlicher Hersteller. Alle Geräte nutzen den LoRaWAN-Standard. Nachfolgend werden die Entscheidungskriterien für Gateways beschrieben.

5.1 LoRaWAN Luftschnittstelle

Wichtig ist bei der Beschaffung der Geräte, dass diese in dem europäischen Netzbereich 868 MHz arbeiten. Es gibt Geräte mit 1, 2 oder 8 Kanälen, wobei 8 Kanäle derzeit als Standard anzusehen sind. Einige High-End-Geräte bieten bis zu 64 Kanäle, bei denen die 8 Kanäle gesplittet werden und somit eine höhere Verarbeitungsbreite der eingehenden Funksignale erreicht wird.

Für die reguläre Beschaffung ist es wichtig, dass die Gateways mindestens 8 Kanäle haben. Bei Geräten mit weniger Kanälen verliert man bei Sensoren, welche automatisch die Kanäle wechseln, Datenpakete, da die Gateways nicht auf allen Kanälen auf eingehende Nachrichten “lauschen”. Es gibt Sensoren, welche fest auf einen Kanal eingestellt werden können. Dies entspricht jedoch nicht der Spezifikation der LoRa-Alliance. Diese sieht das dynamische Verfahren Adaptive Data Rate (ADR) vor, bei dem ein Node in einem niedrigen Bereich (höhere Bandbreite, kürzere Reichweite) beginnt und mit jedem Sendeversuch so lange einen Kanal nach oben wechselt, bis eine stabile Kommunikation zustande kommt.

Ein weiterer wichtiger Aspekt ist die Antenne. Hier gibt es sehr große Unterschiede hinsichtlich des Gewinns der Antenne, der Dämpfung sowie des Abstrahlwinkels. Diese Parameter müssen

zum Montageort der Antenne passen. Eine Antenne an hoher Stelle mit kleinem Abstrahlwinkel strahlt nicht die ganze Stadt ab und erreicht die Sensoren im Stadtgebiet nicht. Es existieren Antennen unterschiedlicher Bauarten, welche jeweils unterschiedliche Charakteristika aufweisen. Weitere Aspekte zur Auswahl einer Antenne können Vorgaben im Denkmalschutz oder andere bautechnische Gegebenheiten sein.

So kann durch die Auswahl der Antennenform und Farbe eine Anpassung an den Montageort erfolgen.

Große Antennen können einen Gewinn von bis zu 12 dBi erreichen. Solche Antennen wurden bereits ausführlich getestet, die Reichweite der Abdeckung steigt damit an. Allerdings steigt damit auch das Rauschen, welches diese Antenne miteinfängt. Diese Antennen empfangen deutlich mehr Datenpakete mit SF12 mit dem Ergebnis, dass die Luftschnittstelle schneller überlastet wird und das Verhältnis der Paketverluste ebenfalls ansteigt.

Eine Begrenzung der Sendeleistung und des Antennengewinns für die Frequenz von 868 MHz im ISM-Band sind in der Europäischen Norm ETSI EN 300 220 festgelegt. Diese Norm wird von der Europäischen Telekommunikationsnormen-Institut (ETSI) herausgegeben, das für die Standardisierung in der Informations- und Kommunikationstechnologie in Europa zuständig ist.

5.2 Hardware und Performance der Gateways

Ein Gateway ist in der Regel ein einfaches Gerät. Alle Pakete, welche über die Antenne empfangen werden, werden über das Internet an die hinterlegten LNS übertragen. Das ist die einzige Aufgabe der Gateways. Dafür reichen eine einfache Systemarchitektur und Performance.

5.3 WAN-Schnittstelle

Gateways verfügen über eine WAN-Schnittstelle, mit der sie mit dem LNS verbunden sind. Diese Verbindung kann über das öffentliche Internet, ein privates Netzwerk eines Telekommunikationsproviders oder über ein privates Netzwerk erfolgen. Über diese Anbindung laufen dann die LoRaWAN-Datenpakete, Messgrößen (Metriken) des Monitorings sowie ggf. Firmwareupdates für das Gateway. Im TTN-Kontext benötigt ein Gateway aktuell mindestens 1 GB/ Monat, wobei ein großer Teil davon die Monitoringdaten sind.

Die Schnittstelle kann dabei über Mobilfunk oder Ethernet realisiert werden. Für alternative Kommunikationsmedien (z.B. 450MHz, Powerline) muss die Verfügbarkeit der Gerätetechnik geprüft werden.

5.4 Mobilfunk

Das LoRaWAN-Protokoll beinhaltet einige zeitkritische Prozesse, welche bei Nichteinhaltung des Timings fehlschlagen. Aus diesem Grund ist es wichtig, dass die Netzanbindung performant ist und niedrige Latenzen aufweist. In einem Projekt in Bielefeld wurde nach eigener Aussage durch den Wechsel von Mobilfunk zu einer performanteren Ethernet-Anbindung eine deutlich höhere Erfolgsquote der erfolgreichen Übertragung der Datenpakete erzielt.

5.5 Ethernet

Diese Art der Anbindung stellt das Optimum dar, erfordert aber auch einige Sicherheitsmaßnahmen sowie ggfs. zusätzliche Aufwände für die Verkabelung. Das Gateway hängt mit seiner Antenne in der Regel an einem der höchsten Punkte des Gebäudes und ist damit gefährdet durch Blitzeinschläge oder elektromagnetische Störungen (EMV). Daher erfordert eine Antenne, welche auf dem Dach eines Gebäudes installiert wird, besondere Beachtung hinsichtlich Blitzschutz und Überspannungsableitung bei der Integration ins Hausnetz. Das ist auch konkret einer der Gründe, warum beispielsweise das LoRa-Gateway auf dem Dach eines Rechenzentrums über Mobilfunk und nicht ein Kupferkabel mit dem LNS verbunden ist.

Weiterhin ist es wichtig, dass die Geräte den vollen Funktionsumfang der technischen Spezifikation der LoRa-Alliance abdecken (z.B. die Unterstützung von Multicast, Multipacketforwarder). Der Bedarf für diese Funktionen ergibt sich aus den umzusetzenden Anwendungsfällen.

5.6 Multicast

Bei Multicast-Nachrichten sendet man vom LNS ein Datenpaket. Dieses wird jedoch von allen Gateways versendet und von allen Sensoren, welche das passende Schlüsselmaterial besitzen, verarbeitet. Ein klassischer Anwendungsfall für Multicast ist die Steuerung der Straßenbeleuchtung, möchte man hier alle Schaltstellen einer Region schalten, kann man dies mit einem entsprechenden Multicast-Datenpaket tun.

5.7 Multipacketforwarder

Hier werden im Gateway mehrere unterschiedliche LNS eingetragen. Alle eingehenden Datenpakete werden an alle eingetragenen LNS gesendet. Auch das ist neben dem Packet Broker eine weitere Möglichkeit, mehrere unterschiedliche LoRaWAN-Netze mit einem Gateway zu unterstützen. Diese Vorgehensweise multipliziert allerdings das Datenvolumen der WAN-Anbindung mit der Anzahl der eingetragenen LNS und, viel wichtiger, alle LNS teilen sich die Air Time für Nachrichtenübertragungen vom Gateway zu den Nodes. Damit sind für einen rechtskonformen Betrieb des Gateways strengere Regeln vorzusehen, da die Einhaltung des Duty Cycles in der Regel netzwerkintern gemanaged wird und nun mehrere Netzwerke ein Gateway nutzen, für welches die Vorgaben unabhängig von der Anzahl der angebundenen Netzwerke gelten. Eine Möglichkeit kann z.B. die Begrenzung der Anzahl Downlinks pro Tag und Netzwerk sein, wie es bei TTN realisiert wird. Eine solche Realisierung ist jedoch aufwändig und wird durch keinerlei Standardfunktionen seitens des Gateways oder der Netzwerkserver unterstützt.

5.8 Standortauswahl

Der beste Weg, einen optimalen Standort für ein Gateway zu finden, ist eine Begehung vor Ort. Wenn man das Gebiet kennt, in welchem sich später die einzelnen Sensoren befinden sollen, kann man die abzudeckende Fläche definieren. In diesem Gebiet sucht man dann nach möglichen Montageorten, optimalerweise auf eigenen, kommunalen Gebäuden mit Internetanbindung. Eine Empfehlung ist dabei, Gateways temporär aufzustellen und mit Mappern die tatsächliche Abdeckung zu erfassen. Der Mapper ist ein Node im Netzwerk, welcher zyklisch seine GPS-

Position sendet. Da bei jeder Datenübertragung die Qualität der Übertragung mit übermittelt wird, kann anhand dieser Daten eine Karte mit der Qualität der Abdeckung erstellt werden. Es gibt auch Simulationsprogramme, mit denen eine zu erwartende Netzabdeckung berechnet und in einer Karte abgebildet wird. Die Qualität dieser Simulation hängt nicht zuletzt von der Qualität der GIS-Informationen ab. Auch existierende Störquellen sind solchen Systemen in der Regel nicht bekannt und werden daher nicht berücksichtigt. Die Tatsache, dass selbst Wetterereignisse Einfluss auf die Reichweite der Funkübertragung haben, macht deutlich, dass eine tatsächlich überprüfte Datenübertragungsrate besser ist als eine berechnete Vorhersage.

Durch das Mapping entstehen Karten, in denen man die Areale mit der Qualität ihrer Abdeckung erkennen kann.

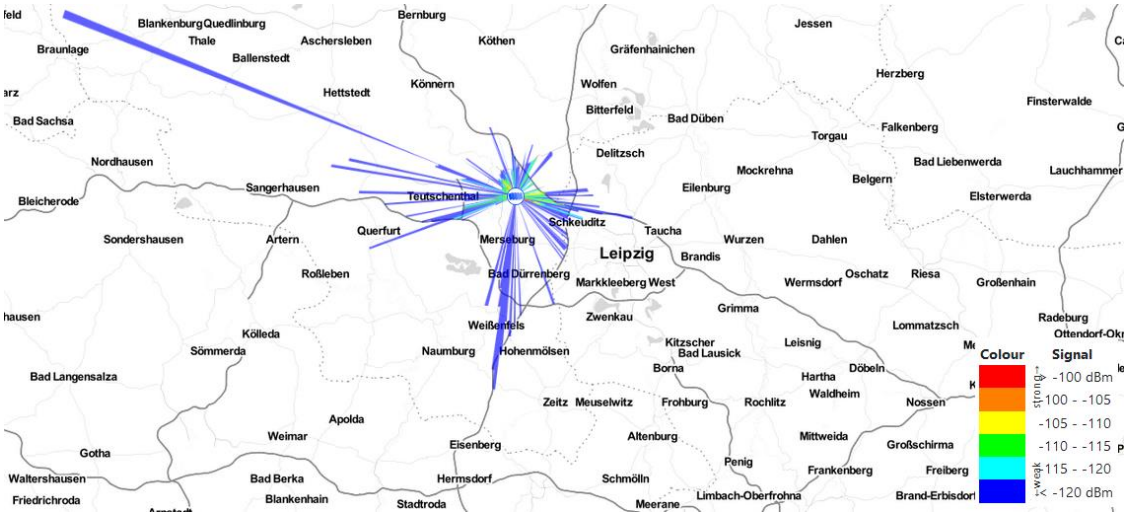


Abbildung 7 - Ergebnis des Mappings eines Gateways, Reichweite[8]

[8] *Quelle: https://ttnmapper.org/radar/gateway/?gateway=gisa-0001-00&network=NS_TTS_V3://ttn@000013*

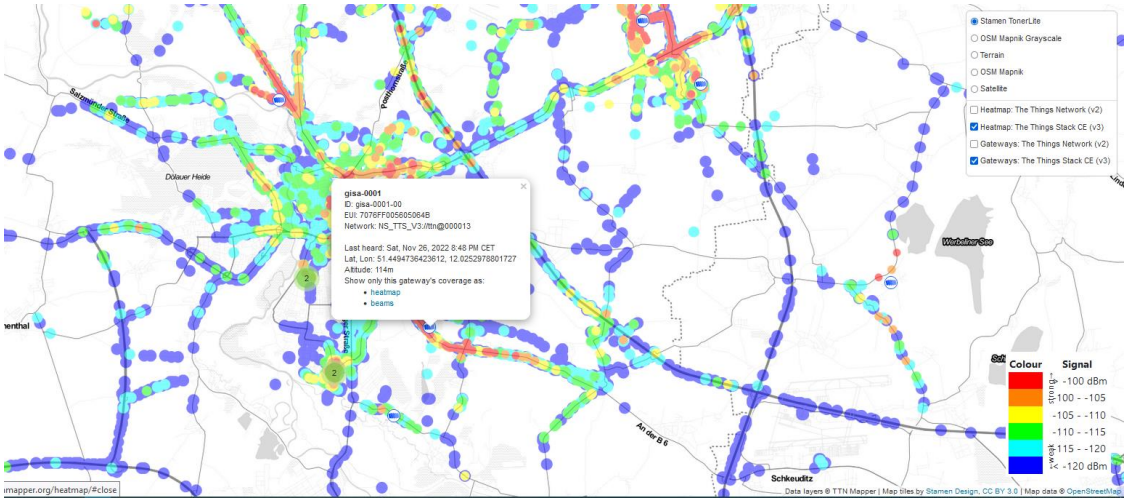


Abbildung 8 - Ergebnis des Mappings eines Gateways, Abdeckung[9]

Um ein genaues Bild der Abdeckung zu erhalten, ist es hilfreich, diese Mapper temporär auf Fahrzeugen zu montieren, welche regelmäßig in dem Gebiet unterwegs sind. Dies können Fahrzeuge des ÖPNV im Stadtgebiet sein, kommunale Fahrzeuge bis hin zur Müllabfuhr bieten ebenfalls eine Möglichkeit. Eine weitere Möglichkeit stellt das Befliegen des Bereiches mit einer Drohne dar, was jedoch durch den Einsatz im öffentlichen Bereich nur durch einen kommerziellen Anbieter mit entsprechender Lizenz erfolgen kann. Eine Herausforderung dieser Methoden besteht darin, dass damit lediglich die Netzabdeckung auf der Straße, nicht jedoch innerhalb der Gebäude gemessen wird. Dieser Umstand muss bei der Bewertung der Messergebnisse berücksichtigt werden.

5.9 Technische Aspekte

5.9.1 LoRaWAN-Network-Server

Der LoRa Netzwerkserver stellt eine Kommunikationsplattform zur Verfügung, welche an die urbane Datenplattform UDSP als ein Eingangskanal angebunden werden muss. Dabei erfolgt die Konsolidierung und Persistierung der Daten in der UDSP. Die UDSP kann neben dem LNS noch weitere Datenquellen erschließen.

Daraus ergeben sich eine Reihe von technischen Aspekten, auf welche bei der Auswahl des LNS zu achten ist.

5.9.2 Offene Bereitstellung der Schnittstelleninformationen

Der LNS muss über offen spezifizierte Schnittstellen Informationen bereitstellen. Zum einen sind bidirektionale Schnittstellen zur Kommunikation der Messwerte der IoT-Geräte erforderlich. Zum anderen soll der LNS eine Schnittstelle zum Anlegen, Ändern und Löschen der IoT-Geräte und

[9] *Quelle: https://ttnmapper.org/heatmap/*

Gateways anbieten. Eine weitere Schnittstelle zum Auslesen der Monitoring-Informationen der Geräte ist zudem wichtig.

Konkret ist zu prüfen, ob und wie umfassend die beschriebenen Schnittstellen offen zur Verfügung stehen und frei zugänglich dokumentiert sind.

5.9.3 Schnittstelle Import/Export Messwerte der Nodes

Diese Schnittstelle muss performant und unter Berücksichtigung aktueller Sicherungsmechanismen einen Austausch der Messwerte zwischen LNS und der UDSP ermöglichen. Zum Einsatz sollte eine aktuelle Technologie wie MQTT, Webhook oder REST.API kommen. Die Schnittstelle muss neben den Daten vom Node zum LNS auch die Gegenrichtung vom LNS zum Node unterstützen.

Diese Integration einer MQTT-Schnittstelle zum TTI-LNS ist eine Standardfunktion der UDSP und bereits durch mehrere Beispiel-Node-RED-Flows implementiert.

5.9.4 Schnittstelle Gerätestammdaten

Diese Schnittstelle beinhaltet die Funktion des Anlegens, Änderns und Löschens der Geräte (Gateway, Node) sowie der dazugehörenden Elemente wie Application und oder API-Key einschließlich der zugehörigen Berechtigungen. Mit dieser Schnittstelle erhält der Anwender der UDSP die Möglichkeit, Geräte aus der UDSP heraus zu administrieren. Damit wird es möglich, das Rollen- und Berechtigungskonzept für das Management der Geräte in der UDSP umzusetzen und somit den LNS mandantenübergreifend zu nutzen. Alternativ lassen sich die Geräte direkt manuell in der Kommunikationsplattform (LNS) verwalten, es gibt Händler, welche diese Pflege inkl. des Tests der Einbindung als optionale Leistung beim Kauf der Geräte anbieten.

Bei dieser Funktion handelt es sich um keine Standardfunktion der UDSP, die entsprechenden Funktionen müssen im Rahmen eines Projektes in der UDSP implementiert werden.

5.9.5 Monitoring

Diese Schnittstelle stellt Metriken und Monitoring Informationen zur Verfügung, welche ein Monitoring der Gateways und Nodes aus der UDSP heraus ermöglichen. Die Implementierung dieser Schnittstelle ist eine wichtige Voraussetzung für eine mandantenübergreifende Nutzung der Kommunikationsplattform und eine deutliche Verbesserung der Usability des Gesamtsystems. Alternativ kann das Monitoring mit den im LNS ausgeprägten Bordmitteln erfolgen.

Bei dieser Funktion handelt es sich um keine Standardfunktion der UDSP, die entsprechenden Funktionen müssen im Rahmen eines Projektes in der UDSP implementiert werden.

Sowohl für die Provisionierung der Geräte als auch das Monitoring sind Funktionen vorhanden, um damit diese Aufgaben manuell bzw. direkt in der jeweiligen Kommunikationsplattform (LoRaWAN-LNS, NB-IoT-Plattform) umzusetzen. Gerade unter Berücksichtigung der Tatsache, dass unterschiedliche Kommunikationsplattformen angebunden werden sollen, ist es wichtig, zentrale Funktionen in diesen Kontexten umzusetzen. Dies steht jedoch unter dem Vorbehalt, dass die Kommunikationsplattform die entsprechenden Schnittstellen offen dokumentiert zur Verfügung stellt.

5.10 Gateways

Gateways werden in Preislagen zwischen ca. 200,00 - 6.000,00 EUR und darüber hinaus angeboten. Anhand der konkreten Anforderungen muss man das bestehende Produktportfolio im Markt entsprechend eingrenzen.

Im Communityumfeld werden seit mehreren Jahren erfolgreich Geräte des Herstellers Mikrotik eingesetzt. Diese weisen ein recht günstiges Preis-Leistungsverhältnis auf. Diese Gateways arbeiten auf Basis von RouterOS, einem Linux-OS, welches Mikrotik über viele Jahre entwickelt und verbessert hat. Da Mikrotik seine Wurzeln in der Netzwerktechnik hat, stellt dieses OS auch Dienste zur Verfügung, welche bei den Geräten anderer Hersteller nicht im Angebot sind. Konkret kann ein Mikrotik-Gateway via SNMP gemonitort werden und es besteht die Möglichkeit via VPN direkt aus der Ferne zu warten. Dies ist eine Funktion, die andere Anbieter in der Regel über eine eigene, herstellerspezifische Managementplattform lösen. Nicht selten ist diese Plattform dann auch noch entgeltpflichtig, wobei die Funktionen zum Monitoring der Gateways inkl. der Open-VPN-Anbindung auch noch nicht zum Standard der UDSP gehören, und im Projekt entwickelt werden müssten. An dieser Stelle hat man jedoch mit SNMP einen offenen Standard, auf den man aufsetzen kann.

Professionelle Anwender nutzen oft hochwertigere Geräte von namhaften Herstellern wie RAK, LoRix, Tektelik oder Kerlink.

Geräte von RAK setzen bei ihrer Software auf OpenWRT, einen Softwarestack, der ebenfalls offen ist und seit Jahren u. a. durch die Freifunk-Community erfolgreich eingesetzt und weiterentwickelt wird.

All diese Geräte zeichnet in der Regel aus, dass es sich um Gateways mit mindestens 8 Kanälen handelt, welche wahlweise verschiedene WAN-Anbindungen unterstützen. Einige hochpreisige Geräte unterstützen sogar ein Vielfaches der 8 Kanäle (16 bis zu 64 Kanäle). Dabei handelt es sich zum einen um eine granularere Gliederung der verfügbaren Frequenzen (der Frequenzbereich, welcher bei einem 8-Kanal-Gerät durch einen Kanal abgedeckt wird, wird bei einem 16-Kanal-Gerät auf zwei Kanäle und bei einem 32-Kanal-Gerät auf 4 Kanäle aufgeteilt. Geräte mit 64 Kanälen bestehen intern aus zwei 32-Kanal-Geräten.

Aktuell ergibt sich aus der größeren Anzahl an verfügbaren Kanälen lediglich der Vorteil, dass Geräte eine bessere Performance aufweisen, da diese für die höhere Anzahl an Kanälen ausgelegt ist.

Wichtig ist an dieser Stelle ist aus technischer Sicht nicht nur die Parameter auf dem Datenblatt zu berücksichtigen. Auch eine Berücksichtigung der Einsatzbedingungen sowie das Thema Nachhaltigkeit sollten Aspekte sein, welche berücksichtigt werden sollten.

So hat ein Anwender bei der Montage der Gateways diese zum zusätzlichen Schutz in ein Gehäuse montiert. Dieses hatte eine Tür aus transparentem Acrylglas, was zu einer Erhitzung der Gateways im Gehäuse auf deutlich über 80°C zur Folge hatte. Abgesehen davon, dass sich aus dieser Tatsache ableitet, keine transparenten Gehäuse zu nutzen, muss auch darauf geachtet werden, dass die Geräte einen entsprechenden Arbeitsbereich haben. Hierbei sind insbesondere sogenannte Indoor- und Outdoor-Gateways zu unterscheiden. Indoor-Gateways sind für den

Einsatz im Gebäude gedacht. Outdoor-Gateways sind primär für den Einsatz außerhalb von Gebäuden nutzbar.

Da sich der Markt sehr dynamisch entwickelt und die Verfügbarkeit der Geräte regelmäßig variiert, wird an dieser Stelle kein konkretes Gerät empfohlen. Alternativ kann im Rahmen des Projektes im Rahmen einer Abstimmung eine Empfehlung auf Basis der aktuell verfügbaren Geräte erfolgen.

Sollen die Gateways via LTE an das Backend angebunden werden, empfiehlt sich der Einsatz einer M2M-SIM. Dabei handelt es sich um ein spezielles Mobilfunkangebot für die Maschine-zu-Maschine-Kommunikation. Eine solche SIM unterstützt ausschließlich die mobile Datenübertragung, keine Sprachtelefonie oder SMS. Diese Tarif-Optionen bieten einige Mobilfunkanbieter an. Als Besonderheit gibt es hier teilweise die Option *National Roaming* - die Möglichkeit der Nutzung aller deutschen Mobilfunknetze – was die Netzabdeckung erhöht. Außerdem ist häufig das Datenvolumen für einige Jahre nutzbar – und verfällt nicht zum Ende des Monats.

Eine Anbindung der Gateways über einen privaten APN ist nur sinnvoll, sofern diese Gateways an einen privaten LNS angebunden werden, welcher im eigenen Rechenzentrum betrieben wird.

5.11 Sensorik

Wichtigster technischer Aspekt für die Sensorik ist die Umsetzung des LoRaWAN-Standards. Dieser wird idealerweise durch eine Gerätezertifizierung der LoRa-Alliance nachgewiesen. Da diese Zertifizierung jedoch mit relevanten Kosten verbunden ist, existieren zahlreiche Geräte im Markt, welche zu diesem Standard konform arbeiten, jedoch diese Zertifizierung nicht vorweisen können. Hier sollte zunächst getestet oder auf die Erfahrungen Anderer mit diesen Geräten zurückgegriffen werden.

Ein weiterer wichtiger Aspekt ist die Nachhaltigkeit der Nodes. Grundsätzlich ist die Möglichkeit des Batteriewechsels sowie von Firmwareupdates positiv zu bewerten.

Zuallererst sollte jedoch die Funktionsweise des Sensors eine hohe Abdeckung der Anforderungen des Anwendungsfalles aufweisen.

Als klassisches Beispiel sei hier ein Pegelsensor benannt. Vergleichen wir die Pegelmessung an einem Fließgewässer, einem Salzsilo des Winterdienstes und die Abstandsmessung von oben in einem Carport.

Es gibt Geräte, die messen im 12h- oder 24h- Zyklus und übermitteln das Ergebnis an den LNS, andere Geräte messen wesentlich kurzzyklischer und senden jedoch nur bei Erreichen vorher definierter Grenzwerte und die dritte Art misst kurzzyklisch und sendet bei jeder Veränderung.

Damit sind die Geräte mit Grenzwertüberwachung (Anstieg im Messzyklus bzw. absoluter Pegel) besser geeignet für einen Einsatz als Pegelsensor an Bächen und Flüssen, da erstere Geräte einen starken Anstieg zwischen den beiden lang auseinanderliegenden Messungen gar nicht registrieren würden. Erstere Geräte sind prädestiniert für den Einsatz u.a. in Salzsilos des Winterdienstes. Die letzten Geräte eignen sich dagegen mit ihrem Sendeverhalten für die Parkplatzüberwachung.

5.12 Backend-System

In diesem Abschnitt werden einige zentrale Aspekte zur Steigerung der Usability und der einheitlichen Bedienbarkeit der Kombination UDSP und LNS beschrieben.

Zum aktuellen Zeitpunkt ist die Vielzahl dieser Funktionen in LNS und UDSP in verschiedenen Bestandteilen der Gesamtlösung vorhanden, der Anwender muss wissen, was er wo erreichen kann. Aus diesem Grund empfiehlt dieses Konzept eine Betrachtung der verschiedenen Prozesse und eine schrittweise Integration der einzelnen Prozesse in eine zentrale Oberfläche mit einheitlichem Look & Feel. Mit den, aktuell im Konzept empfohlenen Komponenten ist diese Vorgehensweise realisierbar, es werden die dafür erforderlichen Schnittstellen offen bereitgestellt und sind gut dokumentiert. Bei der Wahl anderer Komponenten ist daher immer zu prüfen, ob und in welchem Umfang eine solch tiefe Integration durch den Urheber der Komponente ermöglicht und unterstützt wird.

5.12.1 Device Management

Erste, einzelne Geräte können, ja sollten sogar manuell in der Kommunikationsplattform angelegt werden. Nur so erlernt man die Basics, welche für ein späteres Verständnis der Gesamtzusammenhänge wichtig sind. Auch die Verwaltung dieser Geräte lässt sich gut in dem jeweiligen LNS realisieren. Sobald jedoch erste relevante Stückzahlen an Gateways oder Sensoren für das System anstehen, ist ein zentrales Gerätemanagement ein wichtiges Ziel. Sollten unterschiedliche Kommunikationsplattformen genutzt werden, muss klar definiert sein, welcher Prozess wo abgebildet ist und auf welcher Plattform die einzelnen Geräte eingebunden wurden. Zudem ist zu beachten, dass ein zentrales Monitoring nur mit entsprechender Integration der Kommunikationsplattformen möglich ist.

5.13 Best Practices

5.13.1 LoRa-Netzwerkserver

Zum Zeitpunkt der Erstellung dieses Dokumentes (Stand: Mai 2023) lautet die Empfehlung definitiv die Nutzung des TTS – The Things Stack. Da es sich beim LNS ausschließlich um die Kommunikationsplattform handelt, ist eine Integration der Funktionen und Berechtigungen des Servers indirekt über Schnittstellen in das Rollen- und Rechtekonzeptes der UDSP sinnvoll. Damit wird eine gemeinsame Nutzung der Kommunikationsplattform durch mehrere Partner des Verbundes möglich.

Der TTS kann als Bestandteil des TTN/TTI-Netzwerkes via Cloudservice bei The Things Industries bestellt werden oder On-Premise im eigenen RZ betrieben werden. Auch der komplett kostenlose Betrieb im On-Premise-Modell ohne die NetID von TTN/TTI ist möglich, jedoch bietet diese Option nicht die Möglichkeit der Netzkopplung via Packet Broker.

Gleiches gilt für den Betrieb einer eigenen Instanz von Chirpstack. Gleichzeitig bedingt der proprietäre Betrieb in einem eigenen Rechenzentrum unter Berücksichtigung einer zu z.B. TTI vergleichbaren Verfügbarkeit den Aufbau eines hochverfügbaren Clusters, wodurch finanzielle, personelle und technische Aufwände entstehen, welche die Servicekosten eines Cloudservice bei weitem übersteigen. Gerade bei geringen Stückzahlen übersteigen die Anforderungen, welche sich

auf die Verfügbarkeit beziehen, bei weitem die Anforderungen an die Systemperformance aufgrund der angebundenen Sensormenge.

Sollte zu einem späteren Zeitpunkt eine relevante Sensoranzahl existieren, welche den Aufbau einer eigenen IT-Landschaft rechtfertigen, ist eine Migration in ein selbst gehostetes System vorstellbar und muss neu bewertet werden.

5.13.2 LoRa-Gateways

Zum aktuellen Zeitpunkt ist der Einsatz des Kerlink-Gateways Wirnet iStation zu empfehlen. Dieses Gerät ist bei zahlreichen Betreibern im Einsatz und weist eine minimale Fehlerquote auf. Auch die Geräte von Mikrotik, das wap LR8 Kit und das LtAP LR8 LTE Kit sind empfehlenswert, jedoch ist die Einrichtung deutlich aufwändiger.

Für die Geräte beider Hersteller gibt es Anbieter im Markt, welche Beschaffung sowie Einbindung in das Netzwerk des Auftraggebers anbieten. Um jedoch das Know-How auch selbst aufzubauen, empfiehlt es sich, einzelne Geräte auch selbst einmal einzurichten.

Diese Empfehlung kann zu einem späteren Zeitpunkt anders aussehen, die technische Entwicklung in diesem Bereich ist aktuell so kurzlebig, dass eine längerfristige Aussage nicht möglich ist.

Als Antennen wurden sehr gute Ergebnisse mit den Antennen von Taoglas Baracuda 868MHz mit 5dBi oder 8dBi Antennengewinn erzielt. Beim Einsatz von Überspannungsschutzmodulen sind die Module von Citel empfehlenswert. Als Antennenkabel wurden positive Erfahrungen mit Ecoflex 10 Koaxialkabel (50 Ohm) gemacht. Dieses kann über diverse Webshops maßgenau konfektioniert mit den passenden N-Steckern bezogen werden.

Ein anderer Anwender mit mehrjähriger Betriebserfahrung betreibt seine Mobilfunk-Gateways mit SIM-Karten von Telefónica mit einem gepoolten Datenvolumen von 3 GB je Karte und Monat. Auch ein Mischbetrieb mit Karten mit 1 GB und 3GB je Karte und Monat ist gepoolt möglich, hier wird das verbrauchte Datenvolumen aller Karten mit dem gebuchten Datenvolumen aller Karten abgerechnet.

Für M2M-Karten (Mobilfunkkarten für die Datenkommunikation, i.d.R. ohne Sprachoption) bekommt der Auftraggeber bei Telefónica Zugriff auf das M2M-Portal, einer Plattform, in welcher neben allen rechnungsrelevanten Themen auch die Möglichkeit der Anpassung der Karten hinsichtlich Datenvolumen, Aktivierung und Netzstatus der einzelnen Karten besteht. Ob und in welchem Zusammenhang der Zugriff auf vergleichbare Portale bei anderen Anbietern möglich ist, müsste ggf. geprüft werden, es können dabei jedoch nicht zu vernachlässigende Kosten entstehen.

5.13.3 Nodes

Hier konkrete Empfehlungen auszusprechen ist wegen der großen Anzahl unterschiedlicher Hersteller, Anbieter und Modelle nicht möglich. Grundsätzlich sollte sich über die Definition der Anforderungen und einer Betrachtung der in Abschnitt 3.1.3 - Technische Aspekte Sensorik beschriebenen Kriterien im Markt umgesehen werden. Nach Spezifikation der Anforderungen ist unter Berücksichtigung bereits vorhandener Erfahrungen sicherlich eine Empfehlung möglich, allerdings beschränkt sich diese meistens auf den eingeschränkten Bereich der bekannten Nodes.

5.13.4 Exemplarische manuelle Einrichtung eines Gateways im TTS

Um ein Gateway einzurichten, wird ein Useraccount im Tenant des jeweiligen LNS benötigt. Der Begriff Tenant kommt aus dem Englischen und bedeutet so viel wie Mieter oder Pächter. Auf unseren Kontext übersetzt kann man es am besten als Mandant oder Kundenkonto übersetzen. Innerhalb des Tenants werden User angelegt, Berechtigungen vergeben und die Geräte verwaltet. Im Communitynetzwerk TTN laufen alle Devices im Tenant „The Things Network“. Im TTI erhält jeder Kunde einen eigenen Tenant. Im Kontext mehrerer Organisationen (Kommunen, Firmen, Vereine …) wäre für TTI entweder ein übergreifender Tenant oder ein Tenant je Organisation vorstellbar – dies hängt davon ist ab, wie eng die Zusammenarbeit ist und wie wichtig eine Trennung von Admin- und Userrechten und der Daten (DSGVO) ist.

Zu einem eigenen Account im TTN kommt man durch Anmeldung unter https://thethingsnetwork.org. Einen eigenen Tenant muss man sich bei TTI unter https://thethingsindustries.com oder über einen TTI-Partner bestellen.

Zu beachten ist, dass beide Varianten auf demselben Open Source Softwarestack laufen. TTI verkauft nicht die Software, sondern den Betrieb, die Wartung und den Support, welcher in den Angeboten enthalten ist. Für beide Varianten gibt es am Ende einen Account. Es können weitere Nutzer im TTN/ dem TTI-Tenant angelegt werden, welche dann die notwendigen Berechtigungen erhalten.

Allerdings gibt es immer nur einen Hauptaccount. Grundsätzlich ist es möglich, das System mit nur einem Account/ User zu managen. Weitere Benutzer können auf Grund interner Vorgaben (Arbeiten im System grundsätzlich mittels personalisierter Accounts usw.) erforderlich sein.

Nach der Anmeldung im Tenant landet man auf folgender Seite:

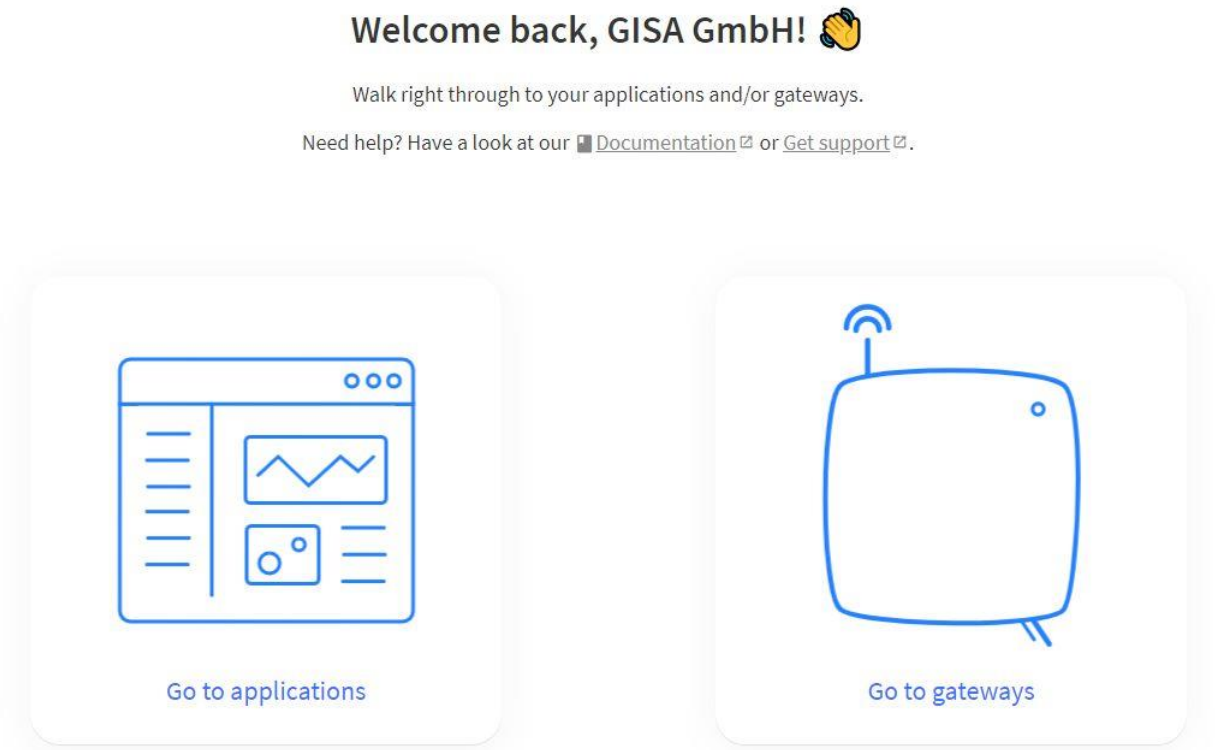


Abbildung 9 – Einstiegsseite von The Things Stack[10]

5.13.5 Exemplarische manuelle Einrichtung eines Nodes

Um einen Node im LNS anzulegen, sind verschiedene Schritte notwendig, welche in diesem Abschnitt am Beispiel des TTS beschrieben wird.

5.13.5.1 Anlage einer Application

Im TTS werden einzelne Nodes über eine Applikation gebündelt. Diese Applikation stellt die APPEUI bereit, welche alle Nodes gemeinsam nutzen können. In der Praxis wird die AppEUI jedoch schon durch den Hersteller des Gerätes definiert, im TTS können die Nodes eine, von der übergeordneten Applikation abweichende APPEUI besitzen. Die Nodes können innerhalb einer Applikation nach selbst zu definierenden Kriterien gebündelt werden, es existieren keine zentralen Vorgaben, einen Node kann man genau einer Application zuweisen. Möchte man die Funktion des Application-Payload-Decoders nutzen, dann sollte man in einer Applikation ausschließlich Nodes mit gemeinsamen Payloaddecoder anlegen. Da jedoch auch jedem Node ein eigener Payloaddecoder zugewiesen werden kann, ist auch eine Bündelung nach Eigentümer der Nodes, dem zeitlichen Verlauf der Inbetriebnahme der Nodes oder andere Kriterien denkbar.

[10] *Quelle: https://eu1.cloud.thethings.network/console/*

Weiterhin können in einer Applikation weitere zentrale Attribute definiert werden: Zugriffe anderer Nutzer des Tenants werden im Rahmen des Contributorenkonzeptes definiert, Schnittstellen zur Weitergabe der Daten aller Nodes einer Applikation im Rahmen der Integration. Weiterhin lassen sich an dieser Stelle applikationsweite Payloadformatter für Uplink (vom Node zum LNS) und Downlink (vom LNS zum Node) hinterlegen. Dabei ist zu beachten, dass die Nutzung dieser zentralen Funktionen auf Node-Ebene definiert wird. Es ist auch möglich, innerhalb einer Applikation für jeden Node einen eigenen Formatter bzw. den aus dem Noderepository des LNS zu verwenden.

Für TTN bzw. TTS empfiehlt sich ein Blick in das GitHub-Device-Repository, in dem Skripte und Payloaddecoder für eine Vielzahl unterstützter Nodes und Sensoren zu finden sind.

Im Rahmen dieses Konzeptes gilt die Empfehlung, den Raw-Payload des Sensors, d.h. den ursprünglich übertragenen Datenstring des Sensors, nicht auf der LNS-Plattform, sondern erst innerhalb der UDSP zu dekodieren. Diese Vorgehensweise bietet die Garantie, dass selbst wenn man die Formatierung des LNS nicht selbst in der Hand hat oder wenn Dritte Zugriff auf den LNS haben, sich bei der Einbindung fremder Sensoren via API im Laufe des Betriebes nichts verändert.

Einer Applikation können weiterhin einzelne Attribute zugeordnet werden.

Für die Anlage einer Applikation wird eine Application-ID sowie ein Application-Name benötigt. Die Application-ID findet sich in jedem, vom LNS übertragenen Datenpaket wieder und muss auf dem Server eindeutig sein. Der Name kann eine Applikationsbezeichnung im Klartext enthalten. In dem weiteren Feld Description kann optional ein Text zur Beschreibung der Applikation angegeben werden

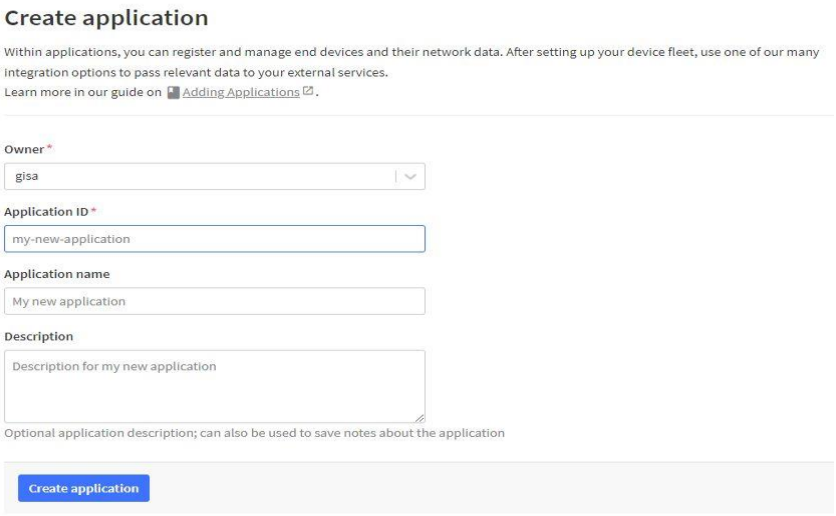


Abbildung 10 - Anlage einer Applikation im TTN[11]

[11] *Quelle: https://eu1.cloud.thethings.network/console/applications/add*

5.13.5.2 Anlage eines Nodes

Im TTS lassen sich Nodes mittels eines Wizards anlegen. Dazu muss der Nutzer zu Beginn Hersteller und Typ auswählen. Sofern der Hersteller alle relevanten Informationen hinterlegt hat, kann ein Node relativ einfach mit den Informationen aus dem Geräte-Repository angelegt werden.

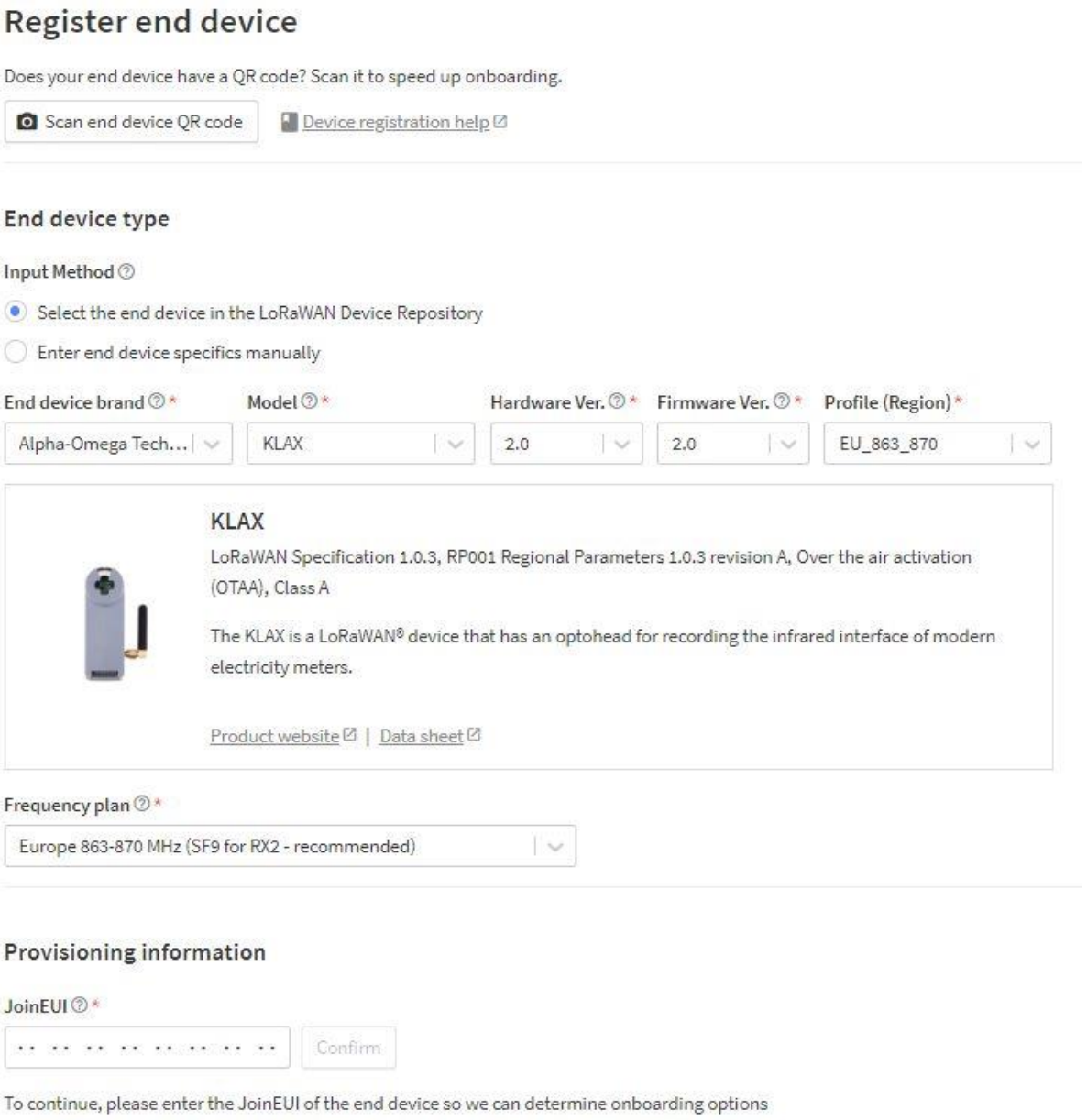


Abbildung 11 - Startbildschirm Registrierung Nodes mittels Device-Repo (TTN)[12]

[12] *Quelle: https://eu1.cloud.thethings.network/console/applications/gisa-parking/devices/add*

Verfügt der Node über einen QR-Code, kompatibel zu den Vorgaben der LoRa-Alliance, kann das Gerät auch einfach durch Einscannen dieses QR-Codes angelegt werden.

Schwieriger wird es, wenn weder QR-Code noch eine Vorlage im Geräte-Repository vorhanden sind. Dann muss der Sensor komplett manuell angelegt werden, alle abgefragten Informationen müssen gepflegt werden.

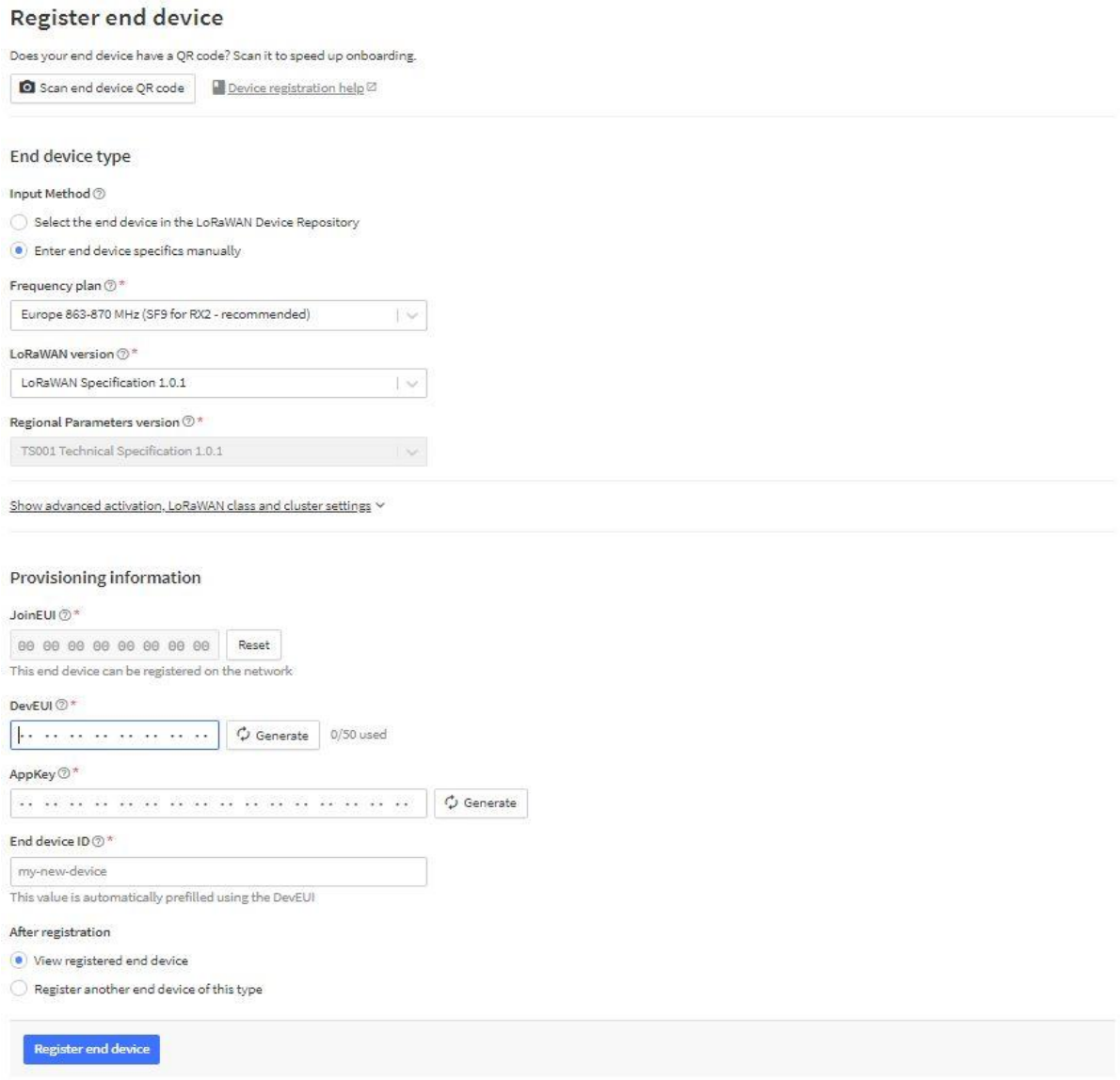


Abbildung 12 - Manuelle Anlage eines Nodes im TTN[13]

[13] *Quelle: https://eu1.cloud.thethings.network/console/applications/gisa-parking/devices/add*

Die Angaben hinsichtlich Frequenz, LoRa-Version sowie ggf. erforderlicher regionaler Parameter (in Abhängigkeit von der LoRa-Version) müssen vom Hersteller kommen, alternativ kann mit Default-Werten getestet werden. Diese Default-Werte sind in der Regel in den Auswahlboxen als solche gekennzeichnet.

Die drei Werte ‚*Join-EUI*', ‚*DEV-EUI*' und ‚*App Key*' müssen immer von Hand eingetragen werden und vom Hersteller des Sensors kommen. Alternativ kann man bei diversen Sensoren diese Werte anpassen, z.B. über Programmierung via Herstellertool oder NFC. Danach können diese im Server eingetragen werden. Wichtig - diese drei Werte sind der Vertrauensanker der Kommunikation, bei Fehlern kann keine Kommunikation aufgebaut werden bzw. können die Daten nicht entschlüsselt werden.

Die ‚*End-Device-ID*' ist später ebenfalls in jedem übermittelten Datenpaket enthalten und muss auf dem Server eindeutig sein. Für selbst gebaute Nodes besteht hier im TTN die Option, einen App Key selbst generieren zu lassen. Eine Möglichkeit, einen End-Device-Name sowie eine Beschreibung zum Node zu ergänzen, findet man nach dem Anlegen des Nodes unter ‚*General Settings*'. Hier besteht auch die Möglichkeit, weitere Attribute zu ergänzen. Neben den General Settings findet sich auch der Reiter ‚*Location*' analog der Locationsangabe beim Gateway.

Wichtig ist auch der Reiter ‚*Payload Formatter*', er trägt ggf. die Einstellungen zum Dekodieren des originären Payload (die Benennung mit „Formatter" stammt von TTI, gemeint ist die Dekodierung). Sollte der Payload des Nodes bereits auf dem LNS decodiert werden, ist hier für Uplink (Node -> Server) und Downlink (Server -> Node) getrennt jeweils ein Parser einzutragen. Entstammt das Gerät aus dem Geräte-Repository des Servers, bestehen gute Chancen, dort auch einen passenden Parser vorzufinden, welchen man mittels der Option ‚*Use Device Repository Formatter*' auf Applikationsebene auswählen kann. Im Node muss dann nur noch eingestellt werden, dass der Node den ‚*Applikationsformatter*' nutzt. Alternativ kann man für jeden Node einer Applikation auf Node-Ebene einen speziellen Formatter definieren.

5.13.5.3 Einrichten eines API-Keys zur Datenweiterleitung an die UDSP

Auf Applikationsebene kann man einen API-Key einrichten, welcher es der UDSP ermöglicht, die vorher definierten Informationen mit dem LNS für alle Geräte der Applikation auszutauschen.

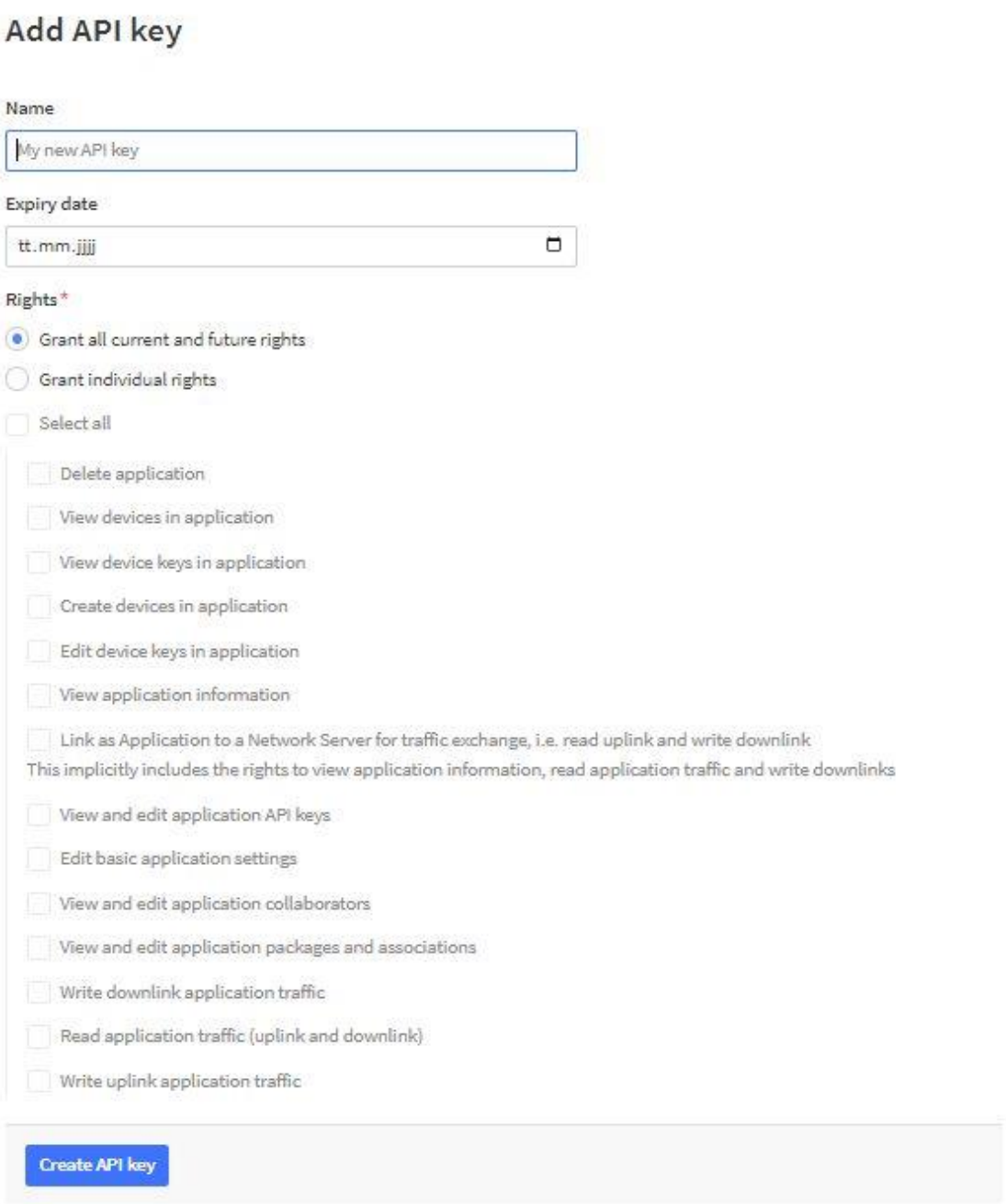


Abbildung 13 - Ansicht zum Anlegen eines API-Keys im TTN[14]

Hier muss ein Name vergeben und kann eine Befristung der Gültigkeit des Keys eingetragen werden. Weiterhin können die Rechte dieses Keys definiert werden. Wichtig ist, den API-Key am

[14]*Quelle: https://eu1.cloud.thethings.network/console/applications/gisa-parking/api-keys/add*

Ende wirklich zu sichern, denn wird das Fenster geschlossen und die Information geht verloren bleibt nur die Option, den Key neu zu vergeben.

Im Anschluss kann in der Applikation unter ‚*Integrations*' die gewünschte Technologie (MQTT, Webhook uvm.) ausgewählt werden und die Credentials entnommen werden.

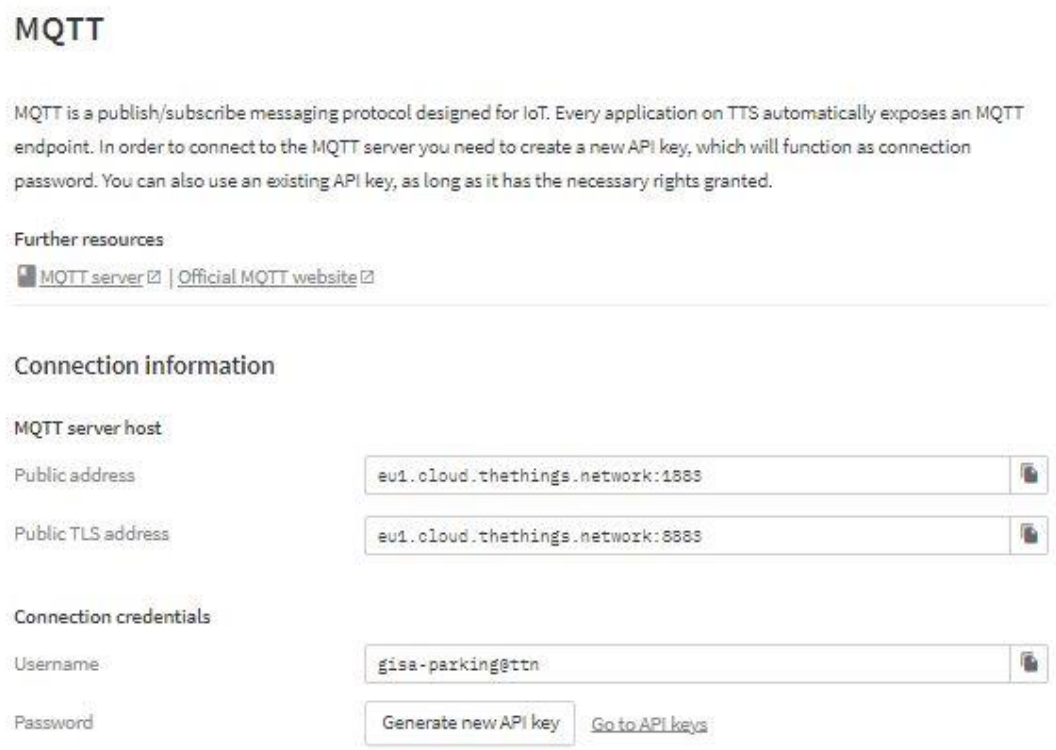


Abbildung 14 - Übersicht der Credentials der Integration (MQTT)[15]

5.14 Sonstige Aspekte

Dieses Kapitel gibt Anhaltspunkte zu den Kosten eines LoRaWANs und schlägt ein IoT-Inventar vor, welches die Wartung und Inventarisierung der Sensoren erleichtert.

5.14.1 Total Cost of Ownership

Die Total Cost of Ownership eines LoRaWANs berechnet sich aus:

- Kosten des LoRa-Servers (TTI, NIOTA, …)
- Installation und Wartung der Gateways

[15]Quelle: https://eu1.cloud.thethings.network/console/applications/gisa-parking/integrations/mqtt

- Installation und Wartung der Sensoren - hier insbesondere der Tausch von Batterien, sowie eine regelmäßige Reinigung z.B. bei Regensensorik aber ggf. auch ganze Geräte
- Kosten für Hardware, z.B. Gateways, Sensoren und Batterien
- Kommunikationskosten der Gateway-Backendanbindung
- Kosten für die Nutzung von Gatewaystandorten (in fremden Objekten)

5.14.2 IoT-Inventar

Um die Wartung der LoRaWAN-Hardware (Gateways und Sensoren) zu unterstützen, ist es wichtig, ein aktuelles IoT-Inventar zu pflegen. In diesem Kapitel unterbreiten wir einen Vorschlag, zu den notwendigen Attributen, die zu jedem Gerät festgehalten werden sollten.
Die UDSP verfügt aktuell noch nicht über eine IoT-Inventar-Funktion. Im Rahmen des ganzheitlichen IoT-Managements verschiedener Quellen und Technologien gibt es zwei Möglichkeiten, dieses Inventar zu pflegen:

1. Die zentrale Datenhaltung des Inventars (z.B. in GitLab, einer Open-Source Asset-Management-Software wie SnipeIT (https://snipeitapp.com/), Thingsboard, OpenRemote oder einem anderen System), das Parallelbearbeitungen handhaben kann.
2. Die dezentrale Datenhaltung in den jeweiligen IoT-Plattformen. TTI/TTN und NIOTA unterstützen die Speicherung von Attributen zu Geräten. Fotos werden nicht unterstützt, sind aber durchaus hilfreich.

Wichtig: die Erfassung dieser Daten sollte möglichst früh – das heißt vor oder während der physischen Installation - passieren. Diese Daten im Nachgang zu erfassen ist unverhältnismäßig aufwändiger.

- Type
- Hersteller
- Ablageort der Spezifikationen (Möglichst in eigener Verwaltung)
  - Payload-Spezifikationen
  - Sensor-Spezifikation
  - Zertifikate (CE, …)
  - EUI's und Keys in einem abgesicherten Datenspeicher
- Firmware-Version
- Aufstellort
  - Foto
  - GPS-Koordinaten
  - Höhe über n.N.
  - Höhe über Bodenoberfläche
  - Adresse
  - Stockwerk
  - Raum

    - Datum Inbetriebnahme
    - Datum letzter Batterie-Tausch
    - Sonstiges
- Type der Batterie (z.B. 3xAAA, oder Netzversorgung)
- Anzahl der Batterien
- Ablaufdatum (Bei einigen Sensoren optional)
- aktiviertes Join-Verfahren
- aktive Parametrierung
- Bei Gateways
    - Art der Internetverbindung (Mobilfunk, Netzwerk, WLAN)
    - Betreiber der Internetverbindung
    - Kontaktdaten zum Betreiber
    - Zugangsdaten
    - Art der Stromversorgung: Netzteil, PoE, Andere

5.14.3 Einkaufs-Leitlinie für LoRaWAN-Hardware mit Dos und Don'ts

Die Wahl der richtigen Hardware ist entscheidend für den erfolgreichen Betrieb von IoT-Anwendungen. Es gibt eine Vielzahl von Faktoren zu berücksichtigen, um sicherzustellen, dass die Hardware den Anforderungen der Anwendung entspricht und zuverlässig funktioniert.

Im Folgenden werden einige wichtige Aspekte aufgeführt, die beim Einkauf von LoRaWAN-Hardware berücksichtigt werden sollten. Diese Leitlinien können dabei helfen, Risiken zu minimieren. Zudem geben wir Hinweise bezüglich der regulatorischen Vorgaben und den Anforderungen der LoRa-Alliance. Final sind die Gerätehersteller in der Pflicht, konforme Hardware zu bauen und zu vertreiben.

5.14.3.1 Externe Systeme

Die folgenden Best Practices beziehen sich auf externe Systeme (Schnittstellen anderer IoT-Plattformen, Rest-Schnittstellen anderer Systeme) im Allgemeinen und nicht ausschließlich auf IoT-Geräte.

Verfügbarkeit mindestens einer externen Schnittstelle: Es ist wichtig, dass jedes System oder jede Hardware mindestens eine externe Schnittstelle hat, um Interaktionen mit anderen Systemen oder Hardware zu ermöglichen.

Die Schnittstelle sollte die Interaktionen unterstützen, die für den geplanten Anwendungsfall erforderlich sind.

Die Daten sollten direkt über eine Schnittstelle abrufbar sein und nicht (nur) indirekt über die Cloud des Anbieters, dies ist **Bad** Practice - aber leider häufiger anzutreffen.

Die Schnittstellen sollten – wenn möglich – einem Standard oder einer Norm entsprechen: Indem Standards oder Normen verwendet werden, kann die Integrierbarkeit der Schnittstelle verbessert werden und die Interoperabilität mit anderen Systemen erhöht werden.

Etablierte Standards sind REST, MQTT, SensorThingsAPI und NGSI. Es kann jedoch auch sinnvoll sein, andere Standards oder Normen zu berücksichtigen, abhängig von den Anforderungen und Gegebenheiten des Anwendungsfalls. Die Schnittstelle sollte aktuelle Daten liefern.

Öffentliche Verfügbarkeit der Schnittstellen-Dokumentation: Auf keinen Fall sollte die Schnittstellen-Dokumentation unter Non-Disclosure-Agreement, kurz NDA, liegen. Wenn die Schnittstellen-Dokumentation unter NDA liegt, kann es schwierig sein, Integrationskomponenten unter einer Open Source-Lizenz zu veröffentlichen. Die Schnittstellen-Dokumentation sollte öffentlich zugänglich sein, um Integrationen zu erleichtern und die Nutzbarkeit zu verbessern.

5.14.3.2 IoT-Geräte

Im Folgenden einige Best-Practices zum Einkauf/Auswahl von IoT-Geräten:

- Keine Fremd-Cloud: Um die digitale Souveränität zu unterstützen und unnötige Abhängigkeiten zu vermeiden, sollten IoT-Geräte – wenn möglich – nicht von proprietären Fremd-Cloud-Lösungen abhängen.
- Verfügbarkeit: Durch die Standardisierung von Schnittstellen sollte sichergestellt werden, dass IoT-Geräte von mehreren Herstellern angeboten werden, um eine breite Verfügbarkeit zu gewährleisten.
- Push vs. Pull vs. Push-Pull: Die Wahl zwischen Push, Pull oder Push-Pull-Kommunikation hängt vom Use Case ab. Bei der Push-Kommunikation sendet der Sender aktiv Informationen an den Empfänger. Der Empfänger muss nicht nach den Informationen suchen, sie werden ihm direkt zugestellt.

  Vorteile Push:

  - Sofortige Lieferung von Informationen
  - Der Empfänger muss nicht aktiv nach Informationen suchen

  Nachteile Push:

  - Kann als aufdringlich empfunden werden, wenn der Empfänger die Informationen nicht erwartet oder benötigt
  - Es kann zu Informationsüberflutung führen, wenn zu viele Informationen gepusht werden

  Bei der Pull-Kommunikation ruft der Empfänger aktiv die benötigten Informationen ab. Der Empfänger sucht und zieht die Informationen, anstatt dass sie ihm zugestellt werden.

  Vorteile Pull:

  - Der Empfänger hat die Kontrolle über die Informationen, die er erhält
  - Vermeidet Informationsüberflutung, da nur benötigte Informationen abgerufen werden.

Nachteile Pull:

- Erfordert aktive Bemühungen des Empfängers, um Informationen zu erhalten
- Es kann zu Verzögerungen in der Informationsübermittlung führen, wenn der Empfänger nicht weiß, dass neue Informationen verfügbar sind

Die Wahl zwischen Push-, Pull- oder einer Kombination aus beiden (Push-Pull-Kommunikation) hängt vom spezifischen Anwendungsfall ab und den Möglichkeiten des anzuschließenden Systems ab. Es ist wichtig, die Vor- und Nachteile jeder Methode sorgfältig zu berücksichtigen, um die effektivste Kommunikationsstrategie zu wählen.

5.14.4 Vorgehensmodell IoT-Use-Case / Integration

Das Vorgehen bei neuen IoT-Use-Cases und deren Integration in die UDSP wird nachfolgend kurz skizziert:

1. Als Erstes wird der geplante Use-Case definiert. Dies dient als Input für die Integrationsplanung.
2. Anschließend wird eine grobe Skizze der Integration erstellt, die die wichtigsten Elemente und Interaktionen (z.B. externe Komponenten, genutzte Schnittstellen und Komponenten der Plattform die involviert sind) darstellt.
3. Im nächsten Schritt wird eine Checkliste erstellt, die sicherstellt, dass die Integration bestimmte Kriterien erfüllt. Dazu gehören die nachfolgenden Punkte 4 und 5.
4. Eine offene Schnittstelle, um die Integrierbarkeit zu unterstützen.
5. Optionen evaluieren:
   - Keine Abhängigkeit von Fremd-Cloud-Lösungen, um die digitale Souveränität zu unterstützen
   - Die Wahl der geeigneten Kommunikationsart (Push, Pull oder Push-Pull) je nach Use-Case
   - Die Verwendung eines der genannten Standards (MQTT, REST oder NGSI) zum Datenaustausch, um die Interoperabilität zu verbessern.
6. Das Plattformteam wird in das Projekt einbezogen, um die Integration gemeinsam zu skizzieren und dabei die Werte der Daten-Souveränität im Auge zu behalten.
7. Wenn die Integration alle Punkte der Checkliste erfüllt, wird eine Fein-Integrations-Skizze erstellt, die weitere Details und Anforderungen enthält.

5.14.5 Nachhaltigkeit: Wirtschaftlichkeit

In diesem Abschnitt werden wir die Aspekte der Nachhaltigkeit in Bezug auf IoT-Sensoren diskutieren. Es ist entscheidend, die langfristigen Betriebskosten und die Flexibilität der Sensoren zu berücksichtigen, um eine nachhaltige und wirtschaftlich sinnvolle Lösung zu gewährleisten.

Kosten für den Dauerbetrieb: Es ist wichtig, nicht nur den Gerätepreis, sondern auch die Kosten für den Dauerbetrieb zu berücksichtigen. Dazu gehören z.B. die Kosten für Batterien und die Betriebsfestigkeit der Geräte an sich. Abhängig vom Anwendungsfall kann es hier große Unterschiede geben: Sensoren in Abfallbehältern oder in Fließgewässern sind wartungsintensiver als z.B. Indoor-Sensoren.

Möglichkeit eines Batteriewechsels: Es ist wichtig, dass Batterien von IoT-Sensoren leicht ausgetauscht werden können, um die Nachhaltigkeit zu verbessern.

Betriebsdauer eines Sensors mit einer Batterie: Es ist wichtig, dass der Sensor eine lange Betriebsdauer hat, um die Häufigkeit von Batteriewechseln zu reduzieren.

Kosten der Batterien: Es ist sinnvoll, die Kosten von Standardbatterien und Spezialbatterien zu vergleichen, um die wirtschaftlichste Wahl zu treffen.

Möglichkeit des Netzwerkwechsels: Es ist wichtig, dass die IoT-Sensoren flexibel sind und leicht zwischen verschiedenen Netzwerken gewechselt werden können, um die Nachhaltigkeit zu verbessern und die Abhängigkeit von einem bestimmten Netzwerk zu verringern.

Sensoren, welche für den Wechsel des Netzwerkes vor Ort-Tätigkeiten, ggf. noch mit Softwaretools des Geräteherstellers, erfordern, stellen in diesem Kontext gerade bei größeren Stückzahlen eine Herausforderung dar.

5.14.6 LoRa-Alliance-Konformität

Im Folgenden werden einige wichtige Aspekte zur LoRa-Alliance-Konformität aufgeführt, die beim Erwerb von IoT-Sensoren berücksichtigt werden sollten.

Einhaltung regulatorischer Vorgaben: Die Einhaltung von regulatorischen Vorgaben, wie z.B. Duty Cycle, Sendeleistung und Frequenzbereiche, ist zwingend erforderlich, um Strafen und Sanktionen durch die Bundesnetzagentur (BNetzA) zu vermeiden. Kommerzielle Sensoren sind so vorkonfiguriert, dass sie die gesetzlichen Vorschriften einhalten. Wenn man aber selbst umkonfiguriert (z.B. Verkürzung des Sendezyklus) muss die Airtime überwacht werden. Dies wird daher nicht empfohlen.

Konformität zu den Vorgaben der LoRa-Alliance: Die Konformität zu den Vorgaben der LoRa-Alliance ist wichtig, um das Risiko von Inkompatibilitäten mit neuen Versionen des LoRa-Netzwerkservers zu minimieren.

Vorgaben des Netzbetreibers: Es ist wichtig, die Vorgaben des Netzbetreibers zu berücksichtigen, um mögliche Einschränkungen oder Sanktionen zu vermeiden.

Exkurs: Duty Cycle
Der Duty Cycle beschreibt das Verhältnis der Gesamtzeit, die für das Senden von Daten verwendet

wird, zu der Gesamtzeit, die für das Senden und Empfangen von Daten verwendet wird. In LoRaWAN gibt es festgelegte Duty Cycle-Beschränkungen, die in verschiedenen Frequenzbereichen gelten. Diese Beschränkungen wurden eingeführt, um die Nutzung von Funkfrequenzen zu reglementieren und zu vermeiden, dass ein einzelnes Gerät oder eine einzelne Anwendung die Funkfrequenzen übermäßig beansprucht. Es ist wichtig, dass IoT-Geräte, die in einem LoRaWAN-Netzwerk betrieben werden, den Duty Cycle-Beschränkungen entsprechen, um die Zuverlässigkeit und Leistung des Netzwerks zu gewährleisten.

5.14.7 Einsatz anderer Netzwerke

Jedes Netzwerk erlaubt die Vergabe eigener Geräteschlüssel. Aus diesem Grund kann eine Eindeutigkeit der Geräte-IDs (dev_eui) nur innerhalb eines Netzwerkes sichergestellt werden. Im Worst Case können aus zwei unterschiedlichen Netzwerken Daten von Sensoren mit der gleichen Geräte-ID geliefert werden.

Aus diesem Grund sollte die Geräte-ID in der Datenplattform neben der dev_eui auch eine eindeutige ID zur Kennzeichnung des Netzwerkes (Net_ID) sowie eine Angabe zur Netzwerkart beinhalten. Nur so kann sichergestellt werden, dass man einen wirklich eindeutigen Schlüssel nutzt.

Beispiel: lora.00013.8C83FC05005A9F74

lora - Angabe, dass es sich um ein LoRaWAN-Netzwerk handelt.

00013 - Net-ID für TTN/TTI, innerhalb der LoRa-Netzwerke soll die Net_ID eindeutig sein

8C83FC05005A9F74 - DEV_EUI des Sensors, innerhalb von TTN/TTI eindeutig

Das eigene Netzwerk kann funktional am besten eingebunden werden, schließlich stehen dem Betreiber alle möglichen Optionen uneingeschränkt zur Verfügung. Die Anbindung fremder Netzwerke dagegen wird lediglich in dem Rahmen möglich sein, welchen der Betreiber des Netzwerkes bereitstellt.

Verschiedene Funktionen liegen direkt beim Betreiber. Das hat zur Folge, dass bestimmte Funktionen nicht implementiert werden müssen, da diese Aufgaben nicht in der eigenen Zuständigkeit liegen. Das kann sowohl das Onboarding neuer Geräte (Nodes und/ oder Gateways), deren Monitoring und Anpassungen im Kommunikationsnetzwerk betreffen.

6 Prozesse und Datenflüsse

Im folgenden Kapitel wird näher auf Prozesse und Datenflüsse eingegangen. Diese ergeben sich aus den jeweiligen Anwendungsfällen sowie der Art und Weise der Implementierung. Im aktuellen Kontext der Datenplattform 5fSWF werden die Datenmodelle durch den NGSI-Standard definiert, können jedoch ergänzt werden.

6.1 Schnittstellen zum Datentransfer LNS-Plattform Integration 5fSWF

Die Datenplattform 5fSWF unterstützt für die Einbindung externer Daten nativ das NGSI-Protokoll mit den darauf abgebildeten NGSI-Smart Data Models. Stand 10/2023 wird NGSIv2 verwendet, der Wechsel auf das modernere und performantere NGSI-LD befindet sich in Planung.

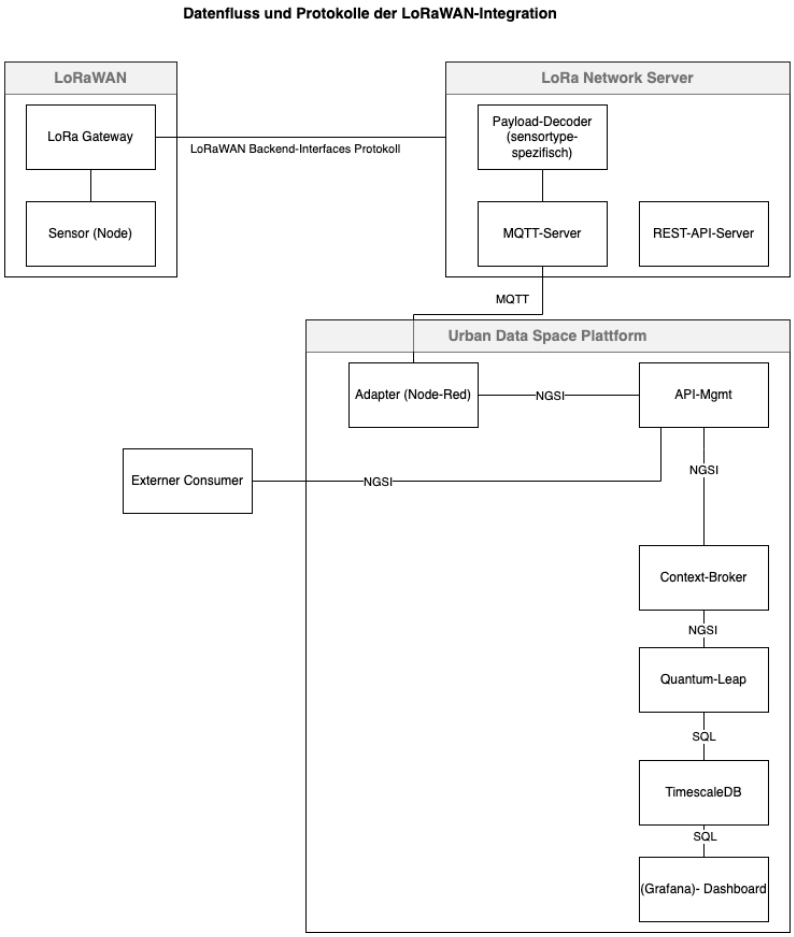
Auf Basis von NGSI werden alle Daten in der Datenplattform erfasst - aus diesem Grund werden alle externen Daten entweder von der Quelle oder von einem Plattformadapter in NGSI gewandelt und in diesem Format weiterverarbeitet.

Im anschließenden Kapitel geht es um die Smart Data Models, die auf NGSI abgebildet werden. Diese stellen die semantische Interpretation der Daten dar.

Weiter geht es um die Alternativen zum Transport über https, die mit der Plattform ebenfalls genutzt werden können.

Abschließend beschäftigt sich die Zusammenfassung des Kapitels mit der konkreten Übertragung der Standards auf das Anwendungsfeld der LNS-Integration.

Die nachfolgende Grafik zeigt den Datenfluss und die verwendeten Protokolle im / zwischen LoRaWAN, LNS und der Datenplattform.



Erläuterungen zur Grafik: alternativ zu MQTT kann die Datenübertragung in die Plattform auch z.B. über REST-Schnittstellen erfolgen. Quantum Leap ist der FIWARE Generic Enabler, der die Datenhistorie behandelt. Mit dem Wechsel von NGSIv2 auf NGSI-LD ist Quantum Leap optional, da der Context Broker dann selber die Historie behandelt. Da die Kommunikation meist bidirektional erfolgt, wurden keine Richtungspfeile in der Grafik eingetragen.

6.1.1 NGSI

FIWARE-Systeme, wie das der 5fSWF nutzen die standardisierte Schnittstelle NGSI.

NGSI ist die Schnittstellen für die Kontextinformationenverwaltung: Dies ermöglicht es Anwendungen, Informationen über ihren aktuellen Kontext, z. B. aktuelle Daten von Sensoren oder anderen Informationsquellen, zu entdecken und zu aktualisieren.

NGSI verwendet eine einfache und leicht verständliche Datenstruktur (auf Basis von JSON) für Kontextinformationen, wodurch es einfacher wird, mit diesen Daten zu arbeiten.

Standardisierte APIs: Durch die Verwendung von NGSI können Datenkonsumente und Datenproduzenten auf standardisierte APIs zurückgreifen, um Kontextdaten aus verschiedenen Quellen in ihre Anwendungen zu integrieren.

6.1.2 Smart Data Models

Smart Data Models stellen in einer großen Nutzergemeinschaft abgestimmte und akzeptierte Informationsmodelle zur Verfügung. Der De-facto-Standard NGSI stellt interoperable und replizierbare (übertragbare) Modelle in verschiedenen Domänen (Smart Cities, Smart Agrifood, Smart Utilities, Smart Industry usw.) bereit.

Diese Datenmodelle sind ein wesentliches Element der gemeinsamen technischen Grundlage, die für standardbasierte offene Innovation und Beschaffung benötigt wird. Datenmodelle spielen eine entscheidende Rolle, da sie die harmonisierten Darstellungsformate und die Semantik definieren, die von Anwendungen sowohl zum Konsumieren als auch zum Veröffentlichen von Daten verwendet werden.

Unter smartdatamodels.org wird ein gemeinsames Kooperationsprogramm betrieben, um die Einführung einer Referenzarchitektur und kompatibler gemeinsamer Datenmodelle zu fördern. Alle Datenmodelle sind öffentliche und lizenzgebührenfreie Spezifikationen.

Diese auf GitHub veröffentlichten Spezifikationen enthalten JSON-Schemata und Dokumentationen zu Smart Data Models für verschiedene Domänen. Für jeden Bereich gibt es ein Repository, das als Untermodule den Link zu den Themen enthält, die alle damit verbundenen Datenmodelle enthalten. Darüber hinaus gibt es einige Themen und Modelle als domänenübergreifende Inhalte.

Die folgenden allgemeinen Grundsätze gelten für das Design und die spätere Veröffentlichung von Smart Data Models. Die bereits veröffentlichten Modelle können frei genutzt werden. Eigene Modelle können der Community zur Beurteilung und Veröffentlichung zur Verfügung gestellt werden.

- **Driven-by-implementation approach**: Die Spezifikationen gelten als stabil, sobald sie von einer ausreichenden Anzahl von Endnutzerorganisationen (z. B. Städten) in der Praxis validiert wurden.
- **Open-closed**. Änderungen an bereits genehmigten Spezifikationen sind nicht erlaubt. Stattdessen sollen neue Versionen Attribute verwerfen, neue Attribute hinzufügen, Aufzählungen erweitern, etc.
- **Open contribution**. Beiträge stehen jedem offen (nicht nur Mitgliedern), während die endgültige Entscheidungsfindung den Administratoren der Domänen und Subjects obliegt.

6.1.3 Transportprotokolle

Der NGSI-Standard sieht die Kommunikation über http - bzw. besser https - als Transport-Protokoll vor. Der von der Datenplattform bereitgestellte Context Broker folgt diesem Standard.

Trotzdem können auch Alternativen bei Bedarf und Sinnhaftigkeit der Architektur genutzt werden.

Grundsätzlich kann man davon ausgehen, dass jede Kommunikation, die mit NGSI über http(s) abgebildet werden kann, auch auf jedem anderen Transportprotokoll mit Nachrichtenorientierung abgebildet werden kann. D.h. jeder Message Broker kann dem NGSI-Protokoll vorgeschaltet werden. Ein Message Broker ist eine Software, die Nachrichten zwischen verschiedenen Anwendungen austauscht, unabhängig von deren Programmiersprachen oder Protokollen. Er dient als Vermittler und Puffer für die Kommunikation.

Message Broker für den Nachrichten Transport können unter anderem Folgende sein:

- MQTT-Protokoll
    - Mosquitto - https://mosquitto.org/
    - HiveMQ - https://www.hivemq.com/
    - EMQX - https://www.emqx.io/
- AMQP
    - RabbitMQ - https://www.rabbitmq.com/
- Kafka – https: //kafka.apache.org/

Diese können je nach Anwendungsfall und dem am besten geeigneten Protokoll der NGSI-Schnittstelle vorgeschaltet werden. Um dies zu erreichen kann ein Adapter implementiert werden, beispielsweise MQTT zu NGSI oder Kafka zu NGSI. Ein solcher Adapter übernimmt den Payload des Message Brokers und überträgt ihn per http Transport weiter.

Dieses Integrationsmuster funktioniert vor allem für die Datenübertragung sehr gut. In diesem Fall ist meistens keine synchrone Kommunikation erforderlich. Für synchrone Kommunikation kann ein solcher Adapter ebenfalls erweitert werden, dies sollte aber nur in Ausnahmefällen genutzt werden, da die Integration mit Rückbestätigung von Aktionen sehr aufwändig wird.

6.1.4 Integration LNS-Datenplattform

Die Datenplattform “denkt” im Kern im NGSI-LD Protokoll und in Smart Data Models. Das bedeutet, dass alle Daten in dieses Format übertragen werden müssen.

Für diese Übertragung gibt es zwei Möglichkeiten:

1. Die Quelle unterstützt nativ NGSI-LD und kann an die Datenplattform direkt NGSI-LD Daten übertragen.
2. Die Quelle unterstützt ein aus der Sicht der Datenplattform proprietäres Format. In diesem Fall wird ein Adapter benötigt, der neben der Protokollumsetzung auf NGSI vor allem auch für die Semantik der korrekten Übertragung in ein oder mehrere Smart Data Models zuständig ist.

Für die LNS-Integration kommt hauptsächlich die Variante mit Konnektor in Frage, da es keinen LNS mit nativer NGSI-Unterstützung am Markt gibt.

Der LNS stellt aus Sicht der Datenplattform eine generalisierte Datenquelle dar. In dieser Datenquelle gibt es verschiedene Sensortypen und Sensoren. Abhängig vom Sensortyp wird das entsprechende Smart Data Model gewählt - siehe Kapitel 6.2.

Diese Logik lässt sich mit einem Adapter generisch abbilden. Auf Basis des Low-Code Systems Node-RED, das zur Implementierung von Adaptern in der Plattform genutzt werden kann, lässt sich eine allgemeine Logik leicht verständlich implementieren und bei Bedarf durch den Nutzer pflegen. Details dazu folgen im Kapitel 6.

Mit diesem grundsätzlichen Ansatz lassen sich Sensoren weitestgehend automatisiert in die Plattform aufnehmen. Die vorhandenen Plattformkomponenten fügen weitere Sensoren nach der ersten Verbindung des LNS einfach hinzu. Der Node-RED Flow muss um das Mapping Sensor Typ auf Smart Data Model erweitert werden. Zusätzlich muss bestimmt werden, in welchem Dataspace (hier vorstellbar wie ein Ordner für Daten zu den selektiv Lese- und oder Schreibrechte gegeben werden können) die Daten abgelegt werden sollen.

Das Mapping im Detail wird im folgenden Abschnitt beschrieben. Weitere Automatisierungen können bei Bedarf durch individuelle Programmierung einer übergreifenden Administrationsumgebung erreicht werden. Ohne diese sind die Konfigurationsschritte manuell in mehreren Komponenten nacheinander durchzuführen.

6.2 Auswahl und Anwendung von Smart Data Models

Smart Data Models dienen zur übertragbaren und standardisierten Modellierung von Datenmodellen in Datenplattformen. Technisch setzen die Smart Data Models NGSI als Protokoll voraus. Mehr technische Anforderungen existieren für den Einsatz nicht.

Über smartdatamodels.org werden die Modelle bereitgestellt und dokumentiert. Die Anzahl der bereitgestellten Modelle wächst kontinuierlich, sodass für viele Sensoren und allgemeinere Datenquellen bereits ein Modell existiert und genutzt werden kann.

6.2.1 Mapping der realen Welt auf Smart Data Models

Smart Data Models fassen eine Reihe von Werten / Attributen zu Modellen zusammen, die die physische Repräsentanz eines Geräts modellieren sollen. Das zu verstehen, geht am einfachsten an einem Beispiel:

Eine Ladestation für Elektrofahrzeuge wird im Smart Data Model wie folgt modelliert:


```
{
    "id": "urn:ngsi-ld:EVChargingStation:ValladolI+D_Covaresa",
    "type": "EVChargingStation",
    "address": {
        "type": "Property",
        "value": {
            "addressCountry": "Espa\u00f1a",
            "addressLocality": "Valladolid",
            "streetAddress": "Paseo de Zorrilla, 191"
        }
    },
    "allowedVehicleType": {
        "type": "Property",
        "value": [
            "car"
        ]
    },
    "capacity": {
        "type": "Property",
        "value": 2
    },
    "chargeType": {
        "type": "Property",
        "value": [
            "free"
        ]
    },
    "contactPoint": {
        "type": "Property",
        "value": {
            "email": "vehiculoelectrico@ava.es"
        }
    },
    "location": {
        "type": "GeoProperty",
        "value": {
            "coordinates": [
                -4.747901,
                41.618265
            ],
            "type": "Point"
        }
```

```
    },
    "name": {
        "type": "Property",
        "value": "Agencia de Innovaci\u00f3n"
    },
    "socketType": {
        "type": "Property",
        "value": [
            "Wall_Euro"
        ]
    },
    "source": {
        "type": "Property",
        "value": "https://openchargemap.org/"
    },
    "powerConsumption": {
        "type": "Property",
        "value": 10.0
    },
    "chargingUnitId": {
        "type": "string",
        "value": "PZEV01-DeltaBharatAC001-SCTLGandhiPark001"
    },
    "stationName": {
        "type": "Property",
        "value": "SmartCityTvmGandhiParkOne"
    },
    "amountCollected": {
        "type": "Property",
        "value": 0.08
    },
    "vehicleType": {
        "type": "Property",
        "value": "e-motorcycle"
    },
    "observationDateTime": {
        "type": "Property",
        "value": {
            "@type": "date-time",
            "@value": "2022-06-28T20:27:29+05:30"
        }
    },
    "@context": [
        "https://smart-data-models.github.io/dataModel.Transportation/context.jsonl
d",
        "https://raw.githubusercontent.com/smart-data-models/dataModel.Transportati
on/master/context.jsonld"
    ]
}
```

Das gesamte Modell EVChargingStation repräsentiert also eine vollständige Ladestation. Die `ID` dient zur eindeutigen Identifikation im Kontext Management der Datenplattform, der `type` referenziert das Smart Data Model, was zur Anwendung kommt.

Alle folgenden Werte sind Eigenschaften, die die Ladestation beschreiben. In diesen Eigenschaften gibt es statische Werte (`capacity` oder `socketType`). Zusätzlich gibt es dynamische Werte (`amountCollected`).

Allgemein kann man das so formulieren, dass Sensoren mehrere Sensorwerte inkludieren.

Das generische Mapping erfolgt wie folgt:

1. Ein Sensortyp entspricht einem Smart Data Model Type
2. Ein Sensorwert entspricht Attributen des Smart Data Model Typs
3. Ein echter Sensor entspricht einer Instanz des Smart Data Models mit einer konkreten ID

6.2.2 Mapping Lücken füllen

Im realen Umfeld sind nicht alle Smart Data Models flexibel genug, um alle Varianten von echten Geräten abzubilden. Um diese Lücken trotzdem abzubilden, gibt es zwei Möglichkeiten.

6.2.2.1 Erweiterung der Smart Data Models

Technisch spricht einer Erweiterung des Datenmodells um weitere Attribute nichts entgegen. Gerade Daten, die vor allem auf der eigenen Plattform verarbeitet werden, können so sehr einfach auf bestehenden Smart Data Models aufgebaut werden und flexibel angepasst werden.

Die Erweiterung lässt sich sehr pragmatisch realisieren, indem das Datenmodell direkt im JSON-Format (siehe oben) durch weitere Attribute erweitert wird.

Geht es um eine allgemeingültige Erweiterung, kann diese natürlich auch zur Aufnahme in den Standard übergeben werden.

Beispiel:

Erweiterung der `EVChargingStation` um weitere Attribute wie `validChargingCards` o.ä.

6.2.2.2 Nutzung zusätzlicher Smart Data Models

Geht es um komplexere Erweiterungen, ist es vorteilhaft, eigene Modelle zu definieren und diese über das Linking entsprechend zu verknüpfen. Dieses Datenmodell kann individuell modelliert werden. Vorteilhaft ist es, dies ebenfalls standardisierbar zu tun.

Beispiel:

Erweiterung der `EVChargingStation` um weitere verknüpfte Objekte `EVChargingPoints` o.ä.

Eine ausführliche Anleitung zur Erstellung und auch zur späteren Einreichung zur Standardisierung ist auf der smartdatamodels.org Webseite zu finden.

6.2.3 Mapping auf Plattformseite

Damit Datenquellen (Geräte, Sensoren, etc.) strukturiert auf der Plattformseite verwaltet werden können, gibt es mehrere Möglichkeiten entsprechende Metadaten zu ergänzen.

- **Ableiten von Metadaten aus den LNS-Daten:** In diesem Fall werden auf Grund von Informationen aus dem LNS (Applikation, Metadaten, etc.) die notwendigen Metadaten in der Plattform zugeordnet. D.h. die Plattform übernimmt externe Informationen und fügt keine weiteren hinzu.
- **Setzen der Metadaten auf Plattformseite:** Mit einem Enrollment (Einrichten des Sensors, Provisionieren des Sensors) kann auf der Plattformseite jedem Sensor der notwendige Metadatensatz aus einem lokalen Customizing zugeordnet werden. Dieses Vorgehen hat den Vorteil, dass man sich nicht auf die Metadaten des Netzwerk-Stacks verlassen muss. Gerade bei Community Daten können so uneinheitliche Benennungen korrigiert werden und die Metadaten dem eigenen Qualitätsanspruch entsprechend ergänzt werden.

Wir empfehlen die zweite Variante, damit eine einheitliche Datenqualität sichergestellt werden kann.

Die bestehenden Node-RED Flows (https://gitlab.com/urban-dataspace-platform/use_cases/integration) setzen diese Variante bereits um. Die Logik und mögliche Erweiterungen werden in den folgenden Unterabschnitten beschrieben.

6.2.3.1 Meta-Daten Ergänzung

Im ersten Schritt werden die administrativen Tätigkeiten auf LoRa-Stack Seite und in der Datenplattform entkoppelt durchgeführt. D.h. der Administrator eines Sensors / einer Datenquelle führt folgende Schritte aus:

- Bei Bedarf: Erstellen der passenden Applikation zur Bündelung verschiedener Sensoren zu einem Anwendungsfall
- Registrieren und Zuordnen des Geräts zur Applikation
- Konfiguration der Datenweiterleitung (vorzugsweise per MQTT)

Auf Seite der Datenplattform wird dann folgendes konfiguriert:

- Die Metadaten zur Identifizierung von Sensoren/Geräten werden hinterlegt (ID/Attribute/etc.) - Datensätze außerhalb dieses Filters werden ignoriert
- Das Ziel für die Speicherung der Daten wird hinterlegt:
    a. Dataspace (logisch getrennter Datenraum)
    b. Pfad im Dataspace (zur Abbildung einer Struktur zur Einordnung der Daten)

Mit diesem Basiscustomizing können die Daten in die Plattform integriert werden. Das inhaltliche Mapping ist als Teil des Node-RED Flows abgebildet. D.h. pro Quelle gibt es einen Pfad durch den Flow, der die Überführung in das passende Smart Data Model implementiert.

6.2.3.2 Customizing des Mappings

Eine potenzielle Erweiterung des beschriebenen Flows stellt das Customizing des Datenmappings dar. Customizing meint in diesem Fall, dass das Mappen der Quelldaten auf die entsprechenden Attribute des Zieldatenmodells pfleg- und änderbar in einer Datenbank hinterlegt wird.

So können Datenquellen mit XML, CSV oder JSON-Struktur - LoRaWAN nutzt in der Regel JSON - auf das entsprechende Smart Data Model dynamisch gemappt werden, D.h. die Einstellungen werden für die Datensätze aus der Datenbank gelesen und angewendet. Insbesondere ähnliche Datenquellen (verschiedene Quellen für Luftdaten, Parksensoren, etc.) können so mit einer gemeinsamen Pfadlogik im Flow genutzt werden.

6.2.3.3 Vollständig integrierte Administration

Eine weitere potenzielle Erweiterung ist die vollständig integrierte Administration. Diese Erweiterung macht sich zu Nutze, dass zum Beispiel der gesamte The Things Stack per API steuerbar ist. D.h. alle Schritte, die auch über die Web UI manuell zu tätigen sind, sind auch per API automatisierbar. Diese API kann dazu genutzt werden, dass aus der Datenplattform heraus der gesamte TTI-Tenant automatisiert konfiguriert wird. Der Administrator muss nur noch Sensoren auf der Datenplattform verwalten. Alle notwendigen Einstellungen im The Things Stack werden von der Datenplattform per API-Orchestrierung vorgenommen.

Die Erweiterung vereinfacht insbesondere bei einer hohen Zahl von Sensoren und Datenräumen die Administration, da evtl. Fehler bei der manuellen Ausführung der Schritte in beiden Plattformen ausgeschlossen werden.

Dies ist Stand 09/2023 in der UDSP noch nicht umgesetzt, sondern befindet sich in Planung. Zur Umsetzung kommen die Integration z.B. von SensorThingsBoard, OpenRemote oder snipeIT in Frage.

6.3 Möglichkeiten zum Anlegen von Geräten und Schnittstellen im LNS

Sowohl Gateways als auch Nodes lassen sich mit den entsprechenden Berechtigungen über Schnittstellen von außen auf dem LNS anlegen, verwalten und löschen. Diese Möglichkeiten reichen bis hin zur Vergabe von Contributorenrechten an Dritte, sofern diese Benutzer im LNS bereits angelegt sind.

Im Falle von TTI erfolgt diese Administration über https-Aufrufe / REST. Damit können die Arbeiten im LNS komplett in Oberflächen der UDSP verlagert werden, wo das “eigene” Berechtigungskonzept greift, welches wesentlich detaillierter sein kann als das des LNS. Damit besteht auch die Möglichkeit, für mehrere Organisationen einen gemeinsamen TTI-Tenant zu betreiben und über eine detaillierte Berechtigungsvergabe in der UDSP verschiedene Partner ausschließlich auf “ihren Bereich” des LNS zugreifen zu lassen. Sofern diese Art der gemeinsamen Nutzung eines LNS-Mandanten nicht durch andere Vorgaben verhindert wird, kann dies einen zukünftigen Weg zur Nutzung maximaler Synergien darstellen.

In diesem Fall müssen neben den Administrationsaufgaben für die Geräte auch deren Monitoringinformationen in die UDSP gehoben werden. Dies ist jedoch insbesondere bei einer

heterogenen Kommunikationsumgebung (neben LoRaWAN noch andere Datenquellen wie Mobilfunk, WLAN, etc.) eine dringende Empfehlung.

Die Vorgehensweise ist an dieser Stelle die, dass ein zentraler Dienstleister des Konsortiums den Tenant betreut und für jede Stadt einen User im System und einen API-Key mit den entsprechenden Berechtigungen anlegt. Diese Informationen werden in der UDSP im entsprechenden Bereich hinterlegt und erlaubt den Zugriff auf die Funktionalitäten.

Mittels dieser Aufrufe können sowohl Gateways, Applikationen und Nodes angelegt, geändert und gelöscht werden. Weiterhin können dazu API-Keys vergeben werden, deren Berechtigungen verwaltet werden.

Welche dieser Prozesse dann im Rahmen des Projektes tatsächlich realisiert werden, muss im Projekt auf Basis der Anforderungen, der dort definierten Vorgehensweise (reicht diese Trennung aus, kann ein gemeinsamer Tenant genutzt werden usw.) sowie der dann zu ermittelnden Aufwände definiert werden.

6.4 Einbindung bestehender LoRa-Netze Dritter

Die Einbindung weiterer LNS-Server weiterer Partner sollte vollständig dem gleichen Integrationspattern folgen, wie auch die eigene Integration. Das heißt, der LNS wird über einen Adapter angebunden, der die Daten in das NGSI-Protokoll und entsprechende Datenmodelle umsetzt.

Zur klaren Trennung der Quellen und evtl. notwendiger Login Informationen oder anderer Daten, sollte die Integration pro LNS über eine weitere Instanz des Konnektors erfolgen. Neben der logischen Trennung der Daten können so auch technische Probleme des einen LNS nicht andere beeinflussen.

Zur Speicherung der Daten sollte auf eine klare Rückverfolgbarkeit geachtet werden. Wenn möglich sollten Daten von Partnern in getrennten Datenräumen gespeichert werden. Wenn Daten aus mehreren Quellen im gleichen Datenraum gespeichert werden müssen, dann sollte eine klare Verortung über den Service Pfad erfolgen.

Je sauberer die Daten getrennt abgelegt werden, um so einfacher kann bei Bedarf ein Datenbestand wieder aus der Plattform entfernt werden. Zusätzlich sollte beachtet werden, dass Berechtigungen der Datenplattform sich über alle Komponenten hinweg ebenfalls auf die Datenräume beziehen.

Hier ein Beispiel eines Datensatzes, welcher über ein eigenes Gateway sowie eines via Paket-Broker empfangen wurde. Dies ist im Bereich „rx_metadata“ erkennbar. Darin existieren zwei Objekte, einmal ein Gateway mit der gateway_id „rroe-lorix-home“ im eigenen Netz und zum anderen ein Gateway mit der gateway_id "packetbroker". In dem zweiten Objekt finden sich die Angaben zum empfangenden Gateway dann in dem Knoten "packet_broker"


```
{
  "name": "as.up.data.forward",
  "time": "2022-12-12T07:57:39.067621692Z",
  "identifiers": [
```

```
    {
      "device_ids": {
        "device_id": "iotracker-0d59",
        "application_ids": {
          "application_id": "rroe-iotracker"
        },
        "dev_eui": "70B3D59930000D59",
        "join_eui": "70B3D5993FFFFFB7",
        "dev_addr": "260BA655"
      }
    }
  ],
  "data": {
    "@type": "type.googleapis.com/ttn.lorawan.v3.ApplicationUp",
    "end_device_ids": {
      "device_id": "iotracker-0d59",
      "application_ids": {
        "application_id": "rroe-iotracker"
      },
      "dev_eui": "70B3D59930000D59",
      "join_eui": "70B3D5993FFFFFB7",
      "dev_addr": "260BA655"
    },
    "correlation_ids": [
      "as:up:01GM2NDSBMCWAQ3F6GYTG2RHDZ",
      "gs:conn:01GKXXKWXD35TF7CDQ3VSVGSHT",
      "gs:up:host:01GKXXKWXJ6J6HNMKZJBBJZ4HT",
      "gs:uplink:01GM2NDS537DTWAPXYV40NM6WD",
      "ns:uplink:01GM2NDS54GR6J6BY9YCWPCGZ6",
      "rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01GM2NDS54KAPC7S8079PXTT34",
      "rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01GM2NDSBK18HWWNKT3Y5XNH9A"
    ],
    "received_at": "2022-12-12T07:57:39.060051514Z",
    "uplink_message": {
      "session_key_id": "AYT/ADTBxgFYv+q+NYBYMg==",
      "f_port": 1,
      "f_cnt": 61,
      "frm_payload": "EwBoiAEgACAAQQAoBJ8IXg==",
      "decoded_payload": {
        "batteryLevel": 104,
        "bluetoothInfo": {
          "addSlotInfo": 2,
          "beacons": [
            {
              "major": "049f",
              "minor": "085e",
              "rssi": -53,
              "slot": 0,
              "type": "ibeacon"
            }
```

```
        ],
        "status": "success",
        "statusCode": 0
      },
      "containsGps": false,
      "containsOnboardSensors": true,
      "containsSpecial": false,
      "crc": 0,
      "maxAccelerationHistory": 8.192,
      "maxAccelerationNew": 8.192,
      "sensorContent": {
        "buttonEventInfo": false,
        "containsAccelerometerCurrent": false,
        "containsAccelerometerMax": true,
        "containsAirPressure": false,
        "containsBluetoothData": true,
        "containsExternalSensors": false,
        "containsLight": false,
        "containsManDown": false,
        "containsRelativeHumidity": false,
        "containsTemperature": false,
        "containsWifiPositioningData": false
      },
      "uplinkReasonButton": true,
      "uplinkReasonGpio": false,
      "uplinkReasonMovement": true
    },
    "rx_metadata": [
      {
        "gateway_ids": {
          "gateway_id": "rroe-lorix-home",
          "eui": "FCC23DFFFE0B89C8"
        },
        "time": "2022-12-12T07:57:38.713524103Z",
        "timestamp": 268500380,
        "rssi": -81,
        "channel_rssi": -81,
        "snr": 7.75,
        "location": {
          "latitude": 50.324825272549106,
          "longitude": 9.262429475784304,
          "altitude": 340,
          "source": "SOURCE_REGISTRY"
        },
        "uplink_token": "Ch0KGwoPcnJvZS1sb3JpeC1ob21lEgj8wj3//guJyBCc+4OAARoMCPK9
25wGEICQ7JUDIODS+p7omCQ=",
        "received_at": "2022-12-12T07:57:38.830796528Z"
      },
      {
        "gateway_ids": {
```

```
            "gateway_id": "packetbroker"
          },
          "packet_broker": {
            "message_id": "01GM2NDS5PS34Z5X231XWQ415X",
            "forwarder_net_id": "000013",
            "forwarder_tenant_id": "alphaomega",
            "forwarder_cluster_id": "eu1.cloud.thethings.industries",
            "forwarder_gateway_eui": "58A0CBFFFE801A98",
            "forwarder_gateway_id": "rroe-minihub-801a98",
            "home_network_net_id": "000013",
            "home_network_tenant_id": "ttn",
            "home_network_cluster_id": "eu1.cloud.thethings.network"
          },
          "time": "2022-12-12T07:57:38.804490089Z",
          "rssi": -45,
          "channel_rssi": -45,
          "snr": 11.25,
          "uplink_token": "eyJnIjoiWlhsS2FHSkhZMmxQYVVwQ1RWUkpORkl3VGs1VE1XTnBURU5L
YkdKdFRXbFBhVXBDVFZSSk5GSXdUazVKYVhkcFlWaFphVTlwU2xwWFIyY3lVbTE0UzFOdFNqWlpWbXhxVTF
Sb1drbHBkMmxrUjBadVNXcHZhV1Z0VG14VVZXaDBWV3h3ZG1SWGVIVmlWMUpRWVVoa1VWWXlOVkZSVTBvNU
xscHNTMVEzUTFKcFpGaEZja1kzYWpoUVpDMWpRbEV1YUZwWVoxQktURGxXY1hsUVYyd3hVaTVmZG1sU1Vua
GxhRFZWUkZwaE9UbFVURWQ0YkhobllrdEdSa04zYnkxTGFYRnFTbFJvYjNaUE1sSjZlSE5DT0ZKZmFVNUJU
bE5SWDJkVWRUQlhUWFJsYzFGeGFIRjRSR1pMZHpSVmQzZHlPV053WDNKelEyeHlTV2xxVjAxcFgzRlhZVlJ
qTlRGMmJuaExVbFl6ZG5KR1UydE9lRXhPZG5abE9HdEtPVmxQYVMxVFUyWjNTMkpRVDJWNFJVTjZhWE5CTl
ROTllsTTVZbWxPTUhORWJUYzVWakJ3Tm1GQ1V6VmhlRlJmY1V0M0xpMVNOMkppVDA5Q1NWUnFhbWwwWDNad
lpUbFZWM2M9IiwiYSI6eyJmbmlkIjoiMDAwMDEzIiwiZnRpZCI6ImFscGhhb21lZ2EiLCJmY2lkIjoiZXUx
LmNsb3VkLnRoZXRoaW5ncy5pbmR1c3RyaWVzIn19",
          "received_at": "2022-12-12T07:57:38.869002750Z"
        }
      ],
      "settings": {
        "data_rate": {
          "lora": {
            "bandwidth": 125000,
            "spreading_factor": 10,
            "coding_rate": "4/5"
          }
        },
        "frequency": "867900000",
        "timestamp": 268500380,
        "time": "2022-12-12T07:57:38.713524103Z"
      },
      "received_at": "2022-12-12T07:57:38.852356823Z",
      "confirmed": true,
      "consumed_airtime": "0.411648s",
      "version_ids": {
        "brand_id": "iothings",
        "model_id": "iotracker3",
        "hardware_version": "3",
        "firmware_version": "1.10",
```

```
        "band_id": "EU_863_870"
      },
      "network_ids": {
        "net_id": "000013",
        "tenant_id": "ttn",
        "cluster_id": "eu1",
        "cluster_address": "eu1.cloud.thethings.network"
      }
    }
  },
  "correlation_ids": [
    "as:up:01GM2NDSBMCWAQ3F6GYTG2RHDZ",
    "gs:conn:01GKXXKWXD35TF7CDQ3VSVGSHT",
    "gs:up:host:01GKXXKWXJ6J6HNMKZJBBJZ4HT",
    "gs:uplink:01GM2NDS537DTWAPXYV40NM6WD",
    "ns:uplink:01GM2NDS54GR6J6BY9YCWPCGZ6",
    "rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01GM2NDS54KAPC7S8079PXTT34",
    "rpc:/ttn.lorawan.v3.NsAs/HandleUplink:01GM2NDSBK18HWWNKT3Y5XNH9A"
  ],
  "origin": "ip-10-100-4-34.eu-west-1.compute.internal",
  "context": {
    "tenant-id": "CgN0dG4="
  },
  "visibility": {
    "rights": [
      "RIGHT_APPLICATION_TRAFFIC_READ"
    ]
  },
  "unique_id": "01GM2NDSBV663FPMRJNN3A5JW7"
}
```


Im Fall eines fremden Tenants innerhalb des TTN/TTI Netzwerks gibt es anstelle der Einbindung des Tenants selbst in die Plattform die Möglichkeit, die Daten anderer Tenants bereits auf TTN/TTI Seite miteinander zu verbinden. Dazu kann die kostenpflichtige Option des Packet Brokers aktiviert werden. Mit Hilfe des Packet Brokers werden Datenpakete eines anderen Tenants dem eigenen Tenant zur Verfügung gestellt.

6.4.1 Details zum Packet Broker

Der Packet Broker bietet in diesem Kontext die Kopplung verschiedener Netze. Dabei wird auf Basis der Net ID das Zielnetz erkannt und sofern dieses Netz via Packet Broker gekoppelt ist, Daten über diesen Weg an das Zielnetz weitergeleitet. Somit kann dieses Zielnetz Daten über die Gateways des eigenen Netzes senden und empfangen. Dies ist eine effektive Methode, um die eigene Netzabdeckung in Gebieten, in denen man selbst keine Gateways betreibt, zu erweitern.

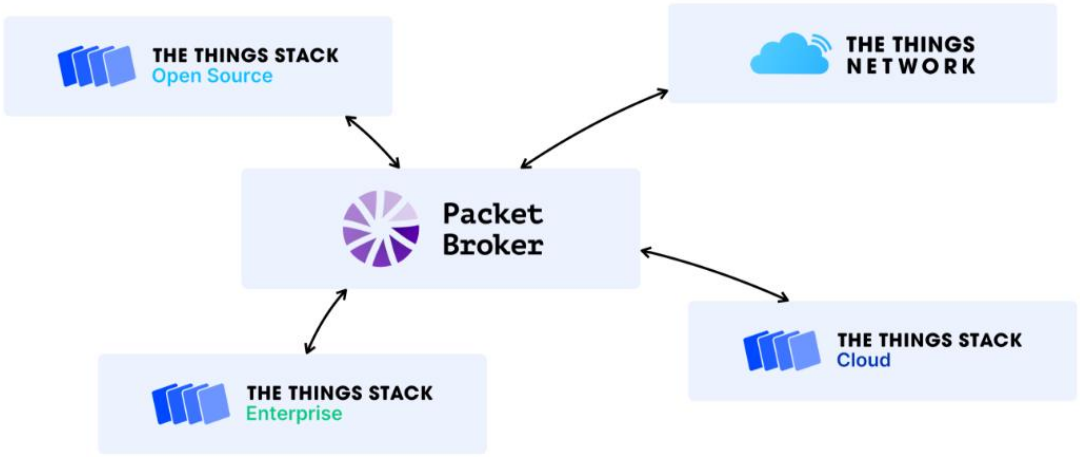


Abbildung 15 - Packet Broker - Vernetzung einzelner Netze[16]

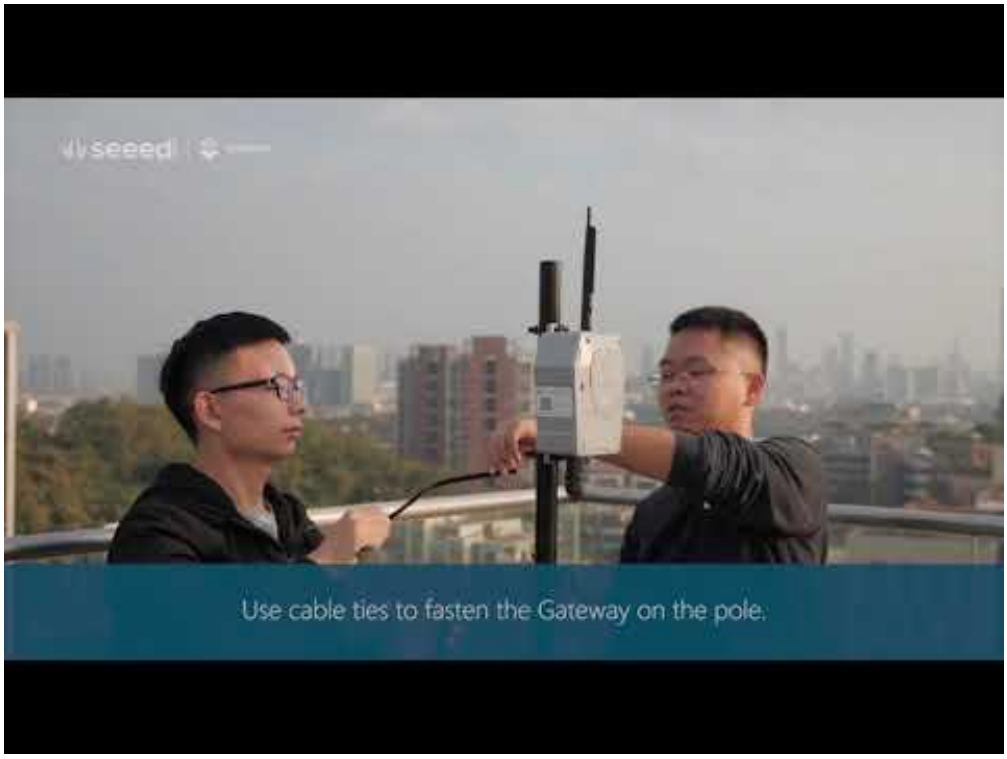


Abbildung 16 - Video zum Packet Brocker (TTI) [17]

[16] *Quelle: https://www.thethingsindustries.com/docs/getting-started/packet-broker/*

[17] *Quelle: https://youtu.be/TQImwfHCraM*

Um dies zu erreichen, wird am eigenen LNS eine Verbindung mit der zentralen Packet Broker Instanz hergestellt. Über diese Verbindung werden dann die Daten mit verbundenen Fremdnetzen ausgetauscht. Dabei kann innerhalb des Packet Brokers genau eingestellt werden, mit welchen Netzen in welcher Richtung Datenpakete getauscht werden sollen. Dabei kann sehr filigran definiert werden, ob und wann welche Informationen zusammen mit den tatsächlichen Nutzdaten an das Zielnetzwerk weitergegeben werden sollen.

Bzgl. der Kopplung TTN/TTI unterstützt The Things Industries das Community-Netzwerk TTN durch die kostenlose Bereitstellung des Packet Brokers für Daten aus TTI-Netzen ins TTN. Benötigt ein Betreiber eines TTI-Netzes Daten aus dem TTN-Netzwerk so kann diese Option kostenpflichtig hinzugebucht werden.

Type	Uplink	Downlink	Description
Join	O	O	Join-request (uplink) and join-accept (downlink) messages
MAC payload	O	O	Data messages with `FPort` `0` (for network layer instructions)
Application payload	O	O	Data messages with `FPort` `1` or higher (for application layer payload)
Signal quality	O		RSSI and SNR information
Localization	O		Gateway locations, timestamps and signal quality

Abbildung 17 - Routing Policies eines Packet Brokers[18]

6.4.2 Empfehlung

Die Einbindung weiterer Sensoren aus anderen TTI/TTN Mandanten lässt sich am einfachsten über die Nutzung des Packet Brokers erreichen.

Die Daten aus dem Packet Broker kommen analog den eigenen Daten über die API der eigenen Applikation an. Lediglich im Element RX-Data ist zu erkennen, dass es sich um Daten aus einem fremden Netzwerk handelt. Handelt es sich um Daten eigener Sensoren, welche lediglich über ein Gateway in einem fremden Netzwerk empfangen wurden, ist eine gesonderte Behandlung nicht erforderlich. Handelt es sich um fremde Sensoren, deren Daten mit in die UDSP einfließen sollen, sollte dies im Rahmen der Sensoridentifikation und Zuweisung zu Datenräumen getrennt werden. Darüber hinaus kann man die Herkunft der Nachricht zurückverfolgen. Falls im Packet Broker freigegeben, sogar bis auf das Gateway genau.

Eine Anbindung zum Packet Broker ist u. a. auch für Chirpstack verfügbar. Ob und in welchem Umfang weitere kommerzielle Anbieter eine solche Anbindung unterstützen (und zu welchen Kosten!) muss im Detail geklärt werden. Technisch ist eine Anbindung an alle Netze, die auf den LoRaWAN-Standard setzen, möglich.

Alternativ kann ein Fremdnetz auch direkt an die UDSP angebunden werden. Dies generiert jedoch neben der zusätzlichen Schnittstelle für die Daten auch dedizierte Aufwände zur

[18] *Quelle: https://www.thethingsindustries.com/docs/getting-started/packet-broker/*

Integration (Monitoring, Gerätemanagement, usw.), ist jedoch bei wenig kooperationsbereiten Netzbetreibern oft die einzig mögliche Lösung.

7 Nutzung der LoRa-Infrastruktur durch die Bürger der Kommunen

Gerade im Kontext LoRaWAN gibt es regional unterschiedlich stark ausgeprägte Communities aber im Durchschnitt finden sich überall Personen vom Level Interessent bis hin zum Nerd. Oft wird die Leistung der Community für produktive Lösungen als gering eingeschätzt, es gab sogar schon Vorträge unter dem Titel “Innovation kann nur aus dem Business entstehen”. Betrachtet man die Projekte, welche aus Communities heraus entstanden sind, ist diese Aussage definitiv falsch.

Zahlreiche Community-Projekte wie sensordaten.info oder das Projekt “Rettet die Schlei” aus Schleswig-Holstein tragen mit Ihrer Arbeit zu wichtigen gesellschaftlichen Themen bei und gingen dabei Themen an, für die der privatwirtschaftliche Bereich kein Interesse zeigte.

Aber wie kann man die Community in seinem Ort/ seiner Region unterstützen? Am einfachsten geht dies mittels TTN/TTI - hier kann der Betreiber der TTI-Gateways grundsätzlich einstellen, das die Netzabdeckung der eigenen Gateways der Community mit zur Verfügung steht. Aber auch Betreiber eigener LNS können die Community unterstützen. Dies kann, sofern eine eigene NET-ID vorhanden ist, via Packet Broker geschehen. Alternativ haben auch die Betreiber von Chirpstack-Servern den Community-Mitgliedern eigene Accounts auf dem LNS bereitgestellt. Dieser Weg ist jedoch aufwändiger und bedarf des Willens des LNS-Betreibers diese Aufwände auf sich zu nehmen.

Wichtig ist jedoch bei der Einbindung von Community-Aktivitäten in eine Smart-City-Plattform die klare Definition der Übergabe der Daten sowie die Absicherung dieses Übergabepunktes.

Communityprojekte werden nicht selten von Einzelpersonen bzw. losen Personengruppen realisiert. Dies bedeutet, im Fehlerfall ist es schwierig einen Verantwortlichen, manchmal auch nur einen Ansprechpartner zu identifizieren. Daher sollten folgende Dinge berücksichtigt werden:

7.1 Datensicherheit

Es sollten klare Datenformate für die Übergabe der Daten in die Plattform definiert werden, deren Einhaltung dann auch überwacht wird. Daten, welche nicht der Vereinbarung entsprechen werden abgewiesen.

7.2 Performance

Es sollte eine max. Frequenz für eingehende/ ausgehende Daten vereinbart werden, diese muss ebenfalls überwacht werden. Indem man eine Schnittstelle für gelegentliche Datentransfers hochfrequent mit Nachrichten bestückt, kann man ein System problemlos in Performanceprobleme treiben.

7.3 Persistierungsdauer

Hier sollte ebenfalls eine, für beide Seiten akzeptable Vereinbarung getroffen werden, welche einen Overflow der internen Datenhaltung der Plattform und / oder hohe Speicherplatzkosten verhindert. Es gibt Anwendungsfälle, welche eine Datenhaltung über mehrere Jahre erfordern, z.B. um später Vorhersagen aus historischen Daten abzuleiten. Dabei ist jedoch zu prüfen, ob dies in der ursprünglichen zeitlichen Auflösung erforderlich ist oder ob eine Verdichtung der Daten stattfinden kann.

7.4 Visualisierung

An dieser Stelle sind die Grundlagen der DSGVO zu berücksichtigen. Beinhaltet der Community-Use-Case DSGVO-relevante Messwerte ist dies immer ein Grund genauer zu prüfen. Der Betreiber der Plattform steht in der Verantwortung. Eventuell sind solche Anwendungsfälle im Citizen Science grundsätzlich auszuschließen und nur nach individueller Abstimmung zu ermöglichen.

7.5 Übernahme der Use-Cases in den festen Funktionsbestand der Plattform

Zahlreiche Communityprojekte haben inzwischen einen Level erreicht, welcher eine Übernahme in die “offiziellen” Anwendungsfälle der Plattform ermöglicht. Man sollte von Beginn an mit der Community besprechen, wann dies zu welchen Bedingungen möglich ist.

8 Entwicklung von Konnektoren in Node-RED

Node-RED ist ein visuelles Tool, mit dem sich IoT-Prototypen erstellen lassen, indem es Onlineservices, APIs und Hardwaregeräte innerhalb eines “Flows” miteinander grafisch verbindet. In diesem Kapitel wird anhand eines Beispiels (CO2-Ampel der Kreisstadt Olpe) der grundsätzliche Aufbau eines solchen Flows beschrieben, der auf andere Anwendungsfälle übertragbar ist.

Ein Flow zur Anbindung externer Daten und deren Speicherung in dem urbanen Datenraum umfasst u.a. 3 Bereiche:

- Datenakquise (und -prüfung)
- Datenkonvertierung (NGSI) und
- Datenablage (Kontext- und historische Daten)

8.1 Datenakquise

Grundsätzlich können Daten aktiv von externen Quellen erfragt werden (i.d.R. erfolgt dies über REST-APIs) oder passiv (z.B. über Abonnements) entgegengenommen werden. Im vorliegenden Beispiel werden Daten aus dem “The Things Industries”-Netzwerk abonniert. Hierfür wird ein MQTT Input-Knoten benutzt, der eine sog. Subscription an einem MQTT Broker vornimmt, welcher die Device-Daten bei Änderungen an den Client verschickt.
Die erhaltenen Daten sind i.d.R. zunächst als Zeichenkette im JSON-Format verfügbar. Vor der weiteren Bearbeitung muss diese Zeichenkette mittels JSON Parser-Knoten in ein JScript-Objekt

umgewandelt werden. Anschließend kann das Objekt in einem Funktions-Knoten auf Vollständigkeit etc. geprüft werden.

Beispiel für einen Funktionsknoten zur Validierung:

```
if (!msg.payload.hasOwnProperty("uplink_message")) {
    node.error("[" + msg.payload.end_device_ids.device_id + "] No uplink_message!\n
Raw message: " + JSON.stringify(msg));
    return null;
} else if (!msg.payload.uplink_message.hasOwnProperty("decoded_payload")) {
    node.error("[" + msg.payload.end_device_ids.device_id + "] No decoded_payload!\
nRaw message: " + JSON.stringify(msg));
    return null;
} else if (!msg.payload.uplink_message.decoded_payload.hasOwnProperty("[04] OpCode"
)) {
    node.error("[" + msg.payload.end_device_ids.device_id + "] No OpCode!\nRaw mess
age: " + JSON.stringify(msg));
    return null;
}

// At this point, we have a decoded_payload-object and an OpCode inside.
let opcode = msg.payload.uplink_message.decoded_payload["[04] OpCode"]

if (opcode === "Measurements") {
    if (!msg.payload.uplink_message.decoded_payload.hasOwnProperty("[09] Measuremen
ts")) {
        node.status({fill:"red",shape:"dot",text:"[" + msg.payload.end_device_ids.d
evice_id + "] No measurements!"});
        node.error("[" + msg.payload.end_device_ids.device_id + "] No measurements!
\nRaw message: " + JSON.stringify(msg));
        return null;
    }
} else {
    // ignore other opcodes (e.g. "Heartbeat")
    return null;
}

return msg;
```

In obigem Beispiel wird die Nachricht sukzessive auf Vollständigkeit geprüft und darüber hinaus nur solche Nachrichten zur weiteren Verarbeitung zugelassen, die auch gemessene Daten enthalten. Wenn eine der Bedingungen nicht erfüllt ist, erfolgt ein Eintrag in das Log des Containers. Dort können die Fehler mit Standard-Verfahren der Container-Plattform ausgelesen und ausgewertet werden. Es kann z.B. festgestellt werden, wenn bestimmte Sensoren wiederholt bzw. dauerhaft falsche oder fehlende Messwerte liefern.
Damit das Payload-Objekt der originalen Nachricht im weiteren Verlauf nicht versehentlich

gelöscht bzw. überschrieben wird, sollte es “gerettet” werden. Hierzu eignet sich ein Change-Knoten.

8.2 Datenkonvertierung

Der urbane Datenraum erwartet die Daten in einem bestimmten Format (siehe Smart Data Models). Mit einem Funktions-Knoten wird zunächst ein JScript-Objekt erzeugt und mit konkreten Werten aus der Uplink-Nachricht gefüllt. An dieser Stelle können weitere Prüfungen erfolgen, um z.B. das gewählte Smart Data Model zu validieren etc.

Beispiel für einen Funktionsknoten zur Erstellung des Smart Data Models:

```
// let's start with an empty dictionary
let ngsi = {};

// fill in the basic informations
ngsi.id = msg.mqtt_payload.end_device_ids.device_id;
ngsi.type = "AirQualityObserved";

// dateObserved
let dateObserved = {};
dateObserved.type = "Date"
dateObserved.value = msg.mqtt_payload.received_at;
ngsi.dateObserved = dateObserved;

// measurements
let measurements = msg.mqtt_payload.uplink_message.decoded_payload["[09] Measuremen
ts"]

// temperature
let temperature = {};
temperature.type = "Number"
temperature.value = measurements["[01] Temperature [°C]"];
ngsi.temperature = temperature;

// relativeHumidity
let relativeHumidity = {};
relativeHumidity.type = "Number"
relativeHumidity.value = measurements["[02] Relative Humidity [%]"];
ngsi.relativeHumidity = relativeHumidity;

// CO2
let CO2 = {};
CO2.type = "Number"
CO2.value = measurements["[05] CO₂ [ppm]"];
ngsi.CO2 = CO2;
```

8.3 Datenablage

Abschließend werden die Kontext-Daten NGSI-konform an den Orion-Context-Broker gesendet. Hierzu wird die REST-API des Context Brokers mit einem HTTP Request-Knoten angesprochen. In einem vorgeschalteten Funktions-Knoten wird die HTTP-Anfrage entsprechend der NGSI-Spezifikation zusammengesetzt.

Beispiel für einen Funktionsknoten zur Erstellung des HTTP-Requests:

```
msg.url = "https://orion-context-broker.de/v2/entities?options=upsert"

msg.headers = {};
msg.headers['Content-Type'] = 'application/json';
msg.headers['Fiware-Service'] = 'Mein-Datenraum';

msg.payload = msg.ngsi_json;
```

Sollen die Daten zusätzlich in den historischen Daten gespeichert werden, ist entsprechend der NGSI-Subscription-API eine Subscription für Quantum Leap vorzunehmen (siehe auch hier). Quantum Leap ist die Komponente in der Gesamtarchitektur, die speziell für die Speicherung und Abfrage von Zeitreihendaten entwickelt wurde. Durch die Subscription werden spezifizierte Daten vom Orion Context Broker an Quantum Leap gesendet und dort gespeichert, was für viele Anwendungsfälle, wie z.B. Dashboards, Trendanalysen oder Vorhersagen, sehr nützlich ist.

Da beim unsachgemäßen Gebrauch der Subscription-API schwer zu entdeckende Seiteneffekte auftreten können, ist hier besondere Vorsicht geboten!

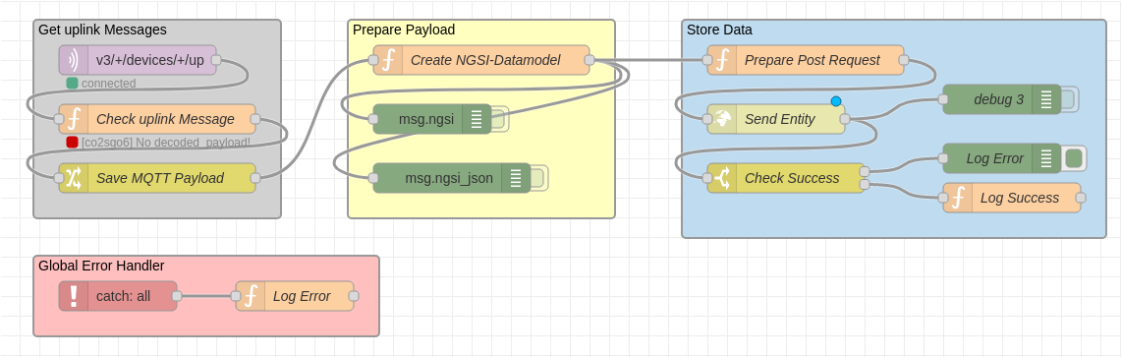


Abbildung 18 - Ein kompletter Node-RED Flow

9 Abkürzungsverzeichnis

Abkürzung	Beschreibung
ADR	Adaptive Data Rate
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
APN	Access Point Name
EMV	Elektromagnetische Verträglichkeit
ETSI	European Telecommunications Standards Institute
FIWARE	Future Internet Core Platforms and Services
IoT	Internet of Things
IRI	Internationalized Ressource Identifier
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation - Linked Data
LNS	LoRa Network Server
LoRaWAN	Long Range Wide Area Network (LoRaWAN)
MIM	Minimal Interoperability Mechanisms
MQTT	MQ Telemetry Transport
MVP	Minimum Viable Product
NB-IoT	Narrowband IoT
NFC	Near Field Communication
NGSI	Next Generation Service Interface
NGSI-LD	Next Generation Service Interface - Linked Data
OASC	Open & Agile Smart Cities
OWL	Web Ontology Language
POC	Proof of Concept
RDF	Ressource Description Framework
RDFS	RDF-Schema
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
TLS	Transport Layer Security
TTI	The Things Industries
TTN	The Things Network
TTS	The Things Stack, der Softwarestack hinter TTN/TTI
UDP	User Datagram Protokoll

Abkürzung	Beschreibung
UDSP	Urban Data Space Platform
XSD	XML Schema Definition
WAN	Wide Area Network

10 Abbildungsverzeichnis