

Staatliche Studienakademie Dresden

– Studienrichtung Informationstechnik –

Entwicklung und prototypische Implementation eines hybriden Lokalisierungssystems auf Basis des Omlox Standards am Beispiel einer definierten Industrieanforderung

Abschlussarbeit zur Erlangung
des staatlichen Abschlusses

Bachelor of Engineering
(B. Eng.)

vorgelegt am 3. August 2023 von

Nicolas Kohnert

Matrikelnummer: 3004318

Ausbildender Praxispartner: ZIGPOS GmbH
Räcknitzhöhe 35a, 01217 Dresden
Begutachtung Praxispartner: M. Sc. Danny Schumann
Begutachtung BA Sachsen: Prof. Dr habil. Andriy Luntovskyy

Autorenreferat

Nicolas Kohnert: *Entwicklung und prototypische Implementation eines hybriden Lokalisierungssystems auf Basis des Omlox Standards am Beispiel einer definierten Industrieanforderung.*

ZIGPOS GmbH. Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Studienrichtung Informationstechnik, Bachelor-Arbeit, 3. August 2023.

52 Seiten

33 Literaturquellen

0 Anhänge

In dieser Arbeit werden die Möglichkeiten des neuen omlox Standards diskutiert und anhand eines Industriebeispiels demonstriert. Vorangehend wird der omlox v2 Industriestandard, bestehend aus Spezifikationen für den omlox Hub und die omlox Core Zone, erläutert und auf geeignete Technologien zur Positionsermittlung eingegangen. Eingeschätzt werden UWB, WiFi, GNSS und Mobilfunk. Aus einem repräsentativen Industriebeispiel werden Anforderungen an das Konzept abgeleitet und klassifiziert. Auf Grundlage der vorangehenden Technologiebetrachtung und den Anforderungen werden je ein Grobkonzept für Mobilfunk, WiFi und GNSS vorgestellt sowie diskutiert. Das GNSS-Konzept wird anschließend weiter spezifiziert und an die existierende Systemarchitektur der omlox Core Zone, einem RTLS der ZIGPOS GmbH, und dem installierten omlox Hub, dem DeepHub der Flowcate GmbH, angepasst. Die prototypische Umsetzung des GNSS-Konzepts enthält das u-blox C099-F9P Application Board mit einem ZED-F9P GNSS-Modul und einem ODIN-W2-Funkmodul sowie eine Middleware zur Datenaufbereitung, geschrieben in Python3. Dabei wird unter Nutzung des MQTT- und des HTTP-Protokolls (letzteres im Kontext einer REST-Schnittstelle) kommuniziert. Die Auswertung der Daten eines abschließenden Systemtest zeigt, dass der entwickelte Prototyp funktionsfähig ist und alle an das Konzept gestellte Anforderungen erfüllt.

Name Nicolas Kohnert

Studiengang Informationstechnologie –
Informationstechnik

Seminargruppe 3IT20 - 2

Auftrag zur Anfertigung der Bachelorthesis

Hiermit erteilen wir Ihnen den Auftrag, folgendes Thema zu bearbeiten:

Entwicklung und prototypische Implementation eines hybriden Lokalisierungssystems auf Basis des Omlox Standards am Beispiel einer definierten Industrieanforderung

1. Gutachter: Dipl.-Ing. (FH) Danny Schumann
2. Gutachter: Prof. Dr. habil. Andriy Luntovskyy

Dresden, 30.04.2023



Leiter des Studienganges Informationstechnologie

Inhaltsverzeichnis

Autorenreferat	a
Abbildungsverzeichnis	e
Tabellenverzeichnis	g
Quellcodeverzeichnis	h
Glossar	i
1 Einleitung	1
2 State of the Art	3
2.1 Softwarearchitekturen und Protokolle	3
2.1.1 Middleware	3
2.1.2 Representational State Transfer	5
2.1.3 Message Queuing Telemetry Transport	6
2.2 omlox	7
2.2.1 omlox v2 Hub	8
2.2.2 omlox v2 Core Zone	11
2.2.3 omlox v2 Complementary Zone	12
2.3 Positionsermittlungsverfahren	13
2.3.1 Received-Signal Strength	13
2.3.2 Time of Flight	13
2.3.3 Time Difference of Arrival	14
2.4 Echtzeitlokalisierungstechnologien	15
2.4.1 Ultra-Wide Band	15

2.4.2	WiFi Positioning System	17
2.4.3	Global Navigation Satellite System	18
2.4.4	Mobilfunk	19
3	Konzeptionierung	21
3.1	Beschreibung des Industriefalls	21
3.2	Anforderungsanalyse	22
3.3	Konzept	23
3.3.1	Grundlegende Systemarchitektur	23
3.3.2	Positionsbestimmung mit Mobilfunk	24
3.3.3	Positionsbestimmung mit WiPS	25
3.3.4	Positionsbestimmung mit GNSS	26
3.3.5	Diskussion und Gesamtkonzept	27
4	Prototypische Implementierung	31
4.1	GNSS Sensor	31
4.2	Middleware	36
5	Evaluation	39
5.1	Test der Implementation	39
5.2	Einschätzung der Testergebnisse	42
5.3	Abgleich mit aufgestellten Anforderungen	46
6	Ergebnisse	48
7	Zusammenfassung und Ausblick	49
7.1	Zusammenfassung	49
7.2	Ausblick	50
7.2.1	Automatisierung	50
7.2.2	Ausfallsicherheit und Skalierung	50
7.2.3	Präzisionsverbesserung	51
	Quellenverzeichnis	53
	Erklärung an Eidesstatt	II

Abbildungen

2.1.1	Beispielhafte Systemarchitektur mit Middleware	4
2.1.2	Kommunikation an einer REST-Schnittstelle	5
2.2.1	Übersicht über die omlox Architektur	8
2.2.2	Grafische Zusammenfassung des Location-Objekts	9
2.2.3	Beispielhafter Zonenaufbau eines omlox Systems	10
2.3.1	Darstellung des SS-TWR-Vorgangs	14
2.4.1	Definition UWB-Signal	16
3.1.1	Schematische Darstellung des vorliegenden industriellen Anwendungsfalls	21
3.3.1	Grobarchitektur einer Mobilfunklösung	25
3.3.2	Grobarchitektur zweier WiPS-Lösung	26
3.3.3	Grobarchitektur einer GNSS-Lösung	27
3.3.4	Heatmap der WLAN-Verfügbarkeitsabschätzung für den industriellen Anwendungsfall	28
3.3.5	Heatmap der WLAN-Verfügbarkeitsabschätzung für den industriellen Anwendungsfall mit simulierten Lastkraftwagen	29
3.3.6	Vereinfachte Systemarchitektur der omlox Core Zone	30
3.3.7	Systemarchitektur der gewählten GNSS-Lösung	30
4.1.1	Foto des C099-F9P Application Boards	32
4.1.2	Konsole von s-center nach Ausführung des Befehls zur Registrierung einer MQTT-Verbindung	33
4.1.3	Zustandsdiagramm für den Wechsel zwischen Command-Modus und Data-Modus des Odin-W2	34
4.1.4	Blockdiagramm der Architektur des C099-F9P Application Boards	35
5.1.1	Systemarchitektur des Testsystems	40

5.1.2	Visualisierung der Positionen in Google Maps	42
5.2.1	Visualisierung der Positionen in Google Maps mit Fokus auf den Übergang der Lokalisierungszonen	43
5.2.2	Visualisierung des Positionsverlaufs zu Beginn und am Ende der Messung	44
5.2.3	Ausschnitt des Positionsverlauf	45
5.2.4	Gegenüberstellung erfasster und approximierter Positionsverlauf	45

Tabellen

2.4.1	Tabellarische Übersicht der Mobilfunkstandards und ihrer Technologie, des genutzten Übertragungstyps und der verfügbaren Datenrate von der ersten bis zur fünften Generation	19
3.2.1	Tabellarische Darstellung der Anforderungen an das Konzept	23
5.2.1	Tabellarische Darstellung der Abweichung zwischen Mess- und approximierten Positionen	46
5.3.1	Tabellarische Darstellung aufgestellter Anforderungen mit Erfüllungsgrad	47

Quellcode

4.1.1	Powershellausgabe bei Flashen des Bootloaders	33
4.1.2	Beispielhafter Auszug der veröffentlichten MQTT-Nachrichten	35
4.2.1	Beispielnachrichten der MQTT-Topic	36
4.2.2	Spaltung der Zeichenkette in die relevanten Informationen	37
4.2.3	Aufbau des Request-Bodies	38
5.1.1	Regelsatz im JSON-Format	40
5.1.2	Auszug aus den aufbereiteten Testdaten	41

Glossar

3GPP	3rd Generation Partnership Project
AES	Advanced Encryption Standard
AoA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
AT	Ad-hoc Tracking
BLE	Bluetooth Low Energy
BMUV	Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz
BPRF	Base Pulse Repetition Frequency
BPS	Battery Powered Satellite
CDMA	Code Division Multiple Access
CLI	Command Line Interface
CORBA	Common Object Request Brokering Architecture
CRS	Coordinate Reference System
CT	Continuous Tracking
DGNSS	Differential Global Navigation Satellite System
DL-TDoA	Downlink-Time Difference of Arrival
DOM	Distributed Object Middleware
EOTD	Enhanced Observed Time Difference
ETSI	European Telecommunications Standards Institute
FBS	Full Blown Satellite
FCC	Federal Communications Commission

FD-MIMO	Full-Dimension Multiple In Multiple Out
FDMA	Frequency Division Multiple Access
FiRa	Fine Ranging
FQAM	Feher's Quadrature Amplitude Modulation
GCP	Ground Control Points
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HPRF	High Pulse Repetition Frequency
HTTP	Hypertext Transfer Protocol
IB	In Band
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IMS	IP Multimedia Subsystem
IoT	Internet of Things
LOS	Line of Sight
LP	Location Provider
LTE	Long Term Evolution
LTE-CAT-M1	Long-Term Evolution for Machines
MEO	Middle Earth Orbit
MIMO	Multiple In Multiple Out
MOM	Message-Oriented Middleware
MPS	Main Powered Satellite
MQTT	Message Queuing Telemetry Transport
MU-MIMO	Multi-User Multiple In Multiple Out
MW	Middleware
NAVIC	Navigation Indian Constellation
NB IoT	Narrow Band Internet of Things
NMEA	National Marine Electronics Association
NTRIP	Network Transportation of RTCM Internet Protocol

OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OoB	Out of Band
OTAU	Over The Air Update
PI	PROFIBUS und PROFINET International
QoS	Quality of Service
REST	Representational State Transfer
RFID	Radio-Frequency Identification
RMC	Recommended minimum sentence C
RMI	Remote Method Invocation
RP	Reference Point
RPC	Remote Procedure Call
RSS	Received-Signal Strength
RSSI	Received-Signal Strength Indicator
RTK	Real Time Kinematic
RTLS	Real-Time Location System
RxT	Receiver Tag
SIP	Session Initiation Protocol
SS-TWR	Single-sided Two Way Ranging
SSL	Secure Sockets Layer
TA	Time-Advance
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TDoA	Time Difference of Arrival
TLS	Transport Layer Security
ToA	Time of Arrival
ToF	Time of Flight
TRxT	Transceiver Tag
TWR	Two-Way Ranging
TxT	Transmitter Tag
UART	Universal Asynchronous Receiver / Transmitter

UL-TDoA Uplink-Time Difference of Arrival

UWB Ultra-Wide Band

VPN Virtual Private Network

WiPS WiFi Positioning System

WLAN Wireless Local Area Network

Kapitel 1

Einleitung

Das „Internet of Things“ (IoT) ist aus dem heutigen Alltag nicht mehr wegzudenken. In vielen Aspekten unseres Lebens verlassen wir uns auf komplexe heterogene Systeme von Sensoren und den aus ihnen gewonnenen Daten. „Smart Home“ und autonomes Fahren sind nur zwei der medienwirksamen Themenbereiche, die sich dem Bereich IoT zuordnen lassen. Abseits der Steuerung und Automatisierung, die durch IoT ermöglicht werden, ist auch die Lokalisierung ein wichtiger Bestandteil dieses Fachbereichs.

Positionsbestimmung, insbesondere die Echtzeitlokalisierung von Objekten und Personen, gewinnt in vielen Anwendungsbereichen an Bedeutung. Während im privaten Bereich das Thema aufgrund von teilweise berechtigten Datenschutzbedenken auf Widerstand stößt, wird in industrieller Umgebung das Potential dieser Technologien zunehmend ausgeschöpft. Eine schnelle und präzise Ortung von Maschinen und Werkstücken sowie die Auswertung der erzeugten Positionsdaten ist für eine Optimierung von Prozessen, und damit eine Effizienzsteigerung, unerlässlich. Jedoch sind die derzeit verfügbaren Echtzeitlokalisierungssysteme, in Englisch Real-Time Location System (RTLS), in der Regel Insellösungen und untereinander nicht kompatibel. Bei dennoch angestrebter Kombination ist aufwändige und komplexe Entwicklungsarbeit notwendig, was dem Effizienzgedanken widerspricht. Die so entstandenen kombinierten Softwarelösungen sind ihrerseits erneut untereinander inkompatibel. PROFIBUS und PROFINET International wollen genau diese Problematik durch den omlox Standard lösen. Dieser Industriestandard soll eine Zusammenführung heterogener Systeme und die Einbindung aller gängigen funkbasierten Ortungsmethoden ermöglichen und somit großen Mehrwert für Industrieunternehmen jeglicher Fachbereiche schaffen.

Die Zusammenführung heterogener Systeme erfordert grundsätzlich eine enge Kooperation und Kommunikation zwischen verschiedenen Technologien und Protokollen. In der omlox Hub Spezifikation soll ein einheitliches Interface zum Austausch und der Aggregation von Daten der verschiedenen Systemen geschaffen werden und somit eine nahtlose Integration von RTLS erleichtern. Durch die Erhaltung der Heterogenität der zugrundeliegenden Funktechnologien der verschiedenen Systemen können technische Limitierungen, beispielsweise Reichweite und Ener-

giebedarf, in anderen System ausgeglichen werden. Systeme zur Indoor-Ortung können mit einer Outdoor-Technologie erweitert werden und so großflächige Positionsermittlung ermöglichen. Die durch den Standard entstehenden Kombinationsmöglichkeiten sind vielfältig und insbesondere für Unternehmen sehr lukrativ. Zudem ist es möglich, die verschiedenen Subsysteme von unterschiedlichen Anbietern zu integrieren, wodurch eine Systemlandschaft unkompliziert erweitert und spezialisiert werden kann.

Im Verlauf dieser Untersuchung wird die Interoperabilität verschiedener Technologien demonstriert. Am Beispiel eines industriellen Use-Cases soll ein hybrides Lokalisierungssystem erstellt werden, welches dem omlox Standard entspricht und eine bestehende Lokalisierungslösung basierend auf Ultra-Wide Band (UWB) durch eine geeignete Lokalisierungstechnologie mit höherer Reichweite ergänzt. Dazu soll geklärt werden, welche Möglichkeiten die omlox Hub Spezifikation zur Einbindung zusätzlicher Technologien bietet und welche Lokalisierungstechnologien für das Industriebeispiel geeignet sind. Zudem werden Probleme in der Umsetzung eines ausgewählten Konzepts identifiziert und gelöst. Das Ergebnis soll realistischen, durch den Industriefall definierten Kriterien entsprechen und der aktuellen Version des omlox Standards unterliegen. Die aktuelle Ausführung des Standards liegt zum Zeitpunkt dieser Untersuchung als Draft vor. Bis zum vollständigen Release können kleine Änderungen nicht ausgeschlossen werden, allerdings ist die Relevanz der Änderungen für diese Betrachtung gering.

Um eine Implementation zu realisieren, werden zuerst die notwendigen Grundlagen erläutert und verschiedene Technologien diskutiert. Der Begriff der Middleware (MW) im Kontext eines Softwaresystems und des IoT sowie die Kommunikationsprotokolle zwischen den einzelnen Architekturbausteinen stellen den Anfang des zweiten Kapitels dar. Die omlox Spezifikationen werden betrachtet und auf elementare Begriffe in diesem Kontext wird eingegangen. Letztendlich werden relevante Positionsermittlungsverfahren und geeignete Funktechnologien zusammengetragen und erklärt.

Zu Beginn des dritten Kapitels wird der zu betrachtende industrielle Anwendungsfall dargestellt. Basierend auf diesem werden Anforderungen abgeleitet und für eine verbesserte Übersichtlichkeit tabellarisch zusammengefasst. Danach wird ein Architekturkonzept erstellt und die für diesen Use-Case geeigneten Technologien ausgewählt. Im vierten Kapitel wird die Umsetzung des Konzepts unter Beachtung der bereits existierenden Lösungen beschrieben. Schwerpunkt liegt dabei auf der Implementation und Einrichtung der Positionsdatenquellen und der verarbeitenden MW.

Um die Qualität der entstandenen Konzeptumsetzung beurteilen zu können, wird im fünften Kapitel ein Testaufbau zur Gewinnung repräsentativer Daten beschrieben und umgesetzt. Das Testergebnis wird anschließend ausgewertet und hinsichtlich der in Kapitel drei aufgestellten Anforderungen beurteilt, um eine Einschätzung der Anforderungserfüllung abgeben zu können. Die Ergebnisse werden im sechsten Kapitel präsentiert und abschließend wird in Kapitel sieben die komplette Ausarbeitung zusammengefasst und ein Ausblick bezüglich weiterführenden Verbesserungen und potentiellen Erweiterungen gegeben.

Kapitel 2

State of the Art

2.1 Softwarearchitekturen und Protokolle

Da im Verlauf der Untersuchung ein Konzept zur Erweiterung eines bestehenden Systems vorgeschlagen werden soll, ist es notwendig, als einführende Grundlage gängige Begriffe und Architekturen zu erläutern. Ebenfalls erforderlich ist eine Betrachtung der verwendeten Protokolle, um diese gemäß ihrer Eigenschaften anwenden zu können.

2.1.1 Middleware

Der Begriff der MW ist in der Softwareentwicklung nicht eindeutig definiert, wird in der Praxis aber als Oberbegriff für vermittelnde beziehungsweise trennende Systemkomponenten genutzt [ASA01, S. 99] [Tre96]. Das Konzept der MW unterstützt elementar den in der Softwareentwicklung zentral vorhandenen Gedanken der Modularität [Koc05, S. 87 ff.] und findet daher weitreichend Anwendung. Die hohe Verbreitung von MW kann zusätzlich mit der verbesserten Wartbarkeit von Systemen durch die entstandene Entkopplung und Verschleierung der ursprünglichen Komponentenkomplexität erklärt werden [Bak03]. Zudem wird die Softwareentwicklung vereinfacht, da beispielsweise Transportalgorithmen nicht mehrfach vollständig implementiert werden müssen [KBL22, S. 99]. Ein beispielhafter Systemaufbau mit MW ist in Abbildung 2.1.1 dargestellt.

Zur genaueren Klassifizierung haben sich, basierend auf der Funktionalität der MW, verschiedene Kategorien etabliert. In der Fachliteratur wird der Begriff der Message-Oriented Middleware (MOM) diskutiert. Dieser beschreibt Softwarekomponenten, die bei der Kommunikation zwischen zwei anderen Systemkomponenten eingesetzt werden [ASA01, 38 ff.] [Tre96] [LG22]. Diese MW bieten Schnittstellen und Protokolle, um die Datenübertragung und den Nachrichtenaustausch zu verwalten. Bekannte Beispiele für MOM sind Message Queuing Systems wie RabbitMQ¹ oder

¹ Homepage: <https://rabbitmq.com>

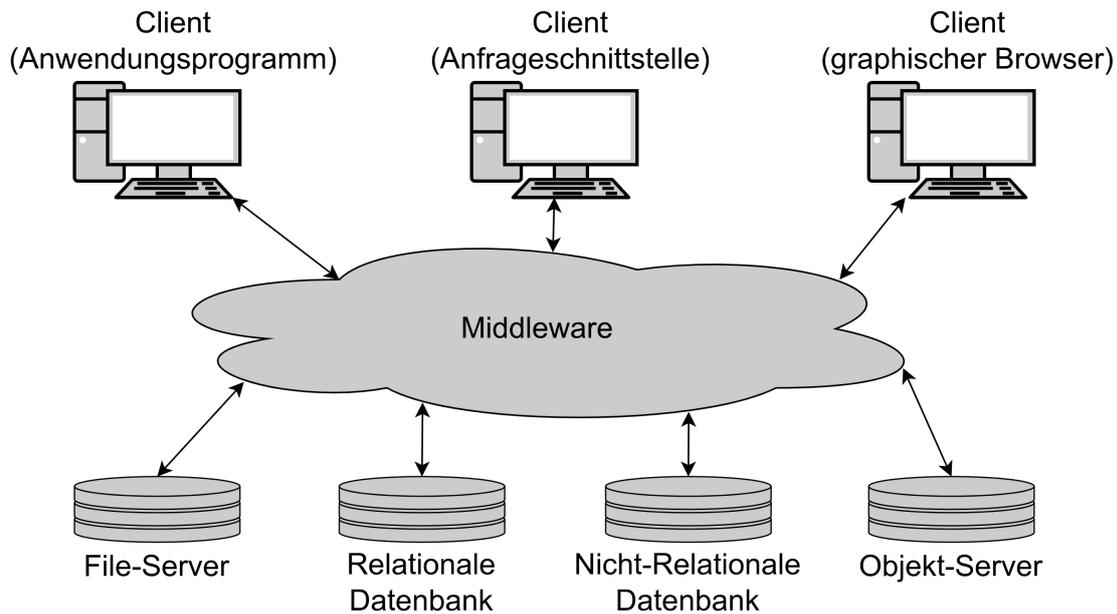


Abbildung 2.1.1: Beispielhafte Systemarchitektur mit MW nach Tresch [Tre96]

Apache Kafka².

Eine weitere Kategorie stellt die Distributed Object Middleware (DOM) dar. Hierbei handelt es sich um MW, welche verschiedene, heterogene Objekte für Systemkomponenten verfügbar macht [ASA01] [Tre96] [Bak03]. Common Object Request Brokering Architecture (CORBA)³ und Java's Remote Method Invocation (RMI) sind verbreitete Vertreter dieses Konzepts [ASA01, S. 101] [Bak03].

Da die Funktionalität von MW durch die generell formulierte Definition sehr weitgreifend ist, entstehen immer neue Kategorien, in denen sich entwickelte MW einordnen lässt. So beschreiben Mahmoud et. al. in ihrem Buch reflektive und adaptive MW, Transaction MW, Peer-to-Peer MW und Grid MW [CTM⁺04, S. 29 ff., 53 ff., 81 ff., 109 ff.], während Tresch den Begriff der Data Warehousing MW beschreibt [Tre96]. Mit weiterführender Sichtung verwandter Publikationen werden weitere Kategorien erörtert, wobei sich diese zunehmend in ihrer beschriebenen Funktionalität überschneiden und eine eindeutige Zuordnung von beispielhafter MW schwer möglich wird.

Im IoT-Bereich ist das Konzept der MW allgegenwärtig. Heterogene Sensorwolken produzieren eine große Datenvielfalt, welche zur Nutzung, Darstellung und Speicherung aufbereitet werden muss. Diese Vorgänge sollten, dem Gedanken des „separation of concerns“ folgend, an ab-

² Homepage: <https://kafka.apache.org>

³ Homepage: <https://omg.org/corba>

gegrenzte und extra dafür geschaffene Systemkomponenten abgegeben werden [KR03]. Zudem ermöglicht das Ausgliedern der verarbeitenden Logik weg von der Sensorik das Nutzen von sehr schlanken Sensorimplementationen. Dies wirkt sich positiv auf den Energieverbrauch, die Verfügbarkeit und damit wiederum auf die Wartbarkeit dieser aus [LG22, S. 143]. Auch der Datenlatenz kommt diese Architektur meist zugute. Da die Sensorclients aus beschriebenen Gründen oft über eine begrenzte Rechenkapazität verfügen, werden Algorithmen signifikant langsamer absolviert als auf dedizierten Hardware. Im schlimmsten Fall sind Daten zum Zeitpunkt des Berechnungsabschlusses bereits wieder irrelevant, wie es beispielsweise bei einer schnelllebigen Positionserfassung möglich ist. Dieses Problem kann aber ebenfalls bei einer zu feingliedrigen Architektur mit vielen Übertragungswegen auftreten, weshalb üblicherweise ein praxisorientierter Mittelweg gewählt wird.

2.1.2 Representational State Transfer

REST ist ein von Roy Thomas Fielding im Jahre 2000 vorgestelltes Architekturkonzept und entwickelte sich aus dem 1994 von ihm entworfenen HTTP Object Model [Fie00]. Es handelt sich dabei um eine Client-Server-Architektur und sieht eine Nutzung des etablierten Hypertext Transfer Protocol (HTTP) vor [NMM⁺99]. Damit ist diese Architektur klar abgrenzbar von Publish-Subscriber-Architekturen wie sie beispielsweise MQTT implementiert. REST ist zustandslos und unterstützt das Caching von Antworten [Fie00, S. 78 f.]. Die REST-Architektur besteht aus Datenelementen (Ressourcen, Repräsentationen, deren Identifier und Metadaten, Kontrolldaten), Konnektoren (Client, Server, Cache, Resolver, Tunnel) und Komponenten (Origin Server, Gateway, Proxy User Agent) [Fie00, S. 786 ff.].

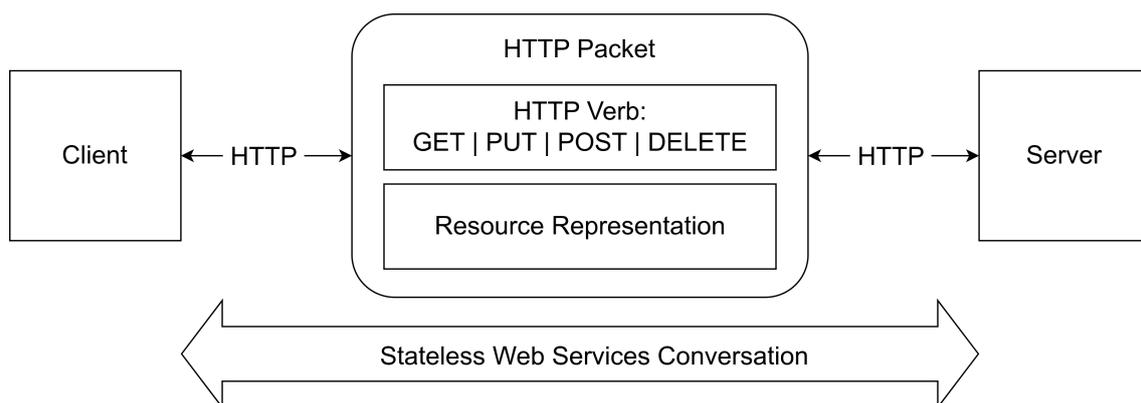


Abbildung 2.1.2: Kommunikation an einer REST-Schnittstelle nach Feng et al. [FSF09]

Für die Datenübertragung zur Schnittstelle werden vier HTTP Operationen vorgeschrieben: GET, POST, PUT und DELETE [FSF09], wie in Abbildung 2.1.2 veranschaulicht. Eine GET-Anfrage

liefert den derzeitigen Zustand einer Ressource in einer vordefinierten Repräsentationsform. GET wird als sicher angesehen, da der Ressourcenzustand nicht beeinflusst wird. Eine POST-Anfrage übergibt Informationen an eine bestimmte Ressource. Sollte die Ressource noch nicht existieren, muss der Server entscheiden, ob ein entsprechendes Objekt anzulegen ist. Durch Nutzung von PUT-Requests werden explizit neue Ressourcen auf dem Server angelegt oder existierende überschrieben. Die Anfrage enthält dafür das neue Objekt. Ein DELETE-Request löscht eine angegebene Ressource. GET, PUT und DELETE sind idempotente Operationen [FSF09] [NMM⁺99]. Für die Operation notwendige Informationen wie Identifikationsnummer oder Zustand können sowohl als Parameter in der Request-URI als auch als Message-Body übertragen werden. Welche Form der Übertragung genutzt wird, muss in der REST-Schnittstelle definiert werden.

Für die Antworten auf die spezifischen Anfragen wird ebenfalls eine HTTP-Nachricht erstellt. Diese enthält mindestens den HTTP Status Code bezüglich der angeforderten Operation, um grundsätzlich eine Form des Feedbacks an den Client zu gewährleisten. Fielding legt zudem nahe, dass die HTTP-Nachrichten selbsterklärend sein sowie über einen Request-Identifizierer eindeutig einer Anfrage zugeordnet werden können sollen [Fie00, S. 134]. Dies erlaubt dem Server ein asynchrones Bearbeiten der Anfragen, so dass zum Beispiel bestimmte Requests bevorzugt bearbeitet werden könnten.

Fielding grenzt seine REST-Architektur klar von ähnlichen Technologien wie RPC und RMI. Er begründet dies mit dem generischen Interface und der standardisierten Semantik, welche abstrahierte Ressourcen auf Netzwerkebene referenziert, wohingegen beispielsweise RPC sprachenspezifisch und nicht frei parametrisierbar ist [Fie00, S. 141 f.].

2.1.3 Message Queuing Telemetry Transport

Das MQTT ist ein vom IBM entwickeltes, auf dem Transmission Control Protocol (TCP) basierendes und durch den ISO Standard ISO/IEC-20922-2016 spezifiziertes Kommunikationskonzept [DC19] [noa16]. Es ermöglicht den Nachrichtentransfer zwischen mehreren Clients über einen zentralen Server, Broker genannt. Anders als bei traditionellen Client-Server-Architekturen wie jene einer REST-Implementation wird bei MQTT ein asynchroner beziehungsweise bidirektionaler Ansatz, vergleichbar mit einem Websocket, verfolgt [noa16].

Clients können sich beim Broker anmelden und für eine beliebige Anzahl Topics registrieren (subscribe). Diese Topics sind frei definier- und benennbar, sind meist in einer Baumstruktur organisiert und können on-demand erzeugt werden. Es ist auch möglich, sich auf alle bestehenden und zukünftig erzeugten Themen zu registrieren.

Der Broker verwaltet die Verbindungen zu den Clients und überwacht beispielsweise die Erreichbarkeit durch einen Health-Check [noa16]. Jedem Client ist es möglich, eine Nachricht auf eine frei definierbare Topic zu veröffentlichen (publishen), der Client muss dafür nicht selbst auf die Topic registriert sein. Die zu übermittelnde Nachricht wird zum Broker gesendet und dieser broadcastet die Nachricht an alle dafür registrierten Clients [DC19]. Der sendende Client erhält

zudem eine Acknowledgement-Nachricht als Bestätigung für eine erfolgreiche Übermittlung an den Broker [noa16]. Dies gilt auch für die Connect-, Disconnect-, Subscribe- und Unsubscribe-Nachrichten. Anzumerken ist, dass ein Client immer nur in Verbindung mit dem Broker steht. Es ist weder vorgesehen, dass die Clients untereinander direkt kommunizieren, noch dass ein Client Kontrolle hat, wer seine verbreiteten Nachrichten konsumiert.

Um die Nachrichtenqualität kontrollierbar zu machen, definiert der ISO Standard drei Stufen der Servicequalität, Quality of Service (QoS) genannt [noa16]. Diese beziehen sich auf die Anzahl der Zustellungen einer publizierten Nachricht. QoS 0 legt fest, dass eine Nachricht maximal einmal zugestellt wird. Dies macht Sinn bei unkritischen Nachrichten, deren Wegfall keinen nennenswerten Einfluss auf die Systemqualität hat. QoS 1 legt fest, dass eine Nachricht mindestens einmal gesendet wird. Dies ist relevant für wichtige, systemrelevante Informationen, deren Verteilung wichtig ist, aber deren potentielle Mehrfachverteilung keinen Schaden hervorruft. QoS 2 vereint beides und definiert, dass die Nachricht genau einmal zugestellt wird [DC19] [noa16]. Die QoS lässt sich für jede Nachricht separat definieren. Allerdings steigt mit erhöhtem Qualitätsniveau auch die Last am System, die eine einzelne Nachricht erzeugt. In der Praxis hat sich daher etabliert, für jede Nachricht die geringste notwendige Qualität auszuwählen und nicht statisch clientweit festzulegen.

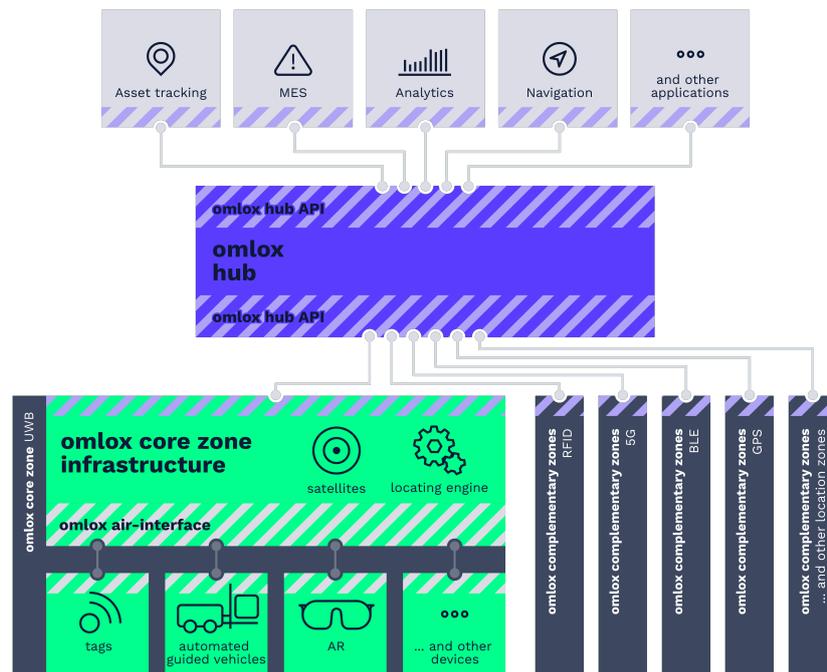
Zur Gewährung der Vertraulichkeit bietet der ISO/IEC-20922-2016 auch Vorschläge zur Authentifizierung und Verschlüsselung [noa16]. Basierend auf SSL und TLS oder anderen Clientzertifikaten sowie einer passenden Datenverschlüsselung wie AES lässt sich der Informationsstrom zuverlässig schützen [DC19].

2.2 omlox

Der omlox Standard wird von der PROFIBUS und PROFINET International Industriegruppierung, im nachfolgenden PI abgekürzt, entwickelt und gepflegt. Ziel dieses Standards ist es, verschiedene Lokalisierungstechnologien in einem System zu verbinden und somit die Vorteile aller Technologien voll ausschöpfen zu können. Dabei konzentriert sich PI vor allem auf industrielle Interessen, anders als vergleichbare UWB-Standards, zum Beispiel dem des FiRa Consortiums.

Zur Zusammenführung der heterogenen Technologien hat sich in der zweiten Version des omlox Standards die in Abbildung 2.2.1 dargestellte Systemarchitektur etabliert.

Das Gesamtsystem besteht aus zwei festen Bestandteilen: dem omlox Hub, hier blauviolett dargestellt, und der omlox Core Zone, hier hellgrün gefärbt. Die omlox Core Zone ist über das omlox air-interface mit Tags verbunden. Diese können an autonome Fahrzeuge oder anderen Geräten befestigt oder integriert werden und verwendet zur Lokalisierung ausschließlich UWB. UWB wird im Kontext der omlox Spezifikation auch In Band (IB) Kommunikation genannt. Für die Netzwerkorganisation werden zusätzlich nach IEEE 802.4.15 standardisierte Out of Band (OoB) Kommunikation im 2,4GHz Bereich eingesetzt. OoB Kommunikation ermöglicht beispielsweise ein

Abbildung 2.2.1: Übersicht über die omlox Architektur⁴

energieeffizientes Empfangen von Daten seitens der Tags. Parallel zur omlox Core Zone sieht die Architektur Möglichkeiten zur Implementation von omlox Complementary Zones vor. Alle implementierten Zonen sind über die interne omlox Hub API zum omlox Hub verbunden. Der Standard sieht vor, dass Nutzanwendungen wie die Datenauswertung oder Navigationsanwendungen an die öffentliche omlox Hub API angeschlossen werden.

2.2.1 omlox v2 Hub

Die omlox Hub Version 2.0.0 Spezifikation, im nachfolgenden als Hub bezeichnet und als Entwurf vorliegend, stellt eine Weiterentwicklung der ersten Version dar. Grundlegende Elemente, wie zum Beispiel die vorgesehene REST-Schnittstelle, definierte Objektschemata und die Möglichkeit auf eventbasierte Asynchronität werden spezifiziert. Ergänzt wird Version 2.0.0 durch Vorschläge zu Implementationen von MQTT und RPC. Die neue Hubspezifikation soll im Folgenden grob umrissen und wichtige Fachbezeichnungen eingeführt werden.

Die für diese Arbeit bedeutsamsten Begriffe sind Trackable und Location Provider (LP). Ein Trackable referenziert ein Objekt oder eine Person in der realen Welt, dessen Position bestimmt werden soll. Dies geschieht durch zugewiesene LPs. Diese in der Regel kleinen Geräte erzeugen

⁴ Entnommen aus dem firmeninternen Abbildungskatalog, öffentlich auch zugänglich auf der Firmenhomepage <https://zigpos.com/de/omlox>, zuletzt aufgerufen am 6.6.2023

Positionsinformationen durch verschiedene Technologien, beispielsweise UWB, RFID oder Wi-Fi Positioning System (WiPS). Bei den Trackables wird im Hub zwischen „omlox compatible“ Trackables und „virtual“ Trackables unterschieden. Omlox kompatible Trackables verfügen über eine Liste ihrer zugehörigen LP und ordnen sich diese automatisch bei Registrierung im Hub zu. Virtuelle Trackables werden über die dafür vorgesehene Hub API erstellt und manuell um LPs ergänzt. Bei den Positionsinformationen wird zwischen Position und Location differenziert. Eine Position ist eine zwei- oder dreidimensionale Koordinate und muss in Bezug auf ein Coordinate Reference System (CRS) interpretiert werden. Eine Location hingegen ist die Gesamtheit aus einer Position, dem zugehörigen Zeitstempel, der Quelle, dem genutzten CRS und ähnlichen relevanten Informationen. In Abbildung 2.2.2 ist dieser Zusammenhang grafisch zusammengefasst.

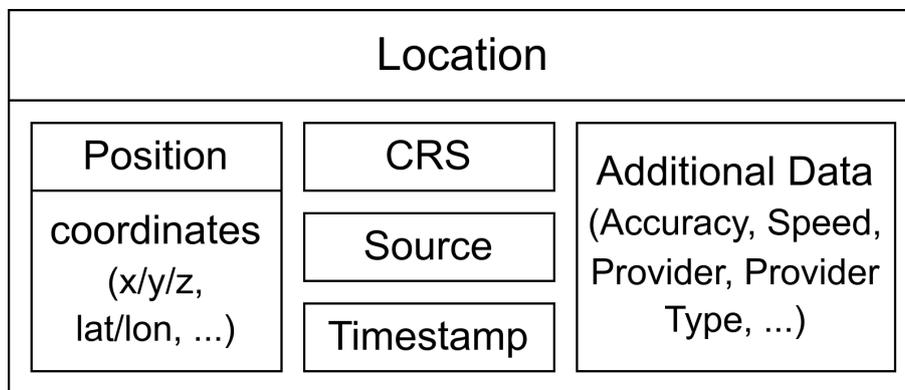


Abbildung 2.2.2: Grafische Zusammenfassung des Location-Objekts

Ein weiteres für diese Arbeit relevantes Konzept ist die Zone. Die Zone bezieht sich auf die verwendete Lokalisierungstechnologie und besitzt eine festgelegte geographische Form, definiert durch Ground Control Points (GCP). Ein GCP bildet eine geographische Koordinate auf ein lokales Koordinatensystem ab. Die Unterteilung in Zonen ermöglicht zum Beispiel die Definition von zwei Zonen, wobei in einer primär UWB genutzt wird, während in der anderen Zone GPS-LPs für Positionsdaten eingesetzt werden. Diese Zonen dürfen sich auch geographisch überlappen und können unabhängig voneinander existieren. Positionsinformationen aus nicht-geographischen Koordinatensystemen, also allen Systemen ausgenommen Global Navigation Satellite System (GNSS), werden vom Hub noch aufbereitet, bevor sie als Position im Sinne der omlox Hub Spezifikation genutzt werden. Darin begründet sich die Bedeutsamkeit der GCPs, ohne diesen Bezug zum geographischen Koordinatensystem ist eine Umwandlung nicht möglich.

Der beschriebene Sachverhalt bezüglich der Zonenorganisation ist in Abbildung 2.2.3 übersichtlich veranschaulicht. Zusätzliche, nicht weiter erläuterten Hub Spezifikationselemente wie Geofence-Objekte oder Events bilden zwar elementare Funktionalitäten des omlox Komplettsystems ab, haben für diese Arbeit allerdings eine geringe Bedeutung und sollen zugunsten des Umfangs hier nicht weiter erläutert werden.

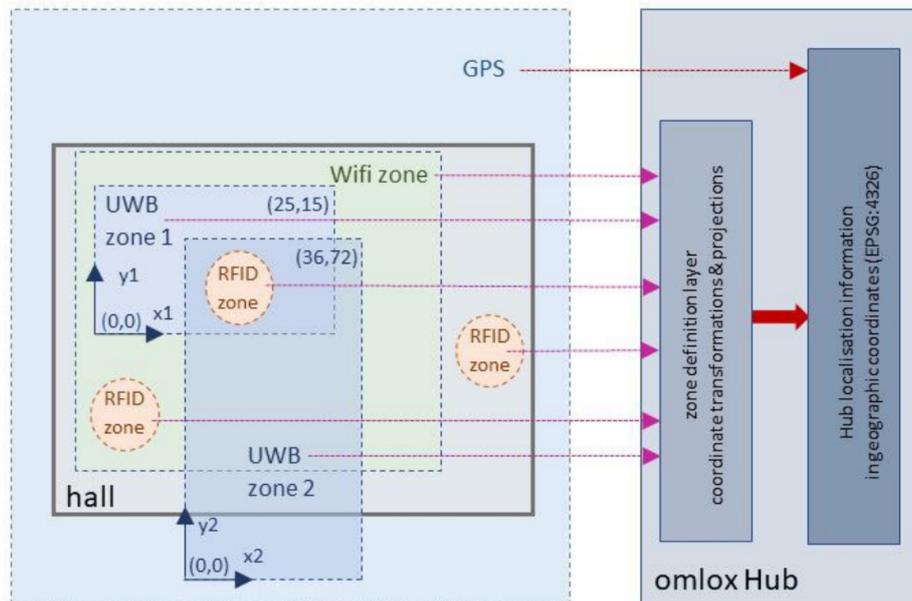


Abbildung 2.2.3: Beispielhafter Zonenbau eines omlox Systems mit sich überlagernden WiFi-, UWB-, RFID- und GPS-Zonen und Andeutung des Datenstroms zum omlox Hub⁵

Alle zuvor besprochenen Objekte und Konzepte können über die Hub-API verwaltet werden, für Trackable, LP und Zonen ist eine REST-Schnittstelle vorgesehen. Grundsätzlich wird bei allen drei Konzepten Methoden zur Erfassung und Löschung einzelner und aller derzeit erstellten Objekte sowie zur Erzeugung neuer Instanzen dieser Art zur Verfügung gestellt. Dies wird ergänzt um objektspezifische Funktionen, zum Beispiel dem Hinzufügen einer neuen Position für einen LP.

Durch die beschriebene Zonencharakteristik und der Option, einem Trackable mehrere LPs zuzuordnen zu können, ist es möglich, dass einem Trackable zeitgleich mehrere legitime Locations zugewiesen werden. Die omlox Hub Spezifikation sieht zur Lösung dieses Problems die Definition von „locating rules“ vor. Mit diesen kann in Abhängigkeit von gegebenen Parametern, zum Beispiel dem verwendeten Technologietyp und der Genauigkeitsangabe, Prioritäten erstellt werden. So ist es möglich, die durch UWB ermittelte Location gegenüber der GPS Location zu bevorzugen, solange eine definierte Genauigkeitsschwelle nicht überschritten wird. Ist keine Regel definiert, wird die jüngste Location übernommen.

Wie in Abbildung 2.2.1 dargestellt besitzt der Hub auch eine API zur Integration von Nutzeranwendungen, zum Beispiel einer grafischen Oberfläche zur Darstellung einer Karte. Für entsprechende Applikationen ist neben der REST-Schnittstelle auch eine Anbindung per Websocket vorgesehen, über welchen, angelehnt an das Publish/Subscriber-Konzept, Locationdaten und Events

⁵ Grafik entnommen aus den omlox Hub v2.0.0 Draft Dokumenten

erhalten werden können. Optional sieht die Hub v2 Spezifikation auch eine Message Queuing Telemetry Transport (MQTT)- und eine Remote Procedure Call (RPC)-Erweiterung vor. Während die MQTT-Anbindung eine ähnliche Funktionalität wie eine Websocketverbindung erfüllt, konzentriert sich die RPC-Implementation auf die Steuerung der Systemgeräte, zum Beispiel auf das Updaten der Firmware oder das Aufleuchten lassen einer Geräte-LED.

Abschließend soll die vorgesehene Verschlüsselung kurz erwähnt werden. Aufgrund der erhobenen, potentiell sensiblen Positionsdaten sieht die Spezifikation vor, dass für sämtliche Kommunikation zum Hub HTTPS beziehungsweise TLS genutzt werden soll. Es wird dabei die Nutzung von TLS 1.3 empfohlen, ergänzt um eine Fallback-Lösung mit TLS 1.2. Für die API Nutzung wird eine Authentifizierungsimplementation basierend auf OAuth2 vorgeschlagen.

2.2.2 omlox v2 Core Zone

Die omlox Core Zone Spezifikation, für diese Betrachtung als Entwurf in der Version 2.1.1 vorliegend und folgend als Core Zone bezeichnet, basiert auf den IEEE-Standards 802.15.4 und 802.15.4z. Die Spezifikation fokussiert sich im Kontext der Positionsbestimmung auf die Nutzung von UWB-Funk. Die Core Zone Spezifikation definiert zwei Arten von Geräten - Satelliten und Tags. Eine Core Zone ist aus einer Vielzahl von Satelliten mit fester, bekannter Position und beliebig vielen Tags, mobilen Geräten, deren Position bestimmt werden soll, aufgebaut.

Die Satelliten werden in drei Rollen mit absteigender Funktionalität eingeteilt. Ein Full Blown Satellite (FBS) sendet regelmäßig Kontrollnachrichten um ein Time Division Multiple Access (TDMA) im System zu ermöglichen. Die Kontrollnachrichten dienen zur pikosekundengenauen Zeitsynchronisation, verhindern somit einen potentiellen Clock Drift und mitigieren umgebungsbedingte Verzögerungen. Zusätzlich sammelt der FBS die Messdaten von den anderen Satelliten. Alle Daten inklusive der Ranging-Antworten werden einem übergeordnetem Layer zur Verfügung gestellt. Ein Main Powered Satellite (MPS) sammelt ebenfalls Messdaten, leitet diese aber exklusiv zu einem FBS, anstatt sie einer übergeordneten Schicht zur Verfügung zu stellen. Ein Battery Powered Satellite (BPS) unterstützt ausschließlich bei der Verwendung von Downlink-Time Difference of Arrival (DL-TDoA). Er kann somit vereinfacht als eine zusätzliche Zeitreferenz für die Tags betrachtet werden. Die Spezifikation stellt frei, im Kontext der Geräteverwaltung auf beliebige OoB-Technologien zurück zu greifen.

Tags sind in drei Kategorien differenzierbar. Transceiver Tag (TRxT), Transmitter Tag (TxT) und Receiver Tag (RxT). Die Funktionalitäten lassen sich kurz analog zu der Benennung beschreiben. Ein TxT kann IB Nachrichten senden, ein RxT nur empfangen. Ein TRxT besitzt beide Funktionen. Aus dieser Differenzierung lassen sich die möglichen Positionsermittlungsverfahren, genauer erklärt in Abschnitt 2.3, ableiten. Senden und empfangen von IB-Funk ermöglichen respektive UL- beziehungsweise DL-TDoA. Ein TRxT eignet sich zudem auch für das Time of Flight (ToF)-Verfahren, erläutert in Abschnitt 2.3.2.

Bezüglich der IB Kommunikation legt die Spezifikation fest, dass Geräte mindestens High Pulse

Repetition Frequency (HPRF) auf Channel 3 (Kernfrequenz 4492,9MHz) oder Base Pulse Repetition Frequency (BPRF) auf Channel 9 (Kernfrequenz 7987,2MHz) unterstützen müssen. Es handelt sich dabei also um angewandte Frequenzmodulation, genauer aufgegriffen in Abschnitt 2.4.1. Zusätzlich wird die Größe der Payload auf maximal 64 Byte beschränkt, um ein maximale Frame-Duration von 333,33 μ s zu erreichen. Die zur Positionsbestimmung genutzte Ranging Phase ist in verschiedene Timeslots unterteilt, die für bestimmte Arten des Trackings reserviert sind. Continuous Tracking (CT) ist für Tags bestimmt, deren Position in jeder Rangingiteration bestimmt werden sollen. Sie haben einen festen zugewiesenen Timeslot. Ad-hoc Tracking (AT) ist für Geräte, deren Position seltener erfasst werden soll. Ihnen können zwar garantierte Timeslots zugeordnet werden, sie dürfen aber auch in einem frei verfügbaren, unreservierten Zeitfenster senden. Reservierte Timeslots können erneut zugewiesen werden, was die Skalierbarkeit einer Core Zone ermöglicht. Geräte, die das DL-TDoA-Verfahren nutzen, brauchen durch die in Abschnitt 2.3.3 beschriebene Natur dieses Prozesses keinen eigenen Timeslot.

Die omlox Core Zone Spezifikation definiert noch weitere Verhaltensweisen, Funktionen und Konventionen, wie beispielsweise einen Over The Air Update (OTAU) Prozess und damit verbundenes Gerätezustandsmanagement oder den konkreten Paketaufbau einer IB-Nachricht. Dies übersteigt aber den Rahmen dieser Betrachtung und soll daher nicht weiter ausgeführt werden.

2.2.3 omlox v2 Complementary Zone

Für eine Implementation einer omlox Complementary Zone, im folgenden nur als Complementary Zone bezeichnet, gibt die omlox Hub Spezifikation wenig vor. Durch die in Abschnitt 2.2.1 genannte Möglichkeit zur Erstellung eines virtuellen Trackables sind LPs direkt zuweis- und miteinander kombinierbar. Da die Positionsdaten nicht die Form von geographischen Koordinaten haben müssen und der Hub eine Umrechnung der gegebenen Daten realisieren kann, ist die genutzte Lokalisierungstechnologien frei wählbar. Die Hub Spezifikation legt fest, dass neben der Core Zone, basierend auf UWB, auch Zonen für WiFi, RFID und iBeacon realisiert werden können müssen. Im Falle von RFID und iBeacon legt die Spezifikation nahe, diesen Zonentyp über eine Position in Form einer geographischen Koordinate zu definieren, anstatt wie bei UWB und WiFi über mindestens vier GCPs.

Aufgrund dieser flexiblen Beschreibung einer Complementary Zone lässt sich ein breites Spektrum an Technologien einsetzen. Wenn für die gewählte Technologie eine Überführung von einem lokalen Koordinatensystem in geographische Koordinaten innerhalb des Hubs realisiert wird, ist diese nutzbar. Man kann also frei aus weit verbreiteten Technologien wie Bluetooth Low Energy (BLE) wählen oder auch eigene Speziallösungen wie ein manuelles Positionsupdate über eine Eingabemaske realisieren.

2.3 Positionsermittlungsverfahren

Im Nachfolgenden sollen verbreitete Konzepte zur Positionsermittlung mittels Funktechnologien erläutert werden. Die Verfahren sind dabei von der Technologie unabhängig, werden aber in der Praxis mitunter unterschiedlich implementiert. Daher sind entsprechenden Konzepte in der folgenden Ausführung generell und allgemeingültig beschrieben, um sie in folgenden Kapiteln in Bezug zu etablierten Technologien zu setzen.

2.3.1 Received-Signal Strength

Received-Signal Strength (RSS) ist ein Verfahren, welches mit Hilfe des Sendens eines Signals und der Messung der eingehenden Signalstärke beim Empfänger eine Entfernungsabschätzung ermöglicht. Für ein eingehendes Signal wird der Zeitstempel und der Received-Signal Strength Indicator (RSSI) erfasst. Der RSSI ist der abstrahierte, hardwareabhängige Signalleistungswert, der für die Berechnung genutzt wird. Um diese Rechengröße verwerten zu können, sind Referenzmessungen, Reference Point (RP) genannt, mit bekannter Entfernung notwendig [CMJS13]. Mit Hilfe der gemessenen RSSI der RP kann die zukünftige Messung der RSSI unbekannter Entfernungen eingeordnet werden.

Da zur Erfassung nur das Empfangen eines Signals notwendig ist, kann RSS grundsätzlich mit jeder Funktechnologie ohne Aufwendung zusätzlicher Hardware genutzt werden [KK04]. Zu beachten ist die Umwelt, in der das Signal übertragen werden soll [CMJS13]. Diese muss für einen minimalen Messfehler identisch mit der für die Referenzmessung genutzten Umgebung sein, um alle Dämpfungsfaktoren zu erfassen. Dies ist praktisch schwer umsetzbar. Außerhalb von Gebäuden ist die Umwelt in permanenter Veränderung, innerhalb eines geschlossenen Raumes sind temporäre Veränderungen selten ausschließbar. Damit können Effekte, die Signalausbreitung verändern, beispielsweise Störstrahlung oder neu entstandene Mehrwegausbreitung durch bewegliche Körper, die Entfernungsberechnung stark verfälschen [CMJS13]. Auch eine Änderung der Ausrichtung des Empfängers kann den Signalwert um bis zu zwei Dezibel verfälschen [CMJS13]. RSS ist somit ein sehr einfaches, aber vergleichsweise ungenaues Verfahren.

2.3.2 Time of Flight

Analog zu RSS bezieht sich auch das Time of Flight (ToF)-Verfahren auf die Kommunikation zwischen zwei Geräten, eine Beschreibung im Kontext der UWB-Technologie hinterlegt das IEEE im Standard 802.15.4z [noa20b]. Im Kontrast zu RSS wird bei ToF der Zeitstempel der gesendeten Pakete ausgewertet und mit der intern ermittelten, aktuellen Zeit, der Time of Arrival (ToA) verglichen. Unter der Annahme, dass sich das Signal gleichmäßig und mit konstanter Geschwindigkeit im Raum ausbreitet, wird die so ermittelte namensgebende Paketflugzeit zur Entfernungsabschätzung verwendet. Dies setzt voraus, dass beide Geräte über eine synchrone, innere Uhr verfügen. Zusätzlich muss eine rechen- und hardwarebedingte Verzögerung in Betracht gezogen

werden. Um diese Messfehler zu minimieren, definiert die IEEE eine Erweiterung des traditionellen ToF-Prozesses um das Senden eines Antwortpaketes. Dies wird Single-sided Two Way Ranging (SS-TWR) genannt und ist in Abbildung 2.3.1 dargestellt.

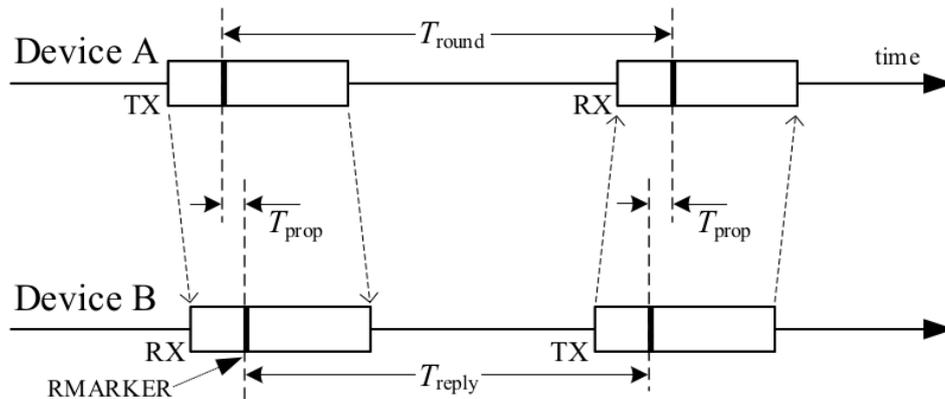


Abbildung 2.3.1: Darstellung des SS-TWR-Vorgangs nach IEEE Standard 802.15.4z [noa20b]

Die Abbildung zeigt den Paketaustausch zwischen Gerät A und Gerät B. Gerät A sendet ein Paket an Gerät B, wobei das Senden beziehungsweise Empfangen mit einem Zeitstempel dokumentiert wird. Nach dem Senden des Antwortpaketes können so die Antwortzeit T_{reply} und die Umlaufzeit T_{round} erfasst werden [noa20b]. Die ToF ermittelt sich somit wie folgt:

$$T_{ToF} = \frac{1}{2} \cdot (T_{round} - T_{reply})$$

Der Standard 802.15.4z des IEEE erweitert diese Formel auch um den Clock Offset C_{offs} . Diese Formel reduziert den Messfehler und gilt, wenn Gerät A eine Möglichkeit hat, die relative Uhrendifferenz zu Gerät B zu messen [noa20b].

$$T_{ToF} = \frac{1}{2} \cdot (T_{round} - T_{reply} - (1 - C_{offs}))$$

2.3.3 Time Difference of Arrival

Das Time Difference of Arrival (TDoA)-Verfahren ist im Bezug auf UWB im IEEE-Standard 802.15.4z genauer definiert [noa20b], findet aber auch bei anderen Technologien Anwendung. Grundsätzlich wird für dieses Verfahren zusätzlich zu dem Mobilgerät eine Menge von zeitsynchronen Systemkomponenten mit bekannter Position benötigt. Das Verfahren kann je nach Kommunikationsrichtung in zwei Unterverfahren differenziert werden, Downlink-Time Difference of Arrival (DL-TDoA) und Uplink-Time Difference of Arrival (UL-TDoA).

Bei DL-TDoA wird ein Paketfluss von den übergeordneten Systemkomponenten hin zum Mobil-

gerät genutzt. Unter Annahme, dass sich Signale gleichmäßig im Raum ausbreiten, senden die positionsdefinierten Geräte ein Signal mit den Positionsinformationen und einem Zeitstempel. Analog zu dem in Abschnitt 2.3.2 beschriebenen Vorgang wird dieses Signal vom mobilen Endgerät empfangen und der Zeitstempel ausgewertet. Durch die Zeitdifferenz und den bekannten Signalursprung kann das Mobilgerät seine Position herleiten. Voraussetzung dafür ist, dass mehrere dieser Positionssignale empfangen werden, da die Position sonst nicht eindeutig bestimmt werden kann. Ein Vorteil dieses Ansatzes liegt in der beliebigen Anzahl an Mobilgeräten im System. Da ein Signal beim Empfangen nicht verbraucht wird, können alle Endgeräte unabhängig voneinander Informationen aus dem selben Signal verwerten.

Analog zum beschriebenen DL-TDoA-Vorgang kann die Kommunikationsrichtung auch umgekehrt werden. Wenn das Mobilgerät eine Zeitinformation sendet, kann über die positionsdefinierten Systemkomponenten ebenfalls über eine Zeitdifferenzbetrachtung die Position des Mobilgeräts hergeleitet werden. Diese Herangehensweise wird genutzt, wenn die gewonnene Positionsinformation systemzentral weiterverarbeitet werden soll oder das mobile Endgerät keine komplexen Gleichungen performant lösen kann.

2.4 Echtzeitlokalisierungstechnologien

Eine Vielzahl von Funktechnologien eignen sich für eine Positionsbestimmung. Beispielsweise Bluetooth Low Energy (BLE) und RFID finden weit verbreitet praxisnahe Anwendung. Nachfolgend soll daher keine vollständige Liste, sondern eine Auswahl an möglichen Technologien diskutiert werden. Dabei wird bevorzugt auf Technologien eingegangen, die für den im kommenden Kapitel beschriebenen Industriefall relevant sind.

2.4.1 Ultra-Wide Band

Ultra-Wide Band (UWB) ist ein durch das Institute of Electrical and Electronics Engineers (IEEE) spezifizierter und durch das European Telecommunications Standards Institute (ETSI) und Federal Communications Commission (FCC) regulierter Nahbereichsfunkstandard. In den 2000er entstanden, charakterisiert sich diese Kommunikationslösung durch ein breites Frequenzband von 3,1GHz [noa23] bis 10,6GHz [FCC20, 968] und einer geringen Sendeleistung von maximal 0,5mW [OHI04]. Der Standard 802.15.4 [noa20c] sowie die Ergänzung 802.15.4z [noa20b] definieren den Signalaufbau, allerdings wird diese Technologie zusätzlich ortsabhängig reguliert. Die Nutzung in Nordamerika wird durch die FCC, in Europa durch das ETSI geregelt und mit den bereits etablierten Funklösungen in Einklang gebracht. Zugunsten einer globalen Verbreitung hat sich eine Achtung beider Regulationsbehörden eingestellt, da außerhalb Europa und Nordamerika viele Staaten die Regulation der Behörden adaptieren. Dies geht aus einem Dokument der Firma Qorvo hervor [noa20a]. Die FCC hat in veröffentlichten Dokumenten die Signalforn verbindlich beschrieben, was folgend mit Abbildung 2.4.1 erläutert werden soll.

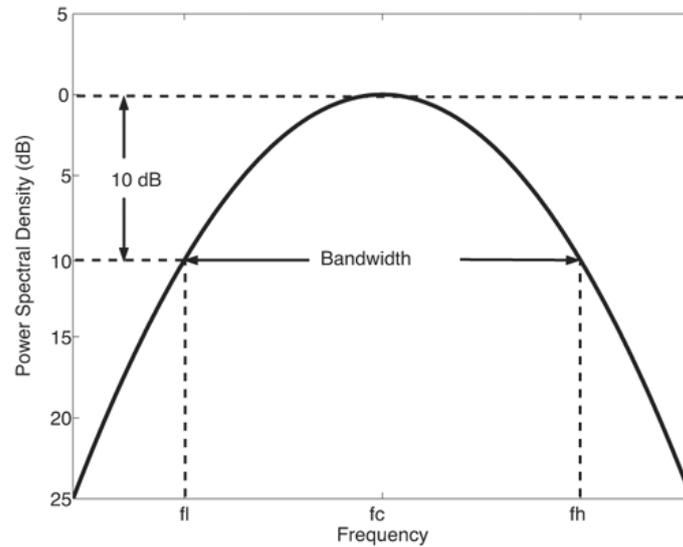


Abbildung 2.4.1: Frequenz-Spektraldichte-Diagramm zur Definition von UWB-Signalen [Aug23]

Die absolute Bandbreite B_{abs} eines Signals wird durch zwei Punkte definiert, welche 10dB unterhalb der maximalen Sendeleistung auf der Dichteparabel liegen. Durch diese zwei Punkte werden die untere Frequenzgrenze f_l und die obere Frequenzgrenze f_h abgeleitet, diese wiederum definieren die mittlere Frequenz f_c sowie die relative Bandbreite B_{rel} .

Die Spektraldichteparabel wird maßgeblich durch drei Werte definiert: dem globalen Maximum bei der Kernfrequenz f_c sowie der unteren und oberen Frequenzgrenze, f_l und f_h , aus denen die absolute Bandbreite B_{abs} sowie die relative Bandbreite B_{rel} abgeleitet wird. Es gilt:

$$f_c = \frac{f_h + f_l}{2}$$

$$B_{abs} = f_h - f_l$$

$$B_{rel} = 2 \cdot \frac{f_h - f_l}{f_h + f_l}$$

Für eine Konformität mit der geltenden FCC-Richtlinie muss die relative Bandbreite B_{rel} größer oder gleich 0,20 sein, oder die absolute Bandbreite B_{abs} mindestens 500MHz betragen [FCC20, 966]. Zusätzlich gilt für jede Frequenz eine maximale zulässige Sendeleistung [Aug23]. Dies verhindert, dass bestehende Technologien, wie beispielsweise das Global Positioning System (GPS), eingeschränkt werden. Durch diese Charakteristiken nutzt die UWB-Technologie vergleichsweise robuste und störsichere Signale, was durch die zunehmend global wachsende Funkverbindungsanzahl wichtig ist.

Die mittlere Frequenz f_c wird zum Definieren der Bandbreitenregion, auf der das Signal übertragen wird, genutzt. Diese wird auch als Channel bezeichnet [noa20b] und wird im IEEE-Standard ein-

deutig definiert, weshalb man in der Praxis zuerst den UWB-Channel bestimmt, um daraus die Signalparameter abzuleiten. Anhand der diskreten Verteilung der einzelnen Channels ist zu erkennen, dass bei dem Standard das Konzept der Frequenzmodulation über die komplette Bandbreite verfolgt wurde.

Zusätzlich zur Frequenzmodulation wird auch das Prinzip der Codemodulation umgesetzt. Durch die Nutzung von Präambel-Codes unterschiedlicher Länge werden Signale auch innerhalb des selben Channels differenzierbar, womit mehrere Systeme unabhängig voneinander im selben Channel existieren können. In der Praxis hat sich zudem eine Zeitmodulation durch Timeslots etabliert, nicht zuletzt durch den hier erläuterten omlox Standard. Durch die Definition von Sendebeziehungsweise Empfangsperioden wird so die vom Channel begrenzte Bandbreite effizient ausgeschöpft. Besonders in Hinsicht auf den erhöhten Kommunikationsbedarf der im IoT üblichen Sensorenwolken überzeugt diese Dreifachmodulation.

Hinsichtlich der in Kapitel 2.3 besprochenen Verfahren ist Abschnitt 2.3.3, TDoA, hervorzuheben. Die weiteren besprochenen Verfahren sind zwar prinzipiell möglich, allerdings sind das in der Betrachtung genauer aufgeführte Two-Way Ranging (TWR), eine Weiterentwicklung des ToF, und das erläuterte TDoA zu bevorzugen. Dies begründet sich auf der mit diesen Verfahren erzielten Genauigkeit, mit UL-TDoA sind Positionen bei bis zu zehn Zentimeter Genauigkeit möglich.

2.4.2 WiFi Positioning System

Als WiFi Positioning System (WiPS) bezeichnet man die Anwendung von Wireless Local Area Network (WLAN) im 2,4GHz und 5,0GHz Frequenzbereich basierend auf dem IEEE Standard 802.11az zur Positionsbestimmung [noa22]. Der traditionelle WLAN-Aufbau, also das gleichmäßige Verteilen von Knoten, Access Point (AP) genannt, unterstützt das bereits beschriebene RSS Verfahren. Zusätzlich kann durch den Aufbau des Netzwerkprotokolls, spezifisch dem Anmelden an einem AP, eine Zellenzugehörigkeit abgeleitet und dadurch die Position auf nahe eines AP festgelegt werden [KHNF05]. Mit zusätzlicher Hardware ist eine Implementation von ToF und AoA möglich, TDoA kann mit zeitsynchronen Geräten implementiert werden [KHNF05]. TDoA setzt zusätzlich eine dichte Verteilung der AP voraus. Durch die breite Auswahl an Verfahren bei dem Einsatz von WiPS zeigt sich die Flexibilität dieser Technologie. Die Problematik liegt im Aufbau der passenden Netzwerkstruktur. Ein AP agiert nicht autonom, sondern ist die Schnittstelle zwischen kabellosen und fest installierten, bereits funktionsfähigem Netzwerk.

Unter Nutzung von RSS und MagicMap, einer Software zur Organisierung einer netzwerkbasierten kooperativen Kontextkenntnis, gelingt Ibach et. al in ihrer Arbeit eine durchschnittliche Genauigkeit von 3-5 Meter mit einer Messverzögerung von 150 Sekunden bei einer Warenhausanwendung zu erreichen [ISL⁺]. In der Arbeit wird betont, dass die Signalstärke stark abhängig von der Umwelt ist. Velayos und Kalrsson bestätigen dies in ihrer Arbeit, so wird bei einer Distanz von 30m bereits ein nicht linearer Signalstärkenverlust von -100dBm erkannt [VK].

2.4.3 Global Navigation Satellite System

Der Begriff Global Navigation Satellite System (GNSS) bezeichnet mehrere Satellitensysteme, welche von unterschiedlichen Nationen betrieben und zur weltweiten Positionsbestimmung genutzt werden. Die bekanntesten sind das US-amerikanische Global Positioning System (GPS), das europäische Galileo und das russische GLONASS. Nennenswert sind auch das chinesische Beidou und das Navigation Indian Constellation (NAVIC) aus Indien. GPS nutzt 1278MHz, 1227MHz und 1575MHz als Kernfrequenz für Signale. Galileo nutzt ebenfalls 1278MHz und 1575MHz, zusätzlich werden 1256MHz, 1278MHz und 1544MHz genutzt. GLONASS wird ebenfalls im 1176MHz und 1575MHz Bereich betrieben, sowie im 1268MHz und 1,246MHz bereich. Hervorzuheben ist, dass bei dem russischen System FDMA und CDMA eingesetzt werden, bei GPS und Galileo gibt es keine Modulation. GPS, GLONASS und Galileo sind MEO-Systeme, NAVIC benutzt geostationäre Systeme. Beidou kombiniert beide Systemarten [LG20, S. 208].

GNSS verwendet DL-TDoA als Positionserfassungstechnologie. Das erlaubt bei ausreichend großem Satellitennetz eine unbegrenzte Anzahl an GNSS Empfängern. Allerdings braucht ein Empfänger eine direkte Line of Sight (LOS) zu einer ausreichend großen Anzahl von Satelliten, in der Regel vier, um die eigene Position aus den ungestörten Signalen ableiten zu können [YBMD14]. Dies bedeutet, dass in Gebäuden oder auch in stark bebauten Gebieten eine Positionsbestimmung mittels GNSS fehlerbehaftet sein kann. Man geht von einer Messgenauigkeit von 2-3m bei idealen Bedingungen aus [YBMD14].

Analog zu dem bereits beschriebenen TDoA Verfahren senden die Satelliten ihre Umlaufdaten und einen Zeitstempel. Da die Position im Falle der MEO-Satelliten nicht konstant ist, wird diese vom Empfänger aus der gesendeten Umlaufbahn und der Zeit abgeleitet [Dig22]. Die Zeitsynchronität des Satellitensystems wird durch eine steuernde Basisstation gewährleistet. Die empfangenen Informationen werden gemäß der Spezifikation der National Marine Electronics Association (NMEA) dargestellt [SA11], die ermittelte, aktuelle Position wird in der Recommended minimum sentence C (RMC) Ausgabe durch eine Latitude und eine Longitude ausgedrückt [SA11]. Um die Zuverlässigkeit der GNSS-Daten zu steigern, ist es möglich, den Messfehler durch einen nahen, stationären GNSS-Empfänger mit bekannter Position zu ermitteln. Man geht durch die Nähe der beiden Empfänger zueinander davon aus, dass der Messfehler identisch ist. Da sich der Fehler bei der lokalen Station berechnen lässt, kann dieser bei dem mobilen Empfänger, Rover genannt, als bekannt betrachtet und somit entfernt werden. Dieses Verfahren wird als Differential Global Navigation Satellite System (DGNSS) bezeichnet [TM17]. Ein ähnliches Verfahren wird bei Real Time Kinematic (RTK) verwendet. Die Korrekturdaten werden hier üblicherweise mit dem Network Transportation of RTCM Internet Protocol (NTRIP) oder direkt per Funk als „pseudorange“ Informationen an den Rover übertragen [TM17].

2.4.4 Mobilfunk

Unter dem Mobilfunk-Begriff werden mehrere alltäglich genutzte Technologien zusammengefasst. Sie werden in der Fachliteratur als Generationen bezeichnet, welche die Entwicklung der großflächigen Funkverbindungen darstellt. Die erste Generation wird aufgrund ihrer analogen Grundstruktur vernachlässigt, bei Diskussionen bezieht man sich daher vor allem auf 2G, Global System for Mobile Communications (GSM), und die folgenden, ebenfalls digitalen, Generationen [LG20]. 5G und das derzeit zu definierende 6G besitzen hohe mediale Aufmerksamkeit. Allerdings nutzen Anwendungen noch heute GSM [LG20].

Luntovskyy et. al. nutzen in ihrem Werk die in Tabelle 2.4.1 sinngemäß dargestellte tabellarische Übersicht [LG20, S. 209].

Generation	Technologie	Übertragungstyp	Datenrate
1G	AMPS	Analog, leitungsvermittelt	-
2G	GMS	Digital, leitungsvermittelt	9,6 kbit/s
2.5G	HSCSD	Digital, leitungsvermittelt	57,6 kbit/s
	GPRS	Digital, paketvermittelt	115 kbit/s
2.75G	EDGE	Digital, paketvermittelt	236 kbit/s
3G	UMTS/UTRA FDD	Digital, meist paketvermittelt	384 kbit/s
	UMTS/UTRA TDD	Digital, paketvermittelt	2 Mbit/s
3.5G	HSPA (HSPD, HSUPA)	Digital, paketvermittelt	14,4 Mbit/s
3.9G	LTE	Digital, paketvermittelt	150 Mbit/s
4G	LTE Advanced	Digital, paketvermittelt	1 Gbit/s
5G	IMT2020	Digital, paketvermittelt	10... 100 Gbit/s

Tabelle 2.4.1: Tabellarische Übersicht der Mobilfunkstandards und ihrer Technologie, des genutzten Übertragungstyps und der verfügbaren Datenrate von der ersten bis zur fünften Generation

Die Tabelle veranschaulicht die rasante Entwicklung, die man von 1990 (GSM) bis 2020 (5G) beobachten konnte. Hinsichtlich der möglichen Datenrate wurden große Steigerungen erreicht, unter Anderem begründet durch den Einsatz von MIMO-Antennentechnik [LG20]. Zugunsten des Umfangs dieser Betrachtung soll folgend nur die vierte und fünfte Generation diskutiert werden.

Long Term Evolution (LTE) implementiert eine Weiterentwicklung von Orthogonal Frequency Division Multiplexing (OFDM) namens Orthogonal Frequency Division Multiple Access (OFDMA). Durch diese adaptive Modulation und der MIMO-Technologie gelang dem 3rd Generation Partnership Project (3GPP) 2004 bis zu 1Gbit/s. Die genutzten Frequenzen sind länderspezifisch, in Deutschland wird für LTE die Bereiche um 800MHz, 1,8GHz, 2,0GHz und 2,6GHz verwendet. Durch die Implementation des SIP-Protokolls wird ein IP Multimedia Subsystem (IMS) zur Verfügung gestellt. Die von 3GPP entwickelte Architektur erlaubt somit die Koexistenz zu Funktechnologien der dritten Generation [LG20].

Der Standard 5G/IMT2020 wird von Netzwerkspezialisten der Industrie geformt [LG20]. Für die fünfte Generation des Mobilfunks wurde ein neues Modulationsverfahren, das Feher's Quadrature Amplitude Modulation (FQAM), entwickelt, um eine bessere Energieverteilung innerhalb

der Funkzelle zu gewährleisten. Auch die Antennentechnik wurde verbessert, so wird für 5G die Nutzung von Multi-User Multiple In Multiple Out (MU-MIMO) und Full-Dimension Multiple In Multiple Out (FD-MIMO) Antennen vorgesehen. Diese komplexen Antennenarrays begünstigen die von der Industrie angestrebten hohen Datenraten. Zudem weisen 5G Netze eine geringe Latenz von fünf Millisekunden auf, verglichen mit 50 Millisekunden Latenz bei LTE [LG20, S. 233].

Durch die komplexe Modulation kann 5G parallel zu vorhergehenden Generationen auf der selben Frequenz betrieben werden. In Deutschland wurde durch das Bundesministerium für Umwelt, Naturschutz, nukleare Sicherheit und Verbraucherschutz (BMUV) zusätzlich die Nutzung des Frequenzbereichs von 3,4GHz bis 3,8GHz und von 24,25GHz bis 27,5GHz freigegeben⁶. Hervorzuheben ist der Bereich von 3,7GHz bis 3,8GHz. Dieses Frequenzband kann ohne Antrag für Anwendungen der Industrie genutzt werden.

Durch die Zellenarchitektur des Mobilfunknetzes ist eine Cell-ID zur Positionsabschätzung und das ToF-Verfahren, in diesem Kontext Time-Advance (TA)-Verfahren genannt, permanent möglich. Wenn sich hinreichend Funkzellen überlagern, ist das Enhanced Observed Time Difference (EOTD)-Verfahren möglich. Dieses funktioniert basierend auf der Netzarchitektur analog zu dem bereits beschriebenen UL-TDoA-Verfahren. Dies ist generell nur in dicht besiedelten Regionen möglich, da in ländlichen Regionen die nötigen Sendeantennendichte nicht erreicht wird. Durch Kombination des Cell-ID-Verfahrens und der TA-Analyse kann innerorts die Position auf bis zu 100m genau bestimmt werden. Außerhalb urbaner Gebiete kann die Genauigkeit auf zehn Kilometer sinken. Bei einer Anwendung des EOTD-Verfahrens kann eine Position auf bis zu 30 Meter genau bestimmt werden [SF15].

Innerhalb des LTE Standards haben sich zwei weitere Standards für den IoT Bereich etabliert, Narrow Band Internet of Things (NB IoT) und Long-Term Evolution for Machines (LTE-CAT-M1), auch LTE-M genannt. Beide Technologien funktionieren mit der gängigen Infrastruktur der vierten Generation und zeichnen sich durch eine hohe Reichweite bei geringem Energieverbrauch aus.

⁶ Siehe www.bmuv.de/faq/welche-frequenzen-nutzen-die-5g-netze, zuletzt aufgerufen am 25.7.2023

Kapitel 3

Konzeptionierung

3.1 Beschreibung des Industriefalls

Ein realitätsnaher industrieller Anwendungsfall soll den Kontext für die Konzeptionierung und Implementierung des hybriden Lokalisierungssystems bestimmen. Es handelt sich um ein US-amerikanisches Logistikunternehmen mit eigenem Fuhrpark auf einem abgeschlossenen, ebenerdigen Gelände. Wie in Abbildung 3.1.1 schematisch dargestellt beinhaltet das beispielhafte Firmengrundstück ein Gate mit Pförtner zum Betreten und Verlassen des Grundstücks, vier Lagergebäude mit mehreren Be- und Entladezonen sowie einen Parkplatz zum Abstellen der firmeneigenen Zugmaschinen und Anhänger. Das Firmengelände umfasst circa ein Quadratkilometer Fläche.

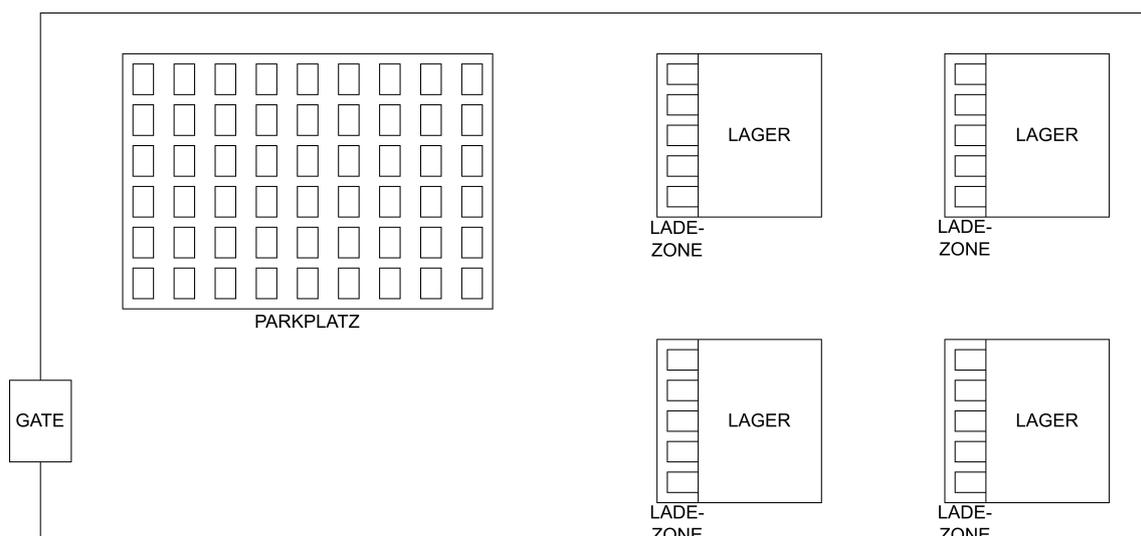


Abbildung 3.1.1: Schematische Darstellung des vorliegenden industriellen Anwendungsfalles

Die Lagerhallen sind gebäudeübergreifend mit einem RTLS der ZIGPOS GmbH und dem Deep-Hub der Flowcate GmbH ausgestattet. Diese Kombination ermöglicht sowohl im Lagerraum als auch in der Be- und Entladezone eine Lokalisierung. Nun wird eine Lokalisierung der Lastwagen außerhalb der Hallen durch eine passende Technologie mit einer Genauigkeit von 10m gesucht. Die gewonnenen Positionsdaten sollen zur Analyse der Logistikvorgänge genutzt werden und eine Grundlage zur Planung von Optimierungen darstellen. Durch eine bereits bestehende Netzwerkinfrastruktur gibt es innerhalb des für die Lokalisierung zu betrachtenden Geländes Zugriff auf das firmeneigene WLAN.

3.2 Anforderungsanalyse

Die elementarste Anforderung an das Hybridsystem ist das Erzeugen von Positionsdaten auf dem gesamten zu betrachtenden Firmengelände. Die minimale Genauigkeit ist auf zehn Meter definiert. Bezüglich der Frequenz der Positionsaktualisierungen gibt es keine Festlegung. Die zu lokalisierenden Objekte sind Lastkraftwagen in verschiedenen Größen, welche sich mit maximal 30km/h über hauptsächlich gerade Strecken bewegen. Auf dem Parkplatz sowie vor und in Ladezonen sind Stillstände zu erwarten. Die Bewegungen der Lastkraftwagen sind schwerfällig, Wenderadien sind mehrere Meter groß und sowohl Beschleunigungs- als auch Bremswege sind lang. Eine hochfrequente, energieintensive Ortung ist daher abzulehnen. Bei einer Frequenz von 0,2Hz, also einer Position je fünf Sekunden, ist der maximale Abstand zwischen den realen Positionen 41,66m. Bezogen auf das Firmengelände lässt sich bei dieser Frequenz der Zustand des Lastkraftwagens, also ob zum Beispiel fahrend, wartend oder parkend, stets zuverlässig herleiten. Auch über das Fahrverhalten lassen sich mit dieser Frequenz Aussagen treffen. Eine hohe Frequenz bietet im Vergleich keinen zusätzlichen Informationsgewinn. Bei niedriger Frequenz wird zudem weniger Energie verbraucht, womit die Sensorik im Betrieb kostengünstiger ist. Wenn die Lokalisierungslösung auf die komplette Fahrzeugflotte angewandt wird, multiplizieren sich die Kosten. Eine hohe Frequenz kann also zu unnötigen Kosten und hoher Netzwerkbelastung führen. Daher soll für diese Betrachtung eine Frequenz von 0,2 Hz festgelegt werden. Allerdings impliziert dies als Voraussetzung, dass einem Endnutzer eine Position aller fünf Sekunden zum Verwerten zur Verfügung steht. Bei einer hochfrequenten Ortung kann das Ausbleiben einer Position durch die hohe Informationsdichte kompensiert werden, bei niedriger Frequenz ist dies nicht möglich.

Wenn die hochgenauen Positionsdaten vom RTLS innerhalb der Lagergebäude vorliegen, sollen diese gegenüber denen der potentiell ungenaueren, ergänzenden Lokalisierungstechnologie bevorzugt werden. Dies betrifft hauptsächlich den Be- und Entladebereich der Lagerhäuser, bei dem potentiell beide Arten von Daten für die selben Objekte vorliegen. Die Priorität der LP soll somit definiert sein. Zusätzlich ist aus wirtschaftlicher Perspektive eine Nutzung bereits vorhandener Soft- und Hardware zu präferieren, da es den Implementationsaufwand limitiert, die Konzeptumsetzung vereinfacht und entstehende Kosten verringern kann.

Für die Übersichtlichkeit und bessere Referenzierbarkeit soll in Tabelle 3.2.1 die Anforderun-

gen und ihre Klassifizierung festgehalten werden. Hierbei wird in Erfolgs- und Qualitätskriterien unterschieden. Erfolgskriterien müssen für eine erfolgreiche Implementierung erfüllt sein. Das Erfüllen von Qualitätskriterien ist ebenfalls Ziel dieser Ausarbeitung, allerdings können diese mit entsprechender Begründung auch nur teilweise erfüllt werden.

Nr.	Kategorie	Kriterium
1	Erfolg	Positionsdaten müssen vorhanden sein
2	Erfolg	Die Genauigkeit beträgt zehn Meter oder weniger
3	Qualität (hohe Priorität)	Die Positionsfrequenz unterschreitet nicht 0,2Hz
4	Qualität (niedrige Priorität)	UWB-Positionen sollen präferenziert werden
5	Qualität (niedrige Priorität)	Vorhandene Infra-/Netzwerkstruktur soll genutzt werden

Tabelle 3.2.1: Tabellarische Darstellung der Anforderungen an das Konzept

3.3 Konzept

Um ein geeignetes Konzept für das vorangegangene Industriebeispiel vorzuschlagen, wird zuerst die existierende Systemstruktur auf Kompatibilität mit den omlox Standards geprüft. Wenn das bestehende System standardkonform erweitert werden kann, soll das Konzept gemäß der Anforderungen auf der bereitgestellten Grundlage aufbauen. Sollte keine Erweiterung möglich sein, so müssen Alternativen vorgeschlagen und bewertet werden.

3.3.1 Grundlegende Systemarchitektur

Im vorliegenden System wird als omlox Hub Implementierung die Software „DeepHub“ der deutschen Firma Flowcate verwendet. Der DeepHub realisiert laut Dokumentation alle Spezifikationen des omlox v1.2 Hub Standards, erweitert um einige Schnittstellen der noch zu veröffentlichen Hub v2.0 Spezifikation. Der DeepHub implementiert somit alle nötigen Funktionen und kann in dieser Betrachtung als omlox Hub genutzt werden. Zusätzlich festzuhalten ist, dass der Hub lokal im Firmennetzwerk installiert wurde. Dadurch liegt die Datenhoheit beim netzwerkbetreiber und potentielle Bedenken bezüglich der sensiblen Positionsdaten und einer Cloudlösung können zerstreut werden.

Auch die bestehende RTLS-Lösung der ZIGPOS GmbH für die Core Zone soll eingeschätzt werden. Es ist festzuhalten, dass das RTLS als alleinstehende Lösung gesehen werden kann. Die Softwarelösung implementiert Schnittstellen für eine eigene grafische Oberfläche sowie Logik zur Berechnung, Darstellung und Speicherung der Position von mehreren tausend Mobilgeräten. Die subsysteminterne Kommunikation der heterogenen Komponenten basiert auf MQTT und REST-API-Anfragen. Durch ein MW wird der Datenstrom zum DeepHub verwaltet, es werden Positionsdaten umgewandelt und weitergeleitet. Das RTLS unterstützt alle Spezifikationen des omlox v2 Core Zone Standards und soll übernommen werden.

Durch die Verwendung der bestehenden Systemkomponenten verbleibt für die Implementierung nur der Entwurf der Complementary Zone. Die Konzeptionierung und Verwendung einer MW, analog zur Architektur des ZIGPOS RTLS, ermöglicht das separate Auswerten und Aufbereiten von Sensordaten. Diese müssen somit nicht auf den Sensorclients bearbeitet werden. Die MW besteht daher aus drei Teilen. Zum einen gibt es die Schnittstelle zum DeepHub. Dieser bietet ein REST-Interface an, welches von der MW zum Datentransfer genutzt werden soll. Eine Lösung auf der Grundlage von HTTP ist somit notwendig. Der zweite Teil der zu entwickelnden MW ist die Schnittstelle zu den Sensoren. Die Art dieser Schnittstelle ist sensorabhängig und wird nachfolgend genauer diskutiert. Der dritte Teil der MW ist die Aufbereitung der gelieferten Sensordaten. Auch dies ist stark von der gewählten Sensorik abhängig. Allerdings ist die Funktionsweise eingrenzbar. So ist beispielsweise von einer internen Datenhaltung abzusehen. Die MW soll ausschließlich eingehende Daten aufnehmen, die gemessenen Werte umrechnen und in passender Form an den DeepHub weiterreichen können.

Um das Konzept weiter ausbauen zu können, soll im folgenden drei potentielle Sensorlösungen skizziert werden. Basierend auf den etablierten Kriterien wird anschließend das beste Konzept ausgewählt und ein Gesamtkonzept für die Complementary Zone erstellt.

3.3.2 Positionsbestimmung mit Mobilfunk

Für dieses Konzept wird eine Hardwarelösung mit Mobilfunkfähigkeit benötigt. Das Sensorboard braucht eine Möglichkeit, eine SIM-Karte zu nutzen. Eine eSIM ist hinreichend. Da die notwendige Datenrate bei einer Position je fünf Sekunden vergleichsweise klein ist, kann man frei aus den in Abschnitt 2.4.4 beschriebenen Möglichkeiten wählen. Zu bevorzugen ist LTE-CAT-M1, da dieses explizit für IoT-Anwendungen reserviert wurde. Zudem ist der Funkablauf bezüglich des Energieverbrauchs optimiert. Da mit Verwendung des Mobilfunks gleichzeitig eine Möglichkeit zum Datentransfer über das Internet besteht, muss die zu entwickelnde MW auch durch dieses erreichbar sein. Dies kann erreicht werden, indem eine entsprechende Schnittstelle öffentlich erreichbar und durch gängige Sicherheitsmechanismen, wie eine Authentifizierung, abgesichert ist oder durch die Nutzung einer VPN-Lösung. Eine Verbindung durch eine VPN Lösung ist zu bevorzugen, da der Datenstrom über das Internet zwingend für einen Erhalt der Vertraulichkeit verschlüsselt werden muss. VPN-Lösungen unterstützen dies grundsätzlich.

Bezüglich der zu verwendenden Lokalisierungstechnologie muss man die Nutzung der Cell-ID und TA ausschließen. Durch die ländliche Position des Firmengeländes ist eine Genauigkeit von 100m zu erwarten. Die notwendige Genauigkeit ist somit nicht zu erreichen, womit nur eine Positionsbestimmung mittels EOTD verbleibt. Wie in der Hub Spezifikation festgelegt, soll für dieses Konzept das Firmengelände als Zone für Mobilfunk-LP deklariert werden, was direkt über die graphische Schnittstelle des DeepHub möglich ist. Zur Übersicht und einem einfachen optischen Vergleich ist dieses Konzept in Abbildung 3.3.1 dargestellt.

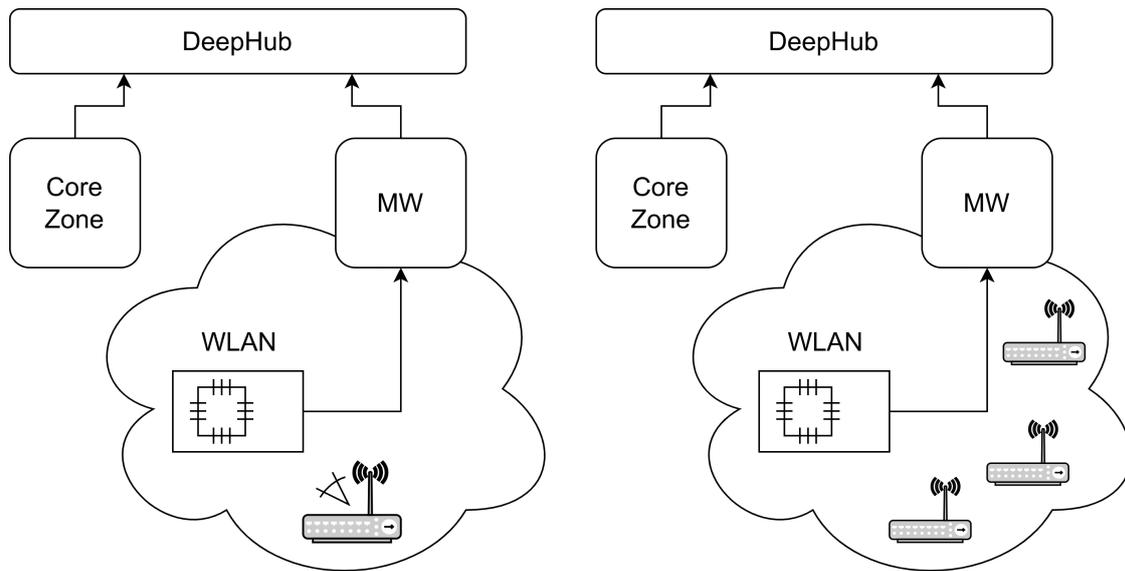


Abbildung 3.3.2: Schematische Darstellung der Grobarchitektur von zwei WiPS-Lösungen, links unter Nutzung von RSS und AoA, rechts mit DL-TDoA

3.3.4 Positionsbestimmung mit GNSS

Anders als die vorangehenden Konzepte benötigt das auf GNSS basierende Konzept keine Definition einer entsprechenden Zone im DeepHub, da jede Fläche als Teil einer Geopositionszone angenommen wird. Bei Betrachtung der notwendigen Hardware für den mobilen Sensor fällt auf, dass die zu nutzende Technologie zur Positionsermittlung keine Möglichkeit zur freien Datenübertragung hat. Das Board muss also neben einem Chip zur Ortung mittels GNSS zusätzlich einen Chip für eine Funktechnologie bieten, welche eine frei richtbare Datenübermittlung ermöglicht. Da die vorliegende Infrastruktur genutzt werden soll, bietet sich eine Verwendung des etablierten WLAN an. Der auszuwählende Board muss also seine eigene Position per GNSS ermitteln und die gewonnenen Daten dann durch das kabellose Firmennetz an die MW schicken können.

Anders als bei den vorhergehenden Konzepten ist bei der Datenverarbeitung der MW keine komplexe Umrechnung der Positionsdaten aufgrund eines lokalen Koordinatensystems notwendig. Da die Ortung mittels GNSS globale Koordinaten erzeugt, müssen diese nur für die Übergabe an den DeepHub umgeformt werden. Die Komplexität der MW beschränkt sich damit auf das Kombinieren der Sensoridentität, zum Beispiel der MAC-Adresse mit den gesendeten Positionsdaten in die vom DeepHub verlangte Form. Für die Übersicht ist das Grobkonzept in Abbildung 3.3.3 noch abschließend skizziert.

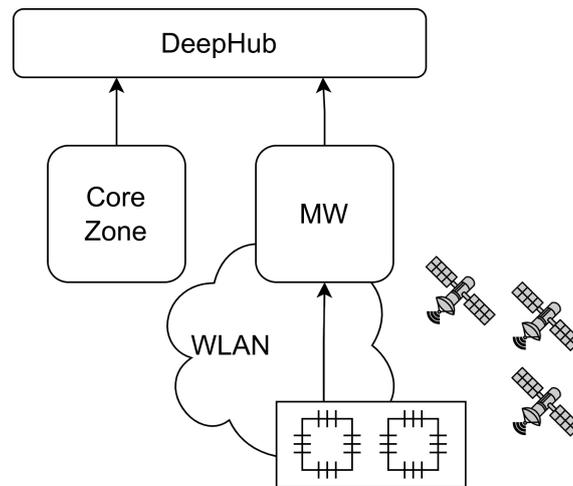


Abbildung 3.3.3: Schematische Darstellung der Grobarchitektur einer GNSS-Lösung

3.3.5 Diskussion und Gesamtkonzept

Das in Abschnitt 3.3.2 dargestellte Grobkonzept, basierend auf Mobilfunktechnologie, ist unabhängig von der Dimensionierung des Firmengeländes. Die Grenzen der Positionen werden nicht von der Mobilfunktechnologie definiert, sondern ausschließlich durch den DeepHub softwarebasiert auf die definierte Zone begrenzt. Zudem ist der benötigte Technologiestack auf dem auszuwählenden Sensorboard klein, weit verbreitet und bei Nutzung von LTE-CAT-M1 für IoT-Anwendungen optimiert. Beschränkt wird dieses Konzept durch den Implementierungsaufwand. Jedes Board benötigt eine separate Registrierung im Mobilfunknetz mittels einer SIM-Karte. Zudem ist durch die notwendige Verschlüsselung pro Gerät ein Schlüsselpaar anzulegen und für die MW erreichbar zu verwalten. Der Aufwand bei einer durchaus erwartbaren Erhöhung der Sensorenanzahl ist somit nicht vernachlässigbar. Zusätzlich wird das Konzept auch von der Lage des Firmengeländes limitiert. Durch die Ferne zu städtischer Infrastruktur ist keine ausreichend hohe Anzahl an Sendestationen gegeben, die Analyse der ToF liefert nicht die notwendige Genauigkeit.

Das WiPS-Konzept verlangt hinsichtlich des Sensorboards ebenfalls einen kleinen, sehr populären Technologiestack. Zudem führt das Konzept keine neuen Funktechnologien ein, womit die Wartbarkeit durch die bereits bestehende Netzwerkadministration begünstigt wird. Die Genauigkeitsanforderungen werden von beiden Teilkonzepten erfüllt. Bei der Überprüfung der Umsetzbarkeit der Konzepte zeigt sich, dass die verwendete WLAN-Lösung kein AoA unterstützt. Zudem zeigt die bei einer Verfügbarkeitsabschätzung entstandene Heatmap, dargestellt in Abbildung 3.3.4, dass die Signalstärke starken Schwankungen unterliegt. Zur Signalsimulation und Grafikerstellung wurde ein firmeninternes Werkzeug der ZIGPOS GmbH genutzt.

Die Simulation veranschaulicht, dass südlich des Parkplatzes ist ein stark abgeschwächtes Signal zu erwarten ist. Zwischen den Lagergebäuden ist das Signal ausreichend stark. Die Gebiete niedri-

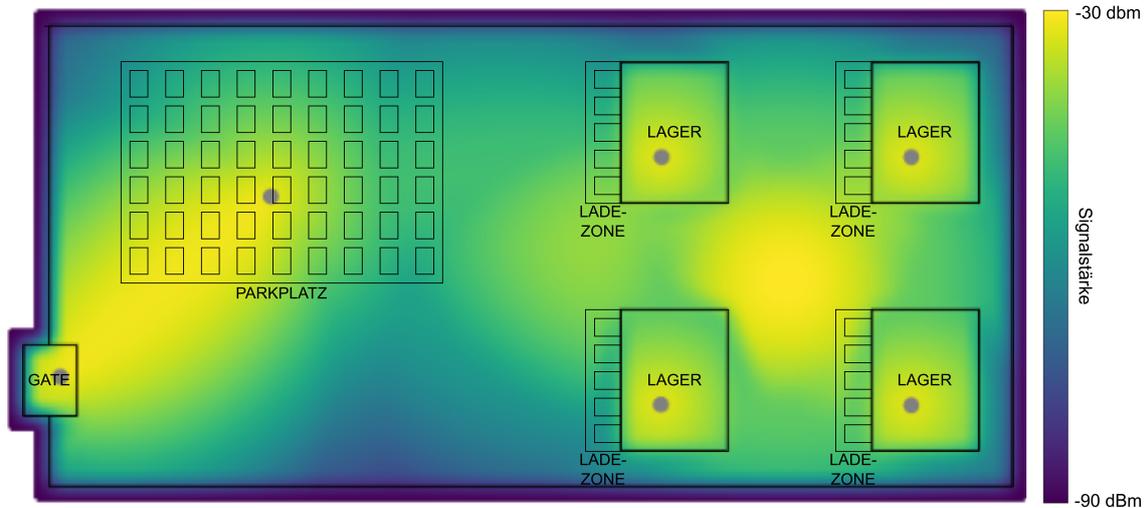


Abbildung 3.3.4: Simulierte Heatmap zur WLAN-Verfügbarkeitsabschätzung für den industriellen Anwendungsfall

ger Signalstärke zeigen auf, dass das alternative TDoA-Konzept eine höhere AP-Dichte erfordert. Dies wiederum erfordert eine Erweiterung des kabelgebundenen Netzes um entsprechende AP, Switches und Kabelverbindungen. Kostenintensive Baumaßnahmen werden notwendig.

Um das Signalverhalten detaillierter abschätzen zu können, werden der Simulation zusätzliche kreuzförmige Hindernisse hinzugefügt, welche die Lastkraftwagen darstellen sollen. Abbildung 3.3.5 zeigt die entstandene Heatmap. Die zu erwartende Signalstärke wird stark von den Lastkraftwagen beeinflusst. Beispielsweise ist bei der angegebenen Lastkraftwagenanordnung an der östlichen Seite des Parkplatzes ein vollständiger Signalverlust zu erwarten. Mit der gegebenen WLAN-Verfügbarkeit ist eine Positionsermittlung mittels WiPS nicht in der geforderten Qualität umsetzbar.

Das vorgestellte GNSS-Konzept ist bezüglich der Positionsbestimmung komplett unabhängig von der vorliegenden Infrastruktur. Zudem ist die Genauigkeit hinreichend und der Berechnungsaufwand durch die MW aufgrund des Vorliegens der geographischen Koordinaten geringer als bei den bereits beurteilten Konzepten. Zu beachten ist allerdings die höhere Komplexität des Sensorboards, da zwei verschiedene Technologien kombiniert werden müssen. Zudem wird das GNSS-Konzept auch durch die bereits dargelegte WLAN-Verfügbarkeit beeinträchtigt. Zwar werden Positionen zuverlässig ermittelt, aber eine Kommunikation innerhalb des WLAN und damit mit der MW ist nicht permanent im gesamten Betriebsgelände gegeben. Die daraus entstehenden zu erwartenden Fehler müssen behandelt werden.

Im Folgenden wird nur das GNSS-Konzept betrachtet. Die Mobilfunklösung ist aufgrund der Lage nicht ausreichend genau, die WiPS-Lösung erfordert kostenintensive und invasive Eingriffe in

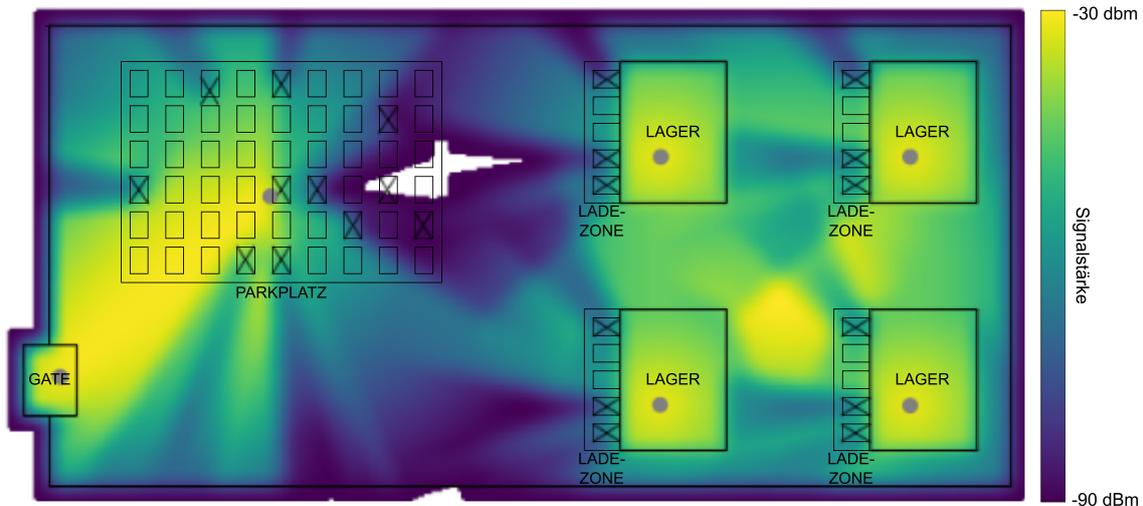


Abbildung 3.3.5: Heatmap zur WLAN-Verfügbarkeitsabschätzung für den industriellen Anwendungsfall mit Lastkraftwagen

die etablierte Hard- und Softwarestruktur. Das Konzept erfordert den Austausch aller relevanten WLAN-AP auf dem gesamten Firmengelände sowie einen hohen administrativen Aufwand bei der Installation und Wiederaufbau des Netzwerkes. Obwohl das GNSS-Konzept einen vergleichsweise hohen Implementationsaufwand aufzeigt, besonders hinsichtlich der Sensorlogik, ist die vorgeschlagene Lösung hinreichend genau, gut skalierbar und zudem modular an die bestehende Systemlandschaft integrierbar.

Weiterführend soll diskutiert werden, wie die erzeugten Daten vom Messgerät zur MW gelangen und wie eine Differenzierung der Datenquelle möglich ist. Dazu soll als erstes eine vereinfachte Architektur der Core Zone betrachtet werden, dargestellt in Abbildung 3.3.6. Das RTLS der ZIGPOS GmbH besteht aus einem zentralen MQTT-Broker. Auf diesen Broker werden die Sensordaten aus den UWB-Messungen veröffentlicht. Durch entsprechende Subscriptions auf die MQTT-Topics bezieht der RTLS-Master die für ihn relevanten Rohinformationen. Der RTLS-Master nutzt diese für die Verwaltung des omlox Netzwerkes, die allgemeine Geräteverwaltung und die Positionsalgorithmen. Die aufbereiteten Positionsdaten werden in das zentrale MQTT-Netzwerk unter Nutzung einer weiteren Topic eingespeist. Diese werden von der ZIGPOS omlox Bridge konsumiert und in standardisierter Form an den DeepHub übergeben.

Der bereits etablierte, zentrale und gut skalierbare MQTT-Broker wird für die Umsetzung des GNSS-Konzeptes genutzt. Analog zum Vorgehen der ZIGPOS GmbH könne die GNSS-Daten auf den MQTT-Broker geleitet und dadurch der MW verfügbar gemacht werden. MQTT begünstigt asynchrones Arbeiten, potentielle Verbindungsverluste können mit der „keep-alive“ Einstellung durch den Broker mitigiert werden und die Nachrichten erlauben ein kurzzeitiges Halten. Durch

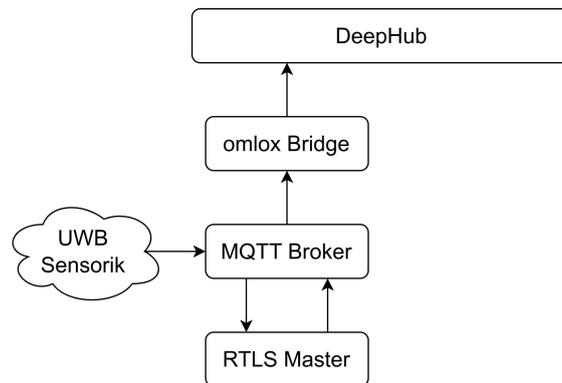


Abbildung 3.3.6: Vereinfachte Darstellung der Systemarchitektur der bestehenden omlox Core Zone von ZIGPOS

die QoS-Einstellungen wird die Problematik einer zu erwartenden kurzzeitigen WLAN Unverfügbarkeit gelöst, da sichergestellt werden kann, dass die Positionsdaten nachträglich gesendet werden können. Zudem ist es möglich, jedem Board eine separate Topic zuzuweisen, beispielsweise „gnss/board-macexample1“. Die MW kann mit entsprechender Subscription auf alle mit „gps/board-“ beginnenden Themen reagieren und durch die Bezeichnung der Topic die Nachrichtendaten einem Board zuordnen. Das erlaubt auch bei entsprechender Skalierung eine feinere Gliederung der Boards, zum Beispiel nach lokalisiertem Objekt. In Abbildung 3.3.7 ist die entwickelte Systemarchitektur dargestellt. Die Architektur des RTLS der ZIGPOS GmbH wurde für eine bessere Übersicht auf die für diese Betrachtung relevanten Komponenten reduziert.

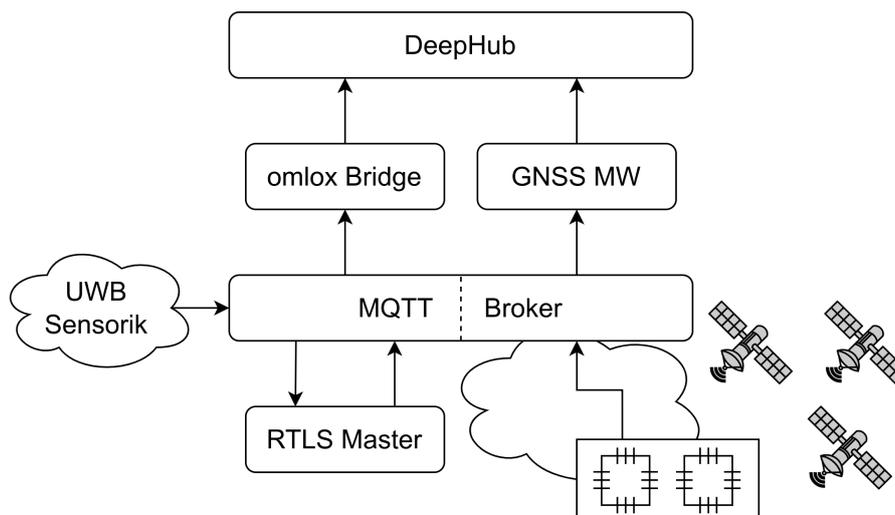


Abbildung 3.3.7: Schematische Darstellung der Systemarchitektur der GNSS-Lösung

Kapitel 4

Prototypische Implementierung

4.1 GNSS Sensor

Basierend auf der vorangehend beschriebenen Boardanforderung wurde für die prototypische Umsetzung in dieser Arbeit das C099-F9P⁷ von u-blox ausgewählt. Das C099-F9P ist ein Application Board für das ZED-F9P⁸ GNSS-Modul, folgend als ZED bezeichnet, und das ODIN-W2⁹ Kurzstreckenmodul, folgend als ODIN betitelt. Der ZED ist ein Multi-Band GNSS Empfänger und unterstützt alle gängigen Satellitensysteme. Zudem werden Optionen zur Anwendung als und mit einem RTK angeboten. Als Antenne wird von u-blox die ANN-MB Serie empfohlen, für diese Arbeit wird eine ANN-MB-00-00 zur Verfügung gestellt. Der ODIN bietet BLE und WiFi Funktionen mit Hilfe einer externen, am Bord anschließbaren Stabantenne und ist durch den AT-Befehlssatz konfigurierbar. Abbildung 4.1.1 zeigt das genutzte Application Board mit den wichtigsten Komponenten.

Für die Konfiguration der Module stellt u-blox entsprechende Software zum Download bereit. Für den ZED ist das u-center¹⁰ vorgesehen, der ODIN soll mit Hilfe von s-center¹¹ konfiguriert werden. Beide Programme sind nur für Windows verfügbar. Bei Anschluss an den für die Konfiguration genutzten PC werden drei neue Einträge im Gerätemanager verfügbar: ZED, ODIN und das Arduino Application-Board. Die Module besitzen ein Command Line Interface (CLI), welche die bereits erwähnte u-blox Software zur Interaktion nutzt. Ein erster Test zeigt, dass der ZED sofort nach Aufbau Positionsdaten erfasst und in Form von National Marine Electronics Association (NMEA)-Nachrichten ausgibt. Problematisch hingegen ist die Konfiguration des ODIN, denn dieser besitzt zwar die Möglichkeiten für eine Verbindung zum WLAN, dies wird aber nicht in dem benötigten Maße von der vorinstallierten Firmware unterstützt. Im Lieferzustand unterstützt der ODIN nur ein Rover-Base-Szenario, also die Kommunikation in einem kabellosem Netzwerk mit

⁷ <https://u-blox.com/en/product/c099-f9p-application-board>, zuletzt aufgerufen am 7.6.2023

⁸ Website: <https://u-blox.com/en/product/zed-f9p-module>, zuletzt aufgerufen am 7.6.2023

⁹ Website: <https://u-blox.com/en/product/odin-w2-series-u-connect>, zuletzt aufgerufen am 7.6.2023

¹⁰ Website: <https://u-blox.com/en/product/u-center>, zuletzt aufgerufen am 7.6.2023

¹¹ Website: <https://u-blox.com/en/product/s-center>, zuletzt aufgerufen am 7.6.2023

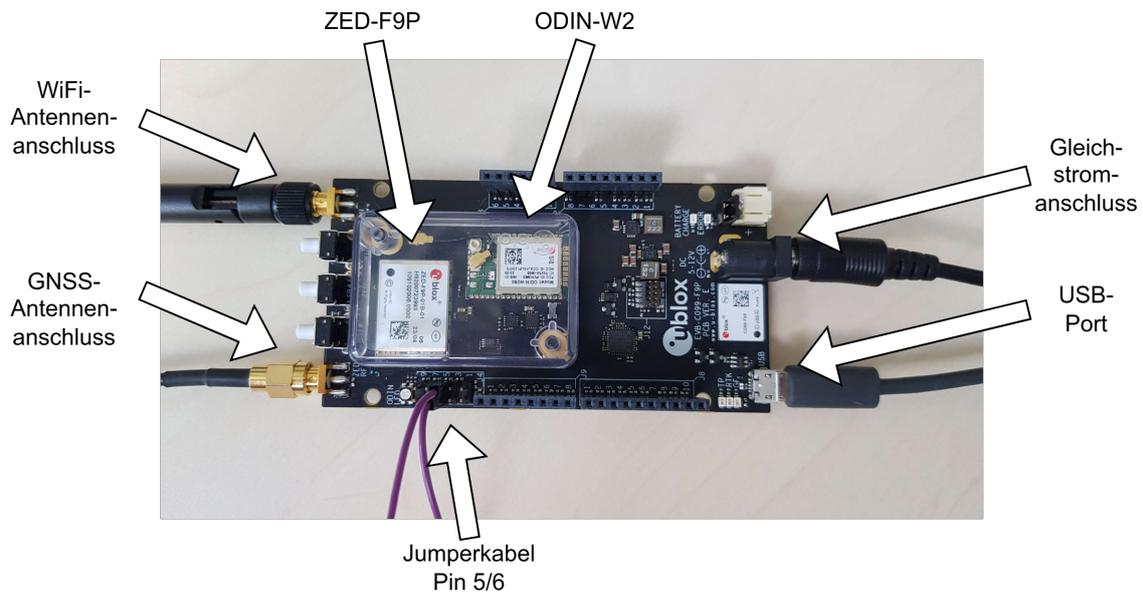


Abbildung 4.1.1: Foto des C099-F9P Application Boards mit Beschriftung der wichtigsten Komponenten

zwei C099-F9P Boards. Auch eine Kommunikation mittels MQTT ist nicht vorgesehen. Daher ist es erforderlich, den ODIN mit alternativer Firmware zu flashen. Die offizielle Boarddokumentation¹² von u-blox bietet in Abschnitt 7.2 Unterstützung bei diesem Vorhaben.

Anstatt der Mbed OS 3 Application Firmware wird die angebotene u-connectXpress Software installiert. Dies setzt das Flashen eines Bootloaders voraus, was mit dem vorgeschlagenen Werkzeug `stm32flash` in der Version 0.7¹³ umgesetzt werden kann, wie Codeblock 4.1.1 zeigt. Als Bootloader wird die Binary „ODIN-W2-BOOT-v0.8.2.bin“ von u-blox genutzt. Durch einen Neustart und dem Verbinden der Pins 9 und 10 auf dem Application Board wird der ODIN in den SafeBoot-Modus versetzt.

Als Firmwaredatei wird die offizielle u-connectXpress-Binärdatei in Version 8.2.1 aus dem u-blox Downloadportal genutzt. Das Flashen der Software mit `stm32flash` bleibt erfolglos, das Modul lässt sich nach abgeschlossenem Schreibvorgang nicht nutzen. Daher wird, anders als in der Dokumentation beschrieben, `s-center` zum Aufspielen der Software genutzt¹⁴. `S-center` ermöglicht eine Installation der Firmware bei einer Baudrate von 115200 ohne Safeboot-Modus. Mit Abschluss des Aufspielvorgangs zeigt die Leuchtdiode einen Normalbetrieb an und übergebene Befehle, in

¹² Einsehbar unter https://content.u-blox.com/sites/default/files/documents/C099-F9P-AppBoard_UserGuide_UBX-18063024.pdf, zuletzt aufgerufen am 7.6.2023

¹³ Herunterladbar unter <https://sourceforge.net/projects/stm32flash/>, zuletzt aufgerufen am 7.6.2023

¹⁴ Basierend auf einem Foreumeintrag, <https://portal.u-blox.com/s/question/0D52p0000AoAVVeCQ0/no-communication-with-odinw2-and-scenter-on-c099f9p>, zuletzt aufgerufen am 7.6.2023

Quellcode 4.1.1: Powershellausgabe bei Flashen des Bootloaders

```

1 PS C:\Users\Nicolas\Desktop> .\stm32flash.exe -b 115200 -2
   \ODIN-W2-BOOT-v0.8.2.bin -S 0x8000000 COM4
2 stm32flash 0.7
3
4 http://stm32flash.sourceforge.net/
5
6 Using Parser      : Raw BINARY
7 Location          : 0x8000000
8 Size              : 10516
9 Interface serial -w32: 115200 8E1
10 Version           : 0x31
11 Option 1          : 0x00
12 Option 2          : 0x00
13 Device ID         : 0x0419 (STM32F42xxx/43xxx)
14 - RAM              : Up to 192kiB (12288b reserved by bootloader)
15 - Flash            : Up to 2048kiB (size first sector: 1x16384)
16 - Option RAM      : 65552b
17 - System RAM      : 30kiB
18 Write to memory
19 Erasing memory
20 Wrote address 0x0812c256 (100.00%) Done.

```

Abbildung 4.1.2 demonstriert, werden erfolgreich ausgeführt. Die Abbildung zeigt den eingegebenen Befehl, eingeleitet von „AT+UDCP=“, und die darauf folgende Ausgabe „+UDCP:1“, „OK“ und einer weiteren automatischen Speichersetzung.

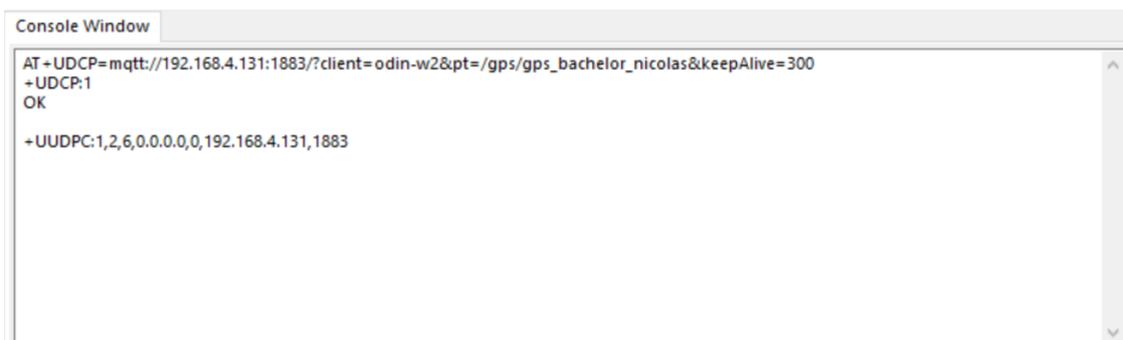


Abbildung 4.1.2: Konsole von s-center nach Ausführung des Befehls zur Registrierung einer MQTT-Verbindung

Für den Betrieb des ODIN sind für diese Arbeit zwei Modi relevant. Der Command-Modus, auch AT-Modus genannt, erlaubt das Setzen von Variablen durch die Nutzung der namensgebenden AT-Befehle. In diesem Modus wird die derzeitige Konfiguration geändert, welche aber auch persistent in den Flashspeicher übernommen werden kann und so nach einem Neustart weiterhin zur Verfügung steht. Der Data-Modus ermöglicht aktive Verbindungen zu anderen Endpunkten. Zu

übermittelnde Daten werden automatisch in Pakete unterteilt und diese werden entsprechend der relevanten Übertragungsprotokolle angepasst. Das Application Board wird im factory default in den Command-Modus gebootet. Sollte in diesem der Befehl „AT+UMSM=1“ ausgeführt werden, wird zukünftig in den Data-Modus gebootet. Es ist möglich, zur Laufzeit zwischen den Modi zu wechseln. Von Command-Modus zu Data-Modus geschieht dies durch die Eingabe des „AT0“ Befehls, von Data-Modus zu AT-Modus ist die Escape-Sequenz oder das Setzen des UART DTR-Pins von High auf Low. Abbildung 4.1.3 stellt diesen Sachverhalt zur Übersicht als Zustandsdiagramm dar.

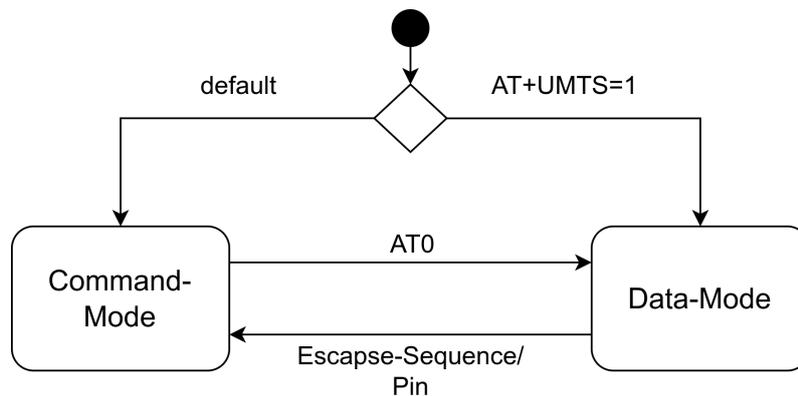


Abbildung 4.1.3: Zustandsdiagramm für den Wechsel zwischen Command-Modus und Data-Modus des Odin-W2

Im AT-Modus werden zuerst die notwendigen Informationen für die Verbindung zum WLAN übergeben. Durch die graphische Oberfläche ist eine direkte Eingabe durch AT-Befehle nicht notwendig, stattdessen werden Informationen wie beispielsweise SSID, Verschlüsselung und Passwort direkt in der WiFi-Station-Maske eingetragen und anschließend über einen Button in entsprechende AT-Befehle konvertiert und übergeben. Bei Erfolg verbindet sich der ODIN direkt mit dem ausgewählten Netzwerk und das „Network Status“-Feld in der Programmoberfläche aktualisiert sich. Eine Einsicht in die DHCP-Einträge des gewählten Netzwerkes bestätigt die erfolgreiche Anmeldung.

Mit dem erfolgreichen Einbinden des ODIN in das WLAN liegt folgend der Betrachtungsschwerpunkt beim Publishen der NMEA-Daten. Zu Testzwecken wird hierfür ein mosquitto MQTT-Broker¹⁵ eingerichtet. Im IoT-Tab der s-center Oberfläche besteht die Möglichkeit, eine MQTT-Verbindung anzugeben. Bei bestehender Verbindung werden im DATA-Modus des ODIN alle Nachrichten an den MQTT-Broker geleitet. Die Dokumentation des C099-F9P zeigt, dass ZED und ODIN durch einen Universal Asynchronous Receiver / Transmitter (UART) Kanal verbunden sind. Abbildung 4.1.4 zeigt den vereinfachten Kanalaufbau des Application Boards.

¹⁵ Website: <https://mosquitto.org/>, zuletzt aufgerufen am 23.7.2023

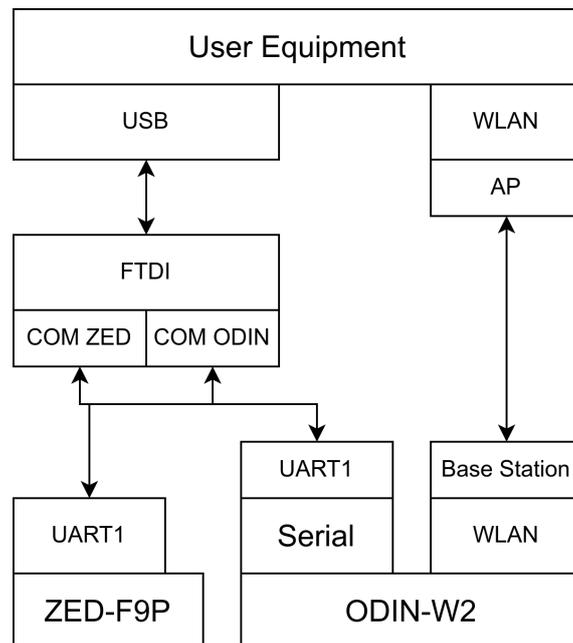


Abbildung 4.1.4: Blockdiagramm der Architektur des C099-F9P Application Boards, reduziert auf aktiv genutzte Komponenten und Verbindungen

Durch die UART Verbindung kann der ZED GNSS Daten mittels NMEA-Nachrichten an den ODIN übergeben, welche dann im Netzwerk übertragen werden können. Da die Verbindung aber physikalisch getrennt ist, muss der UART-Kanal erst mit einem Jumperkabel an PIN 5 und 6 des Application Boards geöffnet werden. Bei einem Testlauf gelangen alle ZED-Ausgaben an den ODIN und damit an den eingerichteten MQTT-Broker. Die Nachrichtenmenge liegt je nach erkannten Satelliten bei 8 bis 15 Nachrichten pro Sekunde. Eine beispielhafter Auszug der MQTT-Kommunikation ist in Textblock 4.1.2 festgehalten. Die GNGLL-Nachricht hat für diese Arbeit die größte Relevanz, denn diese enthält bereits eine berechnete Position mit Angabe der Lat- und Longitude in Grad und Minuten sowie einem Zeitstempel.

Quellcode 4.1.2: Beispielhafter Auszug der veröffentlichten MQTT-Nachrichten

```

1 $GNVTG, ,T, ,M,0.022 ,N,0.041 ,K,A*38
2 $GNGGA,111753.00 ,5101.42977 ,N,01344.59904 ,E,1 ,12 ,0.59 ,174.8 ,M, ...
3 $GNGSA,A,3 ,02 ,08 ,10 ,21 ,03 ,32 ,27 ,14 , , , , ,1.06 ,0.59 ,0.88 ,1*00
4 $GNGSA,A,3 ,70 ,69 ,84 ,76 ,85 ,68 , , , , , ,1.06 ,0.59 ,0.88 ,2*0
5 $GNVTG, ,T, ,M,0.017 ,
6 GBGVS,1 ,1 ,04 ,06 ,21 ,074 ,24 ,11 ,54 ,300 ,42 ,12 ,66 ,142 ,50 ,16 ,22 ,073 ,27 , ...
7 $GBGVS,2 ,1 ,05 ,02 ,05 ,104 , ,05 ,20 ,127 , ,13 ,09 ,070 , ,19 ,01 ,040 , ,0*71
8 $GBGVS,2 ,2 ,05 ,19 ,01 ,040 , ,0*4F
9 $GNGLL,5101.42978 ,N,01344.59904 ,E,111754.00 ,A,A*76

```

Um eine bessere Skalierung und mehr Übersichtlichkeit zu ermöglichen, wird die Nachrichtenlast an ODIN und Broker reduziert. Ein erster Ansatz, um die Ausgabe seitens des ZEDs einzuschränken, ist die Reduktion der akzeptierten Satelliten. Hierfür wird in der Konfiguration festgelegt, dass ausschließlich Satelliten des GPS und des Galileo Systems genutzt werden. Dies reduziert den Ausgabeblock des ZED auf die Hälfte, ohne die Genauigkeit der bestimmten Position zu beeinflussen. Um die Nachrichtenmenge weiter zu reduzieren, kann die Ausgabe bestimmter NMEA-Nachrichten auf den UART-Kanal in den Einstellungen untersagt werden. Für diese Arbeit sollen alle GNVGT (Kurs- und Geschwindigkeitsinformationen der Satelliten), GNGGA (GPS Daten), GNGSA (Operationsmodus und Informationen über die genutzten Satelliten) und GNGSV (Informationen zu Satelliten im Sichtfeld) Nachrichten nicht ausgegeben werden. Nach Deaktivierung verbleibt nur die detaillierte GNGLL-Nachricht, welche sekundlich gesendet wird. Die Funktionsweise des ZED wird dabei nicht beeinträchtigt, da die Änderungen an den ausgegebenen Nachrichten ausschließlich Nachrichtenfluss auf dem UART-Kanal betreffen.

Zusammenfassend ist festzuhalten, dass nach beschriebener Konfiguration des C099-F9P das ZED-F9P Modul eine Geoposition unter Nutzung von GPS und Galileo bestimmt und die entsprechende NMEA-Nachricht über den Boardkanal UART1 veröffentlicht. Das ODIN-W2-Modul kann diese Nachricht empfangen und publiziert diese durch die Nutzung des definierten Netzwerkes an den definierten MQTT-Endpunkt. Es handelt sich dabei um durchschnittlich eine Position pro Sekunde, angegeben im GNGLL-Format des NMEA-Standards.

4.2 Middleware

Die MW-Anwendung wird in einem Docker-Container umgesetzt, um sie besser im bestehenden System integrieren zu können. Durch die Kapselung in einem Container ist die MW unabhängig von Systemdiensten und lässt sich damit plattformunabhängig flexibel in verschiedenen Containerlaufzeitumgebungen installieren. Globale Parameter, wie die URL des Brokers und des DeepHubs, werden bei Programmstart zu übergeben.

Als Programmiersprache für die MW wird Python3 gewählt, da diese Skriptsprache durch eine Vielzahl von Bibliotheken eine einfache und schnelle Umsetzung des Konzepts ermöglicht. Zur Verwaltung der MQTT-Verbindung wird die „paho-mqtt“-Bibliothek verwendet. Diese bietet ein Client-Objekt an, welches die Verbindung zum Broker etabliert und auf elementare Events wie Disconnects oder dem Erhalten einer Nachricht durch Callback-Funktionen reagieren kann.

Quellcode 4.2.1: Beispielnachrichten der MQTT-Topic

```

1 $GNGLL,, , , 132400.00 , V, N*50
2 $GNGLL,5 101.43334 , N, 01344.59343 , E, 142414.00 , A, A*7D
3 $GNGLL,5 101.43336 , N, 01344.59341 , E, 142415.00 , A, A*7C

```

Die eingehenden Nachrichten, beispielhaft dargestellt in 4.2.1, besitzen ein grundlegendes, stan-

standardisiertes Format. Die einzelnen Einträge sind durch Kommata getrennt und enthalten nacheinander Angaben über:

- Format
- Longitude (zwei führende Stellen als Grad, dann Minuten)
- Nord- beziehungsweise Südhalbkugel
- Latitude (drei führende Stellen als Grad, dann Minuten)
- West- oder Osthalbkugel
- Zeitstempel (UTC, Stunden/Minuten/Sekunden/Sekundenbruchteil in Dezimal)
- Modus der Positionsermittlung
- Prüfsumme der Nachricht

Jede für die MW relevante Nachricht beginnt mit „\$GNGLL“, worauf diese geprüft werden. Die Zeichenkette wird an den Kommata getrennt und durch „String-Slicing“ weiter verarbeitet. Der entsprechende Algorithmus ist in Codeschnipsel 4.2.2 dargestellt.

Quellcode 4.2.2: Spaltung der Zeichenkette in die relevanten Informationen

```

1 mqtt_information_list = mqtt_string.split(",")
2 longitude_degree = int(mqtt_information_list[1][:2])
3 longitude_minutes = float(mqtt_information_list[1][2:])
4 longitude_part = mqtt_information_list[2]
5 latitude_degree = int(mqtt_information_list[3][:3])
6 latitude_minutes = float(mqtt_information_list[3][3:])
7 latitude_part = mqtt_information_list[4]
8 timestamp = mqtt_information_list[5]
```

Wird, wie in der beispielhaften Ausgabe 4.2.1 in Zeile eins dargestellt, eine unvollständige Nachricht erhalten, wirft dieser Codeabschnitt eine Exception, welche gemäß EAFP-Ansatz¹⁶ in Python3 in einem umschließenden try-catch-Block abgefangen wird. Nachrichten wie diese erscheinen verstärkt beim Start des Sensors, da zu diesem Zeitpunkt noch nicht ausreichend Informationen für eine Positionsermittlung vorliegen.

Der DeepHub bietet einen REST-Endpunkt zur Positionsaktualisierung an. Mit der „requests“-Bibliothek kann eine entsprechende Anfrage erstellt und versendet werden. Ausgenommen der Objekt-ID sind alle für den Hub notwendige Informationen im Body des Requests enthalten. Der Aufbau ist in Codeausschnitt 4.2.3 dargestellt. Die Identifikationsnummer des Objektes wird in der Request-URI übergeben.

Das hier genutzte „position“-Objekt ist triviales Konstrukt zur Haltung und Umwandlung der in Codeschnipsel 4.2.2 gewonnenen Informationen. Der Wert für „provider_type“ wird auf „gps“ gesetzt, der „crs“-Eintrag ist auf das CRS EPSG:4326, auch WSG84 genannt, festgelegt. Dieser Standard sieht die Angabe von Grad und Minuten als Dezimalzahl vor. Der Positionseintrag folgt

¹⁶ Easier to ask for forgiveness than permission., siehe <https://docs.python.org/3/glossary.html#term-EAFP>, zuletzt aufgerufen am 26.7.2023

Quellcode 4.2.3: Aufbau des Request-Bodies

```

1 json_dict = {
2     "position": {
3         "type": "Point",
4         "coordinates": [ position.lat_in_decimal(),
5                         position.long_in_decimal() ]
6     },
7     "source": position.location_provider_id,
8     "provider_type": self.provider_type,
9     "provider_id": position.location_provider_id,
10    "timestamp_generated": datetime.datetime.strptime(
11        datetime.datetime.utcnow().date().isoformat()
12        + "-"
13        + str(position.timestamp)
14        + "0",
15        "%Y-%m-%d-%H%M%S.%f").isoformat(),
16    "crs": "EPSG:4326"
17 }
```

dem GeoJSON-Standard und enthält eine Liste aus Latitude gefolgt von Longitude im WSG85-Format. Die Umrechnung von Grad/Minutendarstellung in Dezimaldarstellung erfolgt mit folgender Formel:

$$long_{dec} = long_{grad} + \frac{long_{min}}{60}$$

Sollte sich die Position auf der Südhalbkugel befinden, muss der Dezimalwert mit -1 multipliziert werden. Die Berechnung gilt analog für die Latitude, multipliziert mit -1 im Falle einer Position auf der Westhalbkugel.

Der „timestamp_generated“-Eintrag, Zeile neun bis elf, wird mit einem Zeitstempel gemäß ISO 8601 gefüllt. Da die NMEA-Nachricht des GNSS-Sensors nur die UTC-Zeit der Positionsermittlung enthält, muss das Datum manuell ergänzt werden. Zu beachten ist dabei, dass das Datum der koordinierten Weltzeit genutzt werden muss, um Fehler bei einem Datumswechsel in der lokalen Zeitzone auszuschließen.

Der „source“-Eintrag enthält die omlox Zone-ID. Da GPS als globale Zone zu verstehen ist, spezifiziert der omlox Hub Standard für diesen Eintrag eine erneute Angabe der Provider-ID. Sollte kein LP mit der übergebenen Provider-ID existieren, wird mit Absenden des PUT-Requests durch den DeepHub automatisch ein LP-Eintrag erstellt. Zur Darstellung im DeepHub ist somit nur noch das Anlegen eines Trackables nötig, welches die LP als Datenquelle angibt. In der graphischen Oberfläche ist dafür ein Trackable-Untermenü vorgesehen, in der alle Trackables verwaltet werden können. Die Trackable-spezifischen Regeln werden ebenfalls in diesem Menü durch eine Eingabemaske deklariert. Für diese Ausarbeitung wird definiert, dass aktuelle UWB-Positionen höher zu priorisieren sind als GPS-Angaben. Als Maximalalter einer akzeptierten UWB-Position wird prototypisch fünf Sekunden festgelegt.

Kapitel 5

Evaluation

5.1 Test der Implementation

Für eine Evaluation der Umsetzung wird eine Testinstanz des RTLS mit installiertem DeepHub zur Verfügung gestellt. Um Positionen außerhalb der im Testsystem eingerichteten UWB-Zone zu erzwingen, wird eine Teststrecke außerhalb der Firmenräume der ZIGPOS GmbH im Schritttempo abgelaufen. Bei dieser Teststrecke handelt es sich um Straßen zu und Spazierwege in einem öffentlichen Park. Das urbane beziehungsweise dicht bepflanzte Gelände ermöglicht eine Beurteilung der Genauigkeit der GNSS-Positionen unter nicht-optimalen Bedingungen. Als Frequenz für die Positionsermittlung des Sensors wird ein Hertz gewählt. Dies führt zu einer höheren Datenmenge für den Systemtest, wodurch eine genauere Analyse möglich wird. Anomalien und unerwartetes Verhalten sind besser erkennbar, zudem liegen mehr Daten für eine Genauigkeitsabschätzung vor.

Für die Auswertung werden zwei Trackables im DeepHub angelegt. Ein Trackable, folgend als GPS-Trackable bezeichnet, bekommt den GNSS-LP zugewiesen. Das zweite Trackable, folgend als Hybrid-Trackable betitelt, bekommt sowohl den GNSS-LP als auch den UWB-LP zugewiesen. Zusätzlich wird die Positionsregel definiert. Semantisch wird dies durch die im Codeausschnitt 5.1.1 dargestellten Eintrag umgesetzt. Während des Testlaufs werden die Positionsdaten beider Trackables durch ein Skript sekundlich beim DeepHub abgerufen und lokal aufgezeichnet.

Für den Test wird ein mobiler WiFi-Hotspot eines Mobiltelefonies genutzt. Zuerst wird ein Entwicklungslaptop in das vom Smartphone verwaltete Netz eingebunden. Auf dem Laptop läuft ein lokaler mosquito MQTT-Broker, ein Docker Container, der die entwickelte GNSS-MW beinhaltet, und ein VPN-Client, welcher eine Verbindung zum Firmennetzwerk und damit zum DeepHub des Testsystems ermöglicht. Die MW verbindet sich zum lokalen Broker und übermittelt Daten durch die gesicherte Verbindung an den DeepHub. Der Odin ist für das mobile Netzwerk konfiguriert und die MQTT-Verbindung zum Broker etabliert. Nach der Konfiguration wird das ublox-Board durch eine handelsübliche Powerbank mit Strom versorgt. Dieser Testaufbau ist nicht identisch mit dem in Kapitel 3 beschriebenen Konzept, allerdings können alle wesentlichen Kom-

Quellcode 5.1.1: Regelsatz im JSON-Format

```

1  "locating_rules":[
2    {
3      "priority": 9,
4      "expression": "(type = 'uwb') AND (timestamp_diff <= 5000)"
5    },
6    {
7      "priority": 7,
8      "expression": "type = 'gps'"
9    }
10 ]

```

ponenten und Verbindungen in dieser Form getestet werden. Zur besseren Veranschaulichung zeigt Abbildung 5.1.1 den Aufbau des Gesamtsystems.

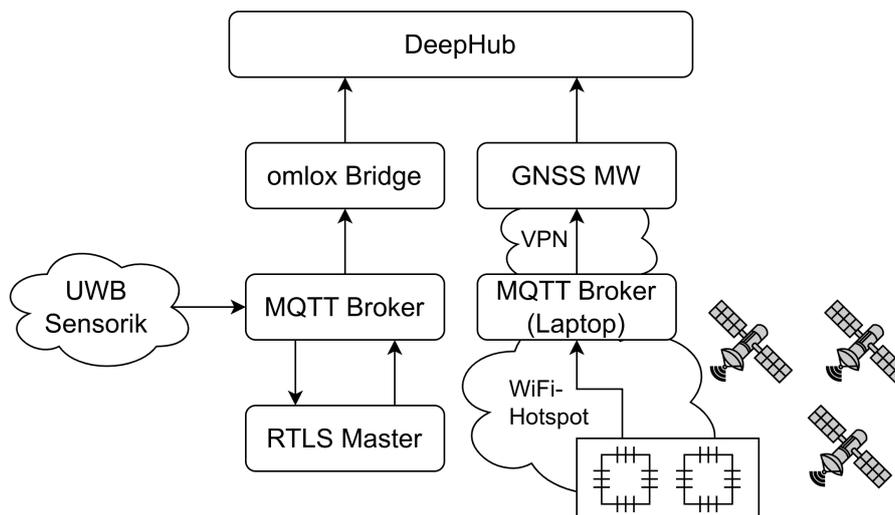


Abbildung 5.1.1: Systemarchitektur des Testsystems

Bei dem Testlauf ist zu erwarten, dass die beiden Trackables an unterschiedlichen Positionen starten. Sobald der UWB-Bereich verlassen wird, beziehungsweise keine UWB-Positionen mehr ermittelt werden können, sollten sich beide Trackables auf die selbe, von GNSS-Daten gespeiste Position beziehen. Die Positionen sollten sich gleichen, bis erneut eine UWB-Position ermittelt werden kann, womit sich die identische Lage der Trackables aufhebt. Durch die Datenerfassung des Skriptes lassen sich die Positionen nachträglich auswerten.

Nach der Testdurchführung steht ein Datensatz von 4488 Einträgen zur Verfügung, 2244 Positionen je Trackable. Eine erste Auswertung bestätigt, dass für das Hybrid-Trackable unterschiedliche Positionsquellen genutzt wurden. In den aufbereiteten Daten kann man dies an dem type-Parameter in den Testdaten erkennen. Ein Ausschnitt der Daten ist in Codeblock 5.1.2 dargestellt,

 Quellcode 5.1.2: Auszug aus den aufbereiteten Testdaten

```

1 { 'position': [13.743308, 51.023839833333334],
2   'timestamp': '2023-07-10 07:17:32.971625', 'type': 'gps' }
3 { 'position': [13.743376848086474, 51.02385576271903, 149.65],
4   'timestamp': '2023-07-10 07:17:33.979713', 'type': 'uwb' }
5 { 'position': [13.743376848086474, 51.02385576271903, 149.65],
6   'timestamp': '2023-07-10 07:17:34.987757', 'type': 'uwb' }
7 { 'position': [13.743376848086474, 51.02385576271903, 149.65],
8   'timestamp': '2023-07-10 07:17:35.997101', 'type': 'uwb' }
9 { 'position': [13.743376848086474, 51.02385576271903, 149.65],
10  'timestamp': '2023-07-10 07:17:37.006164', 'type': 'uwb' }
11 { 'position': [13.743308, 51.023839833333334],
12   'timestamp': '2023-07-10 07:17:38.014344', 'type': 'gps' }
13 { 'position': [13.74337684079372, 51.023856032427986, 149.65],
14   'timestamp': '2023-07-10 07:17:39.022263', 'type': 'uwb' }
15 { 'position': [13.74337684079372, 51.023856032427986, 149.65],
16   'timestamp': '2023-07-10 07:17:40.031221', 'type': 'uwb' }
17 { 'position': [13.74337684079372, 51.023856032427986, 149.65],
18   'timestamp': '2023-07-10 07:17:41.039199', 'type': 'uwb' }
19 { 'position': [13.74337684079372, 51.023856032427986, 149.65],
20   'timestamp': '2023-07-10 07:17:42.046461', 'type': 'uwb' }
21 { 'position': [13.743307666666666, 51.023840166666666],
22   'timestamp': '2023-07-10 07:17:43.056669', 'type': 'gps' }

```

Der Wechsel zwischen UWB und GPS ist erkennbar, und auch die Umsetzung der Positionsregel kann Zeile zwei ff. bestätigt werden. Nach einer GPS-Position werden vier identische UWB-Positionen erfasst, bis erneut eine GNSS-Information priorisiert wird. Die darauf folgenden vier identischen UWB-Positionen mit einem erneutem Wechsel zu GNSS-Informationen legt nahe, dass der physische Standpunkt zu diesem Zeitpunkt genau an der RTLS-Grenze lag und nur wenige verwertbare Positionen erfasst werden konnten.

Die erhobenen Grunddaten werden folgend für eine Visualisierung aufbereitet. Hierzu wird für jede erfasste Position der Trackables überprüft, ob das jeweils andere Trackable ebenfalls auf dieser Position war. Es werden drei Listen mit Positionen erstellt. Zwei Listen mit jeweils exklusiven Positionen für das GNSS- und das Hybrid-Trackable, eine Liste für gemeinsame Positionen. Diese Listen werden in einem passenden Dienst, für diese Arbeit wurde Google Maps ausgewählt, importiert und werden so grafisch interpretierbar. Die einzelnen Positionen werden als Punkte dargestellt und mit Farben codiert, wie in Abbildung 5.1.2 dargestellt¹⁷. Graue Punkte sind Positionen, die von beiden Trackables eingenommen werden. Blaue Punkte sind Positionen, welche exklusiv vom GNSS-Trackable eingenommen wurden, analog gilt dies für grüne Punkte und das Hybrid-Trackable. Diese Punkte treten auf, wenn für das Hybrid-Trackable aktuelle UWB-Positionen zur Verfügung stehen. Die abgebildete Strecke wurde vom Firmengebäude im Osten einmalig gegen den Uhrzeigersinn abgelaufen und mit einer Rückkehr in das Firmengelände beendet.

¹⁷ Interaktive Karte abrufbar unter <https://www.google.com/maps/d/viewer?hl=de&mid=1EY1ReUf7eTsvczFZOHTzNSw7eR8uu4U>, zuletzt aufgerufen am 12.7.2023

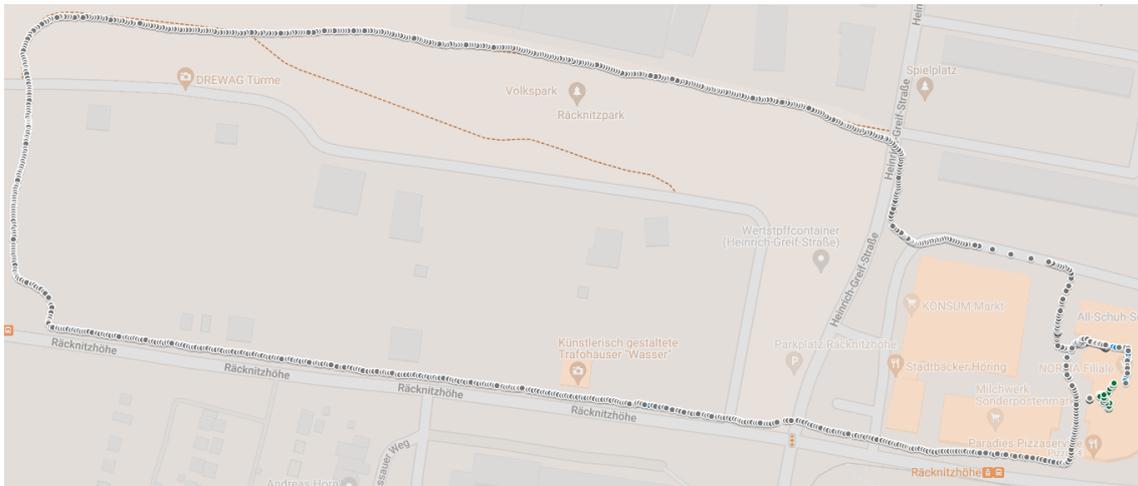


Abbildung 5.1.2: Visualisierung der im Testlauf erhobenen Positionen in Google Maps, unterschieden in die Zugehörigkeit der jeweiligen Trackables

5.2 Einschätzung der Testergebnisse

Initial entsprechen die erhobenen Daten den Erwartungen an den Test. Abseits der UWB Zone erhalten beide Trackables ihre Positionen aus dem GPS-LP. Bei genauerer Betrachtung fallen fünf Positionen außerhalb der UWB-Zone auf, die nicht für beiden Trackables aufgezeichnet wurden. In einem Fall (südlich) handelt es sich um zwei Positionen, welche direkt nebeneinander liegen. Diese können auf die Zeitdifferenz in der Positionsabfrage des Logging-Skripts beider Trackables zurück geführt werden. Zwischen dem Erfassen der Positionsdaten fand eine Aktualisierung der Position im DeepHub statt. Dieser systematische Messfehler hätte mit einer Nutzung des DeepHub-Websockets zum Verfolgen der Positionsevents anstatt eines aktivem Pollings verhindert werden können. Drei andere Positionen sind exklusiv vom GNSS-Trackable eingenommen worden. Dies ist ebenfalls auf das Logging-Verfahren zurück führen. Alternativ ist ein internes Problem im DeepHub nicht ausschließbar. Diese Situation tritt allerdings sehr selten auf und beeinflusst den Positionsverlauf im Gesamten nicht, da anschließend erneut identische Positionen eingenommen werden. Das integrierte Messwerkzeug von Google Maps erlaubt eine visuelle Abschätzung der Genauigkeit. So beträgt die maximale, zur Laufrichtung senkrechte Abweichung zum genutzten Weg drei Meter, was trotz der schwierigen Messbedingungen durch Bäume und Sträucher unter den geforderten zehn Metern liegt.

Der Grenzbereich zum RTLS, dargestellt in Abbildung 5.2.1 auf neutralem und Satellitenhintergrund, wird folgend intensiver betrachtet. In dieser Region wird das Trackable-Verhalten am Zonenrand besonders gut bewertbar.

Wie die Abbildung darstellt, gibt es keine GNSS-Positionen innerhalb des Gebäudes. Bei Betrachtung

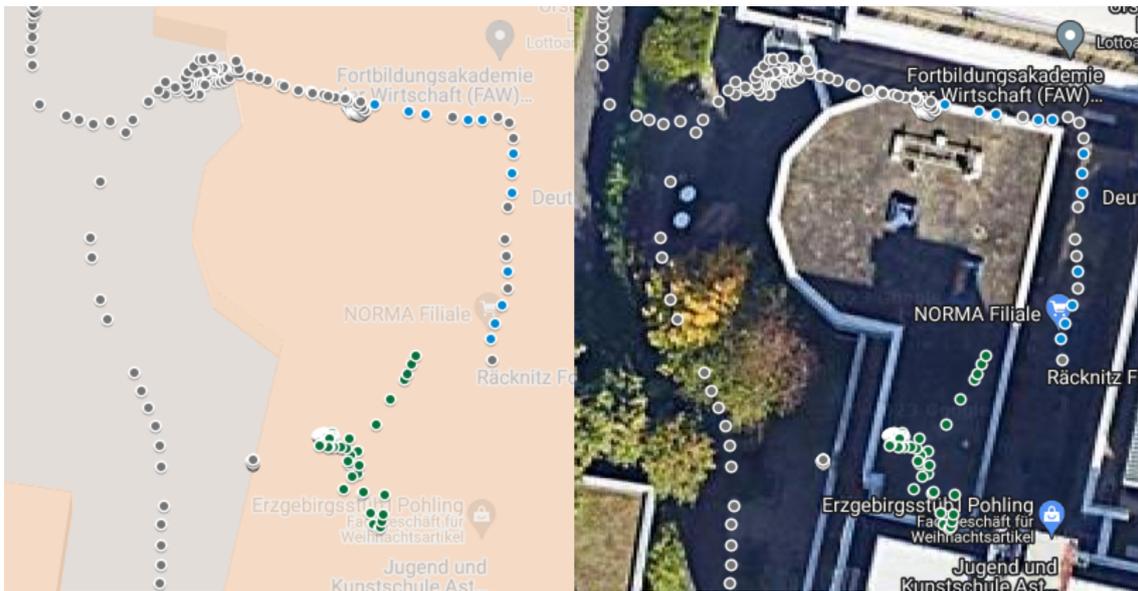


Abbildung 5.2.1: Visualisierung der Positionen in Google Maps mit Fokus auf den Übergang der Lokalisierungszonen, links auf „Wildwasser“-Hintergrund, rechts auf Satellitenhintergrund

tung der abgesonderte Häufung von GNSS-Positionen am westlichen Gebäuderand fällt auf, dass diese ermittelten Positionen nicht die reale Position widerspiegeln. Die Satellitenaufnahme verdeutlicht, dass es sich um eine nicht betretbaren Gebäudebereich handelt. Wenn die Satellitenaufnahmen positionskorrekt und perspektivenlos sind, dann sind die gemessenen Positionen verzerrt. Innerhalb des Gebäudes sind die Positionen des Hybrid-Trackables regelmäßig und weisen keine irregulären Häufungen auf, was durch die Daten aus dem UWB-LP zu erwarten war.

Auf der östlichen Terrasse zwischen zwei Gebäudeabschnitten werden die Positionen aus den Daten des GNSS-LP erzeugt. Obwohl zu allen vier Seiten Gebäude die Sicht auf den Himmel einschränken und damit nur ein schmaler Sichtkegel direkt nach oben verbleibt, weisen die GNSS-Daten durch eine maximale Abweichung von zwei Metern zum tatsächlichen Laufweg auf. Das vermehrte Auftreten von exklusiven GNSS-Positionen soll genauer untersucht werden.

Abbildung 5.2.2 zeigt den Positionsverlauf an der zu betrachtenden Zonengrenze. Links sind die ersten 1000 Positionen der Messreihe dargestellt, rechts die letzten 200. Die rote Markierung zeigt den Positionsverlauf des GNSS-Trackables, grüne Markierungen zeigen Positionen des Hybrid-Trackables. Die Darstellung ermöglicht somit ein differenziertes Betrachten vom Start und dem Ende des Systemtests. Der linke Positionsverlauf zeigt, dass zum Start der Messreihe keine GNSS Positionen auf der Terrasse zwischen den Häusern ermittelt wurden und mit größerem Sichtkegel eine abrupte Positionsänderung stattfindet. Dies bedeutet, dass der GNSS-LP seine zuletzt berechnete Position weiter übermittelt, bis eine neu berechnete Position vorliegt. Die Position des

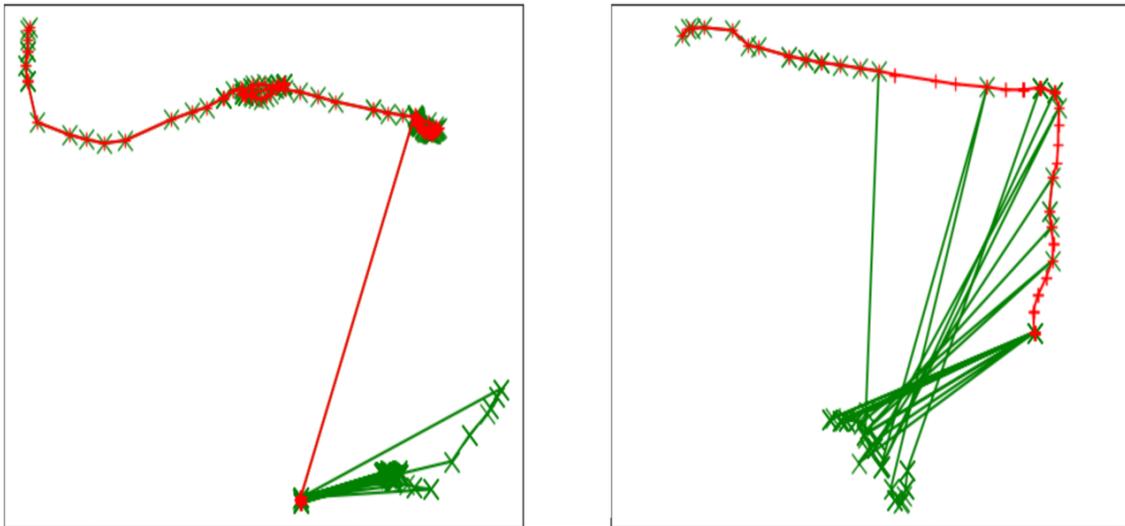


Abbildung 5.2.2: Visualisierung des Positionsverlaufs zu Beginn und am Ende der Messung

Hybrid-Trackables wird sprunghaft geändert und nimmt am Startpunkt der Testmessung wiederholt die Position des GNSS-Trackables an. Die Bewegung durch das Firmengebäude zur Terrasse ist trotz der wiederholten Positionssprünge erkennbar.

In der rechten Abbildung wird gezeigt, dass die GNSS Positionen bis zum Wiedereintritt in das Firmengebäude ermittelt werden. Die exklusiven Positionen des GNSS-Trackables sind durch sprunghaftes Wechseln des Hybrid-Trackables zu neuen UWB Positionen erklärbar. Mit Wiedereintritt in die UWB-Zone werden vom RTLS trotz mangelnder Verbindung zu ausreichend omlox Satelliten vereinzelt falsche Positionsdaten erzeugt, welche entsprechend der Regelsetzung des Hybrid-Trackables priorisiert werden. Dieses Verhalten des RTLS ist bekannt und wird in zukünftigen Versionen des RTLS korrigiert.

Bei weiterer Betrachtung des gesamten Laufweges (Abbildung 5.1.2) fällt ein kleiner Abschnitt im Nordosten des Testweges auf. In diesem Abschnitt gab es vereinzelt Positionen, die im Vergleich mit dem restlichen Positionsverlauf weit auseinander liegen. Bei einem Abgleich der dokumentierten Zeitstempel der gesammelten Lokationsdaten ist hier eine Differenz von jeweils einer Sekunde zu verzeichnen. Es handelt sich damit also nicht um ein Übertragungsproblem, sondern um ein Positionsermittlungsproblem seitens des GNSS-Empfängers. An dieser Passage stehen die Bäume besonders dicht am Straßenrand. Zusätzlich grenzt die Straße an ein Kaufhaus und ein Haus Neubauwohnungen, wodurch eine Messbeeinträchtigung plausibel ist. Die Genauigkeit kann durch eine Betrachtung der zuverlässigen Positionen abgeschätzt werden, wie in Abbildung 5.2.3 dargestellt.

Vergleicht man die markierten Positionen, stellt man einen Abstand von circa 41 Meter bei einer



Abbildung 5.2.3: Ausschnitt des Positionsverlauf mit Markierung der Referenzpositionen

Zeitdifferenz von fünfzehn Sekunden fest. Zum Vergleich werden Positionen approximiert, die bei konstanter Geschwindigkeit und einer linearen Bewegung zwischen diesen markierten Punkten ermittelt werden. Abbildung 5.2.4 zeigt diese Gegenüberstellung.

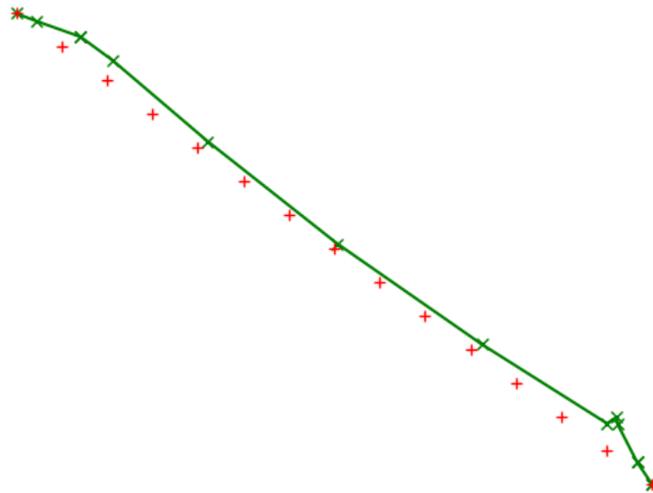


Abbildung 5.2.4: Gegenüberstellung erfasster (grün) und approximierter (rot) Positionsverlauf

Position eins (Startpunkt der Messwerte, in der Abbildung rechts unten dargestellt) ist per Vergleichsdefinition identisch mit dem Startpunkt der approximierten Positionen, gleiches gilt für Position fünfzehn (Endpunkt der Messerte, links oben). Die auf Position eins folgende Messposition wird mit der nächsten angenäherten Position verglichen und der Abstand zueinander berechnet. Dies wird bis einschließlich der fünfzehnten Position wiederholt, wodurch sich Tabelle 5.2.1 ergibt.

Positionpaar	1	2	3	4	5
Abstand	0,0m	2,9m	4,8m	7,8m	9,4m
Positionpaar	6	7	8	9	10
Abstand	12,2m	14,6m	9,5m	3,1m	2,4m
Positionpaar	11	12	13	14	15
Abstand	5,6m	4,8m	1,9m	1,7m	0,0m

Tabelle 5.2.1: Tabellarische Darstellung der Abweichung zwischen Mess- und approximierten Positionen

5.3 Abgleich mit aufgestellten Anforderungen

Für die abschließende Beurteilung des Konzepts sollen die aufgestellten Anforderungen aus Abschnitt 3.2 als Maßstab aufgegriffen werden. Basierend auf einer Skala von null bis fünf, wobei null ein absolutes Nicht-Erfüllen der Anforderung und fünf ein vollständiges Erfüllen der Anforderung darstellt, werden die Kriterien unabhängig voneinander bewertet.

Durch die Ergebnisse des Testlauf lässt sich festhalten, dass abseits des Zonenübergangs zuverlässige Positionsdaten erhoben werden können. Wie die Positionskarte zeigt, sind selbst auf Strecken mit eingeschränktem Blickfeld zum Himmel Positionen ermittelbar. Diese Anforderung wurde vollstens erfüllt. Die minimale Genauigkeit von zehn Metern wird im Testlauf außerhalb der UWB Zone in der Regel erreicht, eine Ausnahme bildet der vorangegangene analysierte Streckenabschnitt. Dieser weist einen Abstand von 12,2m beziehungsweise 14,6m auf. Dies lässt sich auf die stark ausgeprägte Bepflanzung zurückführen, wie sie bei einer Umsetzung am Industriebeispiel nicht zu erwarten ist. Zudem sind die Vergleichspositionen linear angenähert und somit keine fehlerlose Abbildung der Realität. Im Vergleich mit dem Positionsverlauf der verbleibenden GNSS Positionen zeigt die dargelegte Abweichung, dass in seltenen Fällen, wie dem betrachteten Wegabschnitt, eine Genauigkeit von zehn Metern temporär nicht erreicht werden könnte.

Der Test wurde mit einer Positionserfassungsrate von einem Hertz durchgeführt. Eine Analyse der Zeitstempel am DeepHub ergibt, dass der größte Zeitabstand zwischen zwei Positionen beim Hybrid-Trackable 1,025s Differenz beträgt. Verglichen wurde der dokumentierte Zeitstempel, der das Eintreffen der Position im Hub markiert. Diese Differenz kann durch den Übertragungsweg (WLAN, VPN) und der Datenverarbeitung durch die MW begründet werden. Um die Anforderung von mindestens 0,2Hz zu erfüllen, muss die beobachtete Verzögerung beachtet werden, weshalb die Erfassungsfrequenz des Sensors nicht ebenfalls 0,2Hz betragen darf. Allerdings ist die Zeitdifferenz so gering, dass schon bei einer Erfassungsfrequenz von 0,25Hz, also einer ermittelten Position alle vier Sekunden, die minimale Positionsfrequenz von 0,2Hz am Hub als gesichert betrachtet werden kann.

Eine Priorisierung von UWB-Positionen ist bei dem Hybrid-Trackable gegeben. Im Test wurde eine prototypische Regel zur Differenzierung genutzt, der DeepHub ermöglicht aber auch komplexere Regeldefinitionen. Das Kriterium ist somit grundsätzlich erfüllt, im Produktivsystem ist eine genauere allgemeine Regeldefinition zur Laufzeit möglich.

Das Konzept verwendet viele Aspekte der bereits vorhandenen Architektur. Der bereits existierende MQTT-Broker wird als Datenvermittler verwendet, zur Datenübertragung wird das etablierte WLAN genutzt. Der DeepHub ist ebenfalls bereits installiert, womit nur die Middleware und die Sensoren neu in die Infrastruktur des Netzwerks integriert werden müssen. Zur Übersicht wird abschließend die diskutierte Beurteilung in Form der Tabelle 5.3.1 festgehalten.

Nr.	Kriterium	Erfüllungsgrad
1	Positionsdaten müssen vorhanden sein	5/5
2	Die Genauigkeit beträgt zehn Meter oder weniger	4/5
3	Die Frequenz soll 0,2Hz nicht unterschreiten	4/5
4	UWB-Positionen sollen präferenziert werden	4/5
5	Vorhandene Infra-/Netzwerkstruktur soll genutzt werden	5/5

Tabelle 5.3.1: Tabellarische Darstellung aufgestellter Anforderungen mit Erfüllungsgrad

Kapitel 6

Ergebnisse

In der Einleitung wird die Frage gestellt, wie die omlox Hub Spezifikation die Einbindung von zusätzlichen Technologien neben der Installation einer auf UWB basierenden Core Zone unterstützt. Diese Arbeit zeigt im Grundlagenkapitel und durch die Präsentation von drei unterschiedlichen Konzepten, dass die Technologie zur Positionsbestimmung innerhalb einer Complementary Zone frei wählbar ist. Das Übergabeformat der Positionsinformationen an den Hub muss eingehalten werden, aber die Wahl des Verfahrens zur Ermittlung einer Position sowie die genutzten Technologien obliegen dem Systementwickler.

Dominante Probleme bei der Umsetzung des GNSS Konzeptes sind hohen Anforderungen an den zu nutzenden Sensor sowie die Übertragung und Aufbereitung der gewonnenen Rohdaten. Gelöst wurden diese Herausforderungen jeweils durch eine konfigurierbare Sensorlösung von u-blox, eine sensorseitige Selektion der erzeugten NMEA Nachrichten und eine Datenumformung durch eine Python3-MW zur Übertragung an die omlox Hub Schnittstelle.

Es wurde festgelegt, dass die in dieser Arbeit entwickelte Lokalisierungslösung den Kriterien des industriellen Anwendungsfalls entsprechen und den aktuellen omlox v2 Standards folgen soll. Die Ergebnisse werden diesen Forderungen gerecht. Das umgesetzte GNSS-Konzept ist mit allen Spezifikationen des omlox v2 Standards, insbesondere der Hub Spezifikation, konform und erfüllt beim Systemtest alle in Abschnitt 3.2 aufgestellten Anforderungen. Das entwickelte System wurde modular gestaltet, so dass eine Vielzahl von Erweiterungen möglich sind.

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

In dieser Arbeit wurde in den Grundlagen der omlox v2 Standard sowie die Lokalisierungstechnologien UWB, GNSS, WiPS und Mobilfunk und relevante Lokalisierungsmethoden erläutert, gefolgt von einer Erklärung und Analyse des beispielhaften Industriebeispiels. In der Analyse wurden Kriterien ausgearbeitet, bei denen der Frequenz der Positionsermittlung und der Genauigkeit hohe Bedeutung zugewiesen wurden. Zudem wurde gefordert, dass aktuelle UWB-Positionen gegenüber Positionen aus dem ergänzenden Positionssystem zu priorisieren sind. Basierend auf den Grundlagen und den Anforderungen wurden drei verschiedene Konzepte vorgestellt, deren Vor- und Nachteile bezüglich der Anforderungen und der Umsetzbarkeit abschließend diskutiert wurden. Das GNSS-Konzept, bestehend aus dem namensgebenden GNSS-Sensor, einem WLAN-Modul und einer Middleware, war für dieses Industriebeispiel am geeignetsten und wurde genauer spezifiziert. Es wurde eine Anpassung auf die existierende Systemarchitektur vorgeschlagen und ein bereits installierter MQTT-Broker als Vermittler zwischen Sensorboard und Middleware genutzt. Die omlox Core Zone war bereits in Form eines RTLS der ZIGPOS GmbH installiert und stellte den Broker zur Verfügung. Als omlox Hub wurde der DeepHub der Flowcate GmbH installiert. Für die Umsetzung des Konzepts wurde ein u-blox C099-F9P Application Board als GNSS-Sensor mit WLAN-Modul gewählt. Die Konfiguration erfolgte nach dem Flashen der u-connectXpress-Firmware durch die Nutzung des AT-Befehlssatz. Das ZED-F9P-Modul ermittelte die derzeitige Position durch die Nutzung der GPS- und Galileo-Satellitensysteme. Das ODIN-W2-Modul steuerte die Verbindung zum existierenden WLAN und übermittelte die gewonnenen Positionsdaten im NMEA-Standard an den MQTT-Broker. Die Middleware wurde in Python3 geschrieben. Sie verband sich mit dem MQTT-Broker und brachte die Positionsdaten in das vom omlox Hub Standard spezifizierte Format. Ein Systemtest des umgesetzten Konzepts zeigte, dass alle Anforderungen erfüllt wurden.

7.2 Ausblick

Die Möglichkeiten zum Ausbau dieser Arbeit sind vielzählig. Vor allem in den Bereichen Automatisierung, Ausfallsicherheit, Skalierung und Präzision lassen sich Verbesserungen vornehmen und die vorliegende Ausarbeitung erweitern.

7.2.1 Automatisierung

Die Konfiguration des C099-F9P Application Board wird in dieser Arbeit manuell ausgeführt. Bei der Arbeit mit einem fabrikneuen Board müssen Daten, beispielsweise das zu nutzende WLAN und die Adresse des MQTT-Brokers, durch einzelne Befehle manuell eingegeben werden. Das Board bietet allerdings die Möglichkeit, Settings-Dateien zu exportieren und auch zu importieren. Möglich wäre also ein System, welches unter Beachtung von boardspezifischen Daten, zum Beispiel der Mac-Adressen, eine Konfigurationsdatei erstellt und diese an das Board überträgt. Ein solches Werkzeug würde die notwendige Zeit für eine Inbetriebnahme eines C099-F9P maßgeblich reduzieren.

Bei längerem Netzwerkverlust entfernt der ODIN-W2 den MQTT-Eintrag aus seiner Verbindungsliste. Wenn der ODIN sich wieder im Netzwerk befindet, ist ein Senden der Positionsdaten nicht möglich. Der ODIN bietet allerdings eine TCP-Schnittstelle an. Wenn diese das Ausführen von Befehlen und das Senden der Stopsequenz des Data-Modus ermöglicht, kann man bei Anmeldung eines ODINs im Netzwerk automatisch in den AT-Modus wechseln, die Adressdaten des Brokers übermitteln und erneut den Data-Modus aktivieren.

7.2.2 Ausfallsicherheit und Skalierung

Im Kontext dieser Arbeit wurde das C099-F9P Application Board beim Systemtest durch eine Powerbank mit Energie versorgt. Um eine durchgängige Energieversorgung zu gewährleisten, sollte untersucht werden, ob parallel zum Betrieb der Powerbank eine Ladung dieser möglich ist. Ist dies der Fall, so kann man die Powerbank an die Fahrzeugelektronik anschließen und so diese laden. Bei Nicht-Betrieb des Fahrzeugs stellt die Powerbank die notwendige Energie zum Betrieb des Application Boards zur Verfügung. Insofern eine regelmäßige Nutzung des Lastkraftwagens vorliegt, kann der Sensor über einen unbegrenzten Zeitraum betrieben werden. Zusätzlich wäre eine Reduzierung der Positionsermittlungsrate des ZED möglich, gesteuert durch Zonen-Events des DeepHub und der in Abschnitt 7.2.1 angesprochenen TCP-Schnittstelle. So kann der Energieverbrauch des Application Boards stark reduziert werden.

Um den in Abschnitt 7.2.1 angesprochenen Abriss der Netzwerkverbindung zu verhindern, sollte eine umfangreiche Signalanalyse vor Ort durchgeführt werden. Eine solche Analyse bezieht lokale Faktoren ein, welche nicht durch eine Simulation abbildbar sind. Auch die Erweiterung des Netzwerks um zusätzliche AP für eine bessere Abdeckung sollte in Betracht gezogen werden.

Bei steigender Sensoranzahl wächst auch die Anforderung an die MW. Sollte diese zu groß, und somit der Informationsfluss und die minimale Frequenz von 0,2 Positionen pro Sekunde gefährdet werden, sollte eine Aufspaltung der MW in mehrere Instanzen in Betracht gezogen werden. Durch die Nutzung von Virtualisierungslösungen wie Docker können die Instanzen parallel zueinander existieren und müssen nur auf unterschiedliche MQTT-Topics angemeldet werden. So kann die Last auf die einzelnen Instanzen aufgeteilt werden. Um den Aufwand der Datenübertragung zum DeepHub zu reduzieren, kann von der Nutzung des REST-Interfaces abgesehen werden und statt dessen eine Verbindung durch einen Websocket eingerichtet werden. Eine entsprechende API wird vom DeepHub angeboten, die Verbindung erfordert bei hochfrequenter Kommunikation weniger Overhead als eine vergleichbare HTTP-Kommunikation.

7.2.3 Präzisionsverbesserung

Die Positionsergebnisse des GNSS-LP erfüllen das aufgestellte Kriterium vollständig. Sollte eine höhere Genauigkeit angestrebt werden, so ergeben sich drei Ansätze.

Der erste Ansatz bezieht sich auf eine Genauigkeitsoptimierung am GNSS-Sensor. Durch die Installation eines RTK-Moduls können dem ZED netzwerkweit Korrekturdaten zur Verfügung gestellt werden. Durch den Wechsel auf DGNSS ist bei den GNSS-Daten eine deutlich höhere Genauigkeit zu erwarten.

Der zweite Ansatz bezieht sich auf den Übergangszone zwischen UWB- und GPS-Lokalisierung. Er sieht die Nutzung der GNGGA-Nachricht des NMEA-Standards vor, da dieser neben den Positionsdaten auch Informationen zur Genauigkeit enthält. Dokumenten der Technischen Universität München ist aber zu entnehmen, dass diese Informationen nur bei Nutzung von GPS-Satelliten verfügbar sind¹⁸. Durch eine Genauigkeitsabschätzung der Positionsdaten sind komplexe Regeln denkbar, welche das „accuracy“-Attribut des DeepHub nutzen. Eine generelle Priorisierung von UWB-Positionen am Rand des Systems kann dann zugunsten eines fließenden Übergangs bei ausreichender Präzision ersetzt werden.

Der dritte Ansatz bezieht sich auf die Datenverarbeitung in der MW. Da sich die Lastkraftwagen durch die Abhängigkeit zu asphaltierten Bereichen nicht frei im Gelände bewegen können, kann eine Aufbereitung der Positionsdaten vor der Übermittlung an den DeepHub vorgenommen werden. Dies wird als Map-Matching bezeichnet. Zum Beispiel wäre eine Korrektur der Position auf die Straße möglich, sollte der GNSS eine Position knapp neben der Straße ermitteln. Weiterführend kann die Position basierend auf der Bewegungsrichtung auf die rechte Fahrspur korrigiert werden. Ähnliche Korrekturen können auf dem Parkplatz oder den Ladezonen vorgenommen werden, da dort der Standbereich klar definiert ist. Um eine solche Korrektur umsetzen, ist eine Datenaggregation, beispielsweise in einer Datenbank, sowie eine Vielzahl an Korrekturregeln notwendig. Der Entwicklungsaufwand ist hoch und die Leistungsanforderung an die Komponente

¹⁸ Siehe <https://campar.in.tum.de/twiki/pub/Chair/NaviGpsDemon/nmea.html>, zuletzt aufgerufen am 19.7.2023

wächst mit steigender Sensoranzahl. Wenn diese Systemkomponente allgemein anwendbar und somit unabhängig von den Randbedingungen des Industriebeispiels entwickelt wird, kann man dieses System nutzen, weiterführend Anwendungen abzuleiten. So wäre es möglich, ein neuronales Netz auf den jeweiligen Anwendungsfall zu trainieren. Diese hoch spezialisierten Systeme benötigen zur Laufzeit oft nur einen Bruchteil der Leistung und Systemkapazität eines traditionellen Systems bei ähnlichen oder sogar besseren Ergebnissen, ein noch unausgeschöpftes Potential im Bereich der Lokalisierungstechnologien.

Quellenverzeichnis

- [ASA01] ASTLEY, Mark ; STURMAN, Daniel C. ; AGHA, Gul A.: Customizable Middleware for Modular Distributed Software - Simplifying the development and maintenance of complex distributed software. In: *Communications of the ACM* 44 (2001), Mai, No. 5, 9. http://osl.cs.illinois.edu/media/papers/astley-2001-cacm-customizable_middleware_for_modular_distributed_software.pdf
(zitiert auf Seiten 3 und 4)
- [Aug23] AUGUST, Nathaniel J.: *Appendix 10.C. UWB Regulations | An Introduction to Ultra Wideband Communication Systems*. Version: Mai 2023. <https://flylib.com/books/en/4.390.1.81/1/>
(zitiert auf Seite 16)
- [Bak03] BAKKEN, David E.: *Middleware - What it is, and How it Enables Adaptivity and Dependability*. Version: Januar 2003. <https://webhost.laas.fr/TSF/IFIPWG/Workshops&Meetings/43/01-Bakken.pdf>
(zitiert auf Seiten 3 und 4)
- [CMJS13] CHAPRE, Yogita ; MOHAPATRA, Prasant ; JHA, Sanjay ; SENEVIRATNE, Aruna: Received signal strength indicator and its analysis in a typical WLAN system (short paper). In: *38th Annual IEEE Conference on Local Computer Networks*. Sydney, NSW : IEEE, Oktober 2013. – ISBN 978-1-4799-0537-9 978-1-4799-0536-2, 304-307
(zitiert auf Seite 13)
- [CTM⁺04] CURRY, Edward ; TAI, Stefan ; MIKALSEN, Thomas ; ROUVELLOU, Isabelle ; JUNGINGER, Markus O. ; LASZEWSKI, Gregor von ; AMIN, Kaizar ; WANG, Nanbor ; GILL, Christopher D. ; SCHMIDT, Douglas C. ; GOKHALE, Aniruddha ; NATARAJAN, Balachandra ; LOYALL, Joseph P. ; SCHANTZ, Richard E. ; RODRIGUES, Craig ; GRAY, Jeff ; MAJUMDAR, Shikharesh ; DEMURJIAN, Steven ; BESSETTE, Keith ; DOAN, Thuong ; PHILLIPS, Charles ; CHRYSANTHIS, Panos K. ; LIBERATORE, Vincenzo ; PRUHS, Kirk ; MASCOLO, Cecilia ; CAPRA, Licia ; EMMERICH, Wolfgang ; WANG, Guijun ; CHEN, Alice ; SRIPADA, Surya ; WANG, Changzhou ; LANGENDÖRFER, Peter ; MAYE, Oliver ; DYKA, Zoya ; SORGE, Roland ; WINKLER, Rita ; KRAEMER, Rolp ; LEUNG, Karl R. P. H. ; NG, Joseph Kee-Yin ; HUI, Calvin Kin-Cheung ; VOGT, Harald ; ROHS, Michael ; KILIAN-KEHR, Roger ;

- MARTÍNEZ, Jesús ; LÓPEZ, Luis R. ; MERINO, Pedro ; KRISHNA, Arvind S. ; KLEFSTAD, Raymond ; CARSARO, Angelo ; SZENTIVÁNYI, Diana ; NADJM-TEHRANI, Simin ; MAHMOUD, Qusay H. (Hrsg.): *Middleware for Communications*. hichester, West Sussex PO19 8SQ, England : John Wiley & Sons, Ltd, 2004. – ISBN 0–470–86206–8
(zitiert auf Seite 4)
- [DC19] DINCULEANĂ, Dan ; CHENG, Xiaochun: Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. In: *Applied Sciences* 9 (2019), Februar, No. 5, 848. <http://dx.doi.org/10.3390/app9050848>. – DOI 10.3390/app9050848. – ISSN 2076–3417
(zitiert auf Seiten 6 und 7)
- [Dig22] DIGGELEN, Frank van: *Indoor GPS theory & implementation*. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=998914>.
Version: 2022
(zitiert auf Seite 18)
- [FCC20] FCC: Part15 - Radio Frequency Devices / Federal Communications Commission. Version: 2020. <https://www.govinfo.gov/content/pkg/CFR-2020-title47-vol1/pdf/CFR-2020-title47-vol1-part15.pdf>. USA, 2020. – Forschungsbericht. – 125 S.
(zitiert auf Seiten 15 und 16)
- [Fie00] FIELDING, Roy T.: *Architectural Styles and the Design of Network-based Software Architectures*. (2000)
(zitiert auf Seiten 5 und 6)
- [FS96] FERSTL, Ernst ; SABARY, Ferdinand: *Unter der Oberfläche: Gedanken mit Tiefgang*. Ed. Va Bene, 1996
(zitiert auf Seite I)
- [FSF09] FENG, Xinyang ; SHEN, Jianjing ; FAN, Ying: *REST : An Alternative to RPC for Web Services Architecture*. (2009)
(zitiert auf Seiten 5 und 6)
- [ISL⁺] IBACH, Peter ; STANTCHEV, Vladimir ; LEDERER, Florian ; WEISS, Andreas ; HERBST, Thomas ; KUNZE, Torsten: *WLAN-BASED ASSET TRACKING FOR WAREHOUSE MANAGEMENT*.
(zitiert auf Seite 17)
- [KBL22] KLYMASH, Mikhailo (Hrsg.) ; BESHLEY, Mykola (Hrsg.) ; LUNTOVSKYY, Andriy (Hrsg.): *Lecture Notes in Electrical Engineering*. Bd. 831: *Future Intent-Based Networking: On the QoS Robust and Energy Efficient Heterogeneous Software Defined Networks*. Cham : Springer International Publishing, 2022. <http://dx.doi.org/10.1007/>

- 978-3-030-92435-5. <http://dx.doi.org/10.1007/978-3-030-92435-5>. – ISBN 978-3-030-92433-1 978-3-030-92435-5
(zitiert auf Seite 3)
- [KHNf05] KITASUKA, T. ; HISAZUMI, K. ; NAKANISHI, T. ; FUKUDA, A.: WiPS: Location and Motion Sensing Technique of IEEE 802.11 Devices. In: *Third International Conference on Information Technology and Applications (ICITA'05)* Bd. 2. Sydney, Australia : IEEE, 2005. – ISBN 978-0-7695-2316-3, 346–349
(zitiert auf Seite 17)
- [KK04] KAEMARUNGS, K. ; KRISHNAMURTHY, P.: Properties of indoor received signal strength for WLAN location fingerprinting. In: *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004*. Boston, MA, USA : IEEE, 2004. – ISBN 978-0-7695-2208-1, 14–23
(zitiert auf Seite 13)
- [Koc05] KOCH, Stefan (Hrsg.): *Free/open source software development*. Hershey, Pa : Idea Group Pub, 2005. – ISBN 978-1-59140-369-2 978-1-59140-370-8 978-1-59140-371-5
(zitiert auf Seite 3)
- [KR03] KULKARNI, V. ; REDDY, S.: Separation of concerns in model-driven development. In: *IEEE Software* 20 (2003), September, No. 5, 64–69. <http://dx.doi.org/10.1109/MS.2003.1231154>. – DOI 10.1109/MS.2003.1231154. – ISSN 0740-7459
(zitiert auf Seite 5)
- [LG20] LUNTOVSKYY, Andriy ; GÜTTER, Dietbert: *Moderne Rechnernetze: Protokolle, Standards und Apps in kombinierten drahtgebundenen, mobilen und drahtlosen Netzwerken*. Wiesbaden : Springer Fachmedien Wiesbaden, 2020. <http://dx.doi.org/10.1007/978-3-658-25617-3>. <http://dx.doi.org/10.1007/978-3-658-25617-3>. – ISBN 978-3-658-25616-6 978-3-658-25617-3
(zitiert auf Seiten 18, 19 und 20)
- [LG22] LUNTOVSKYY, Andriy ; GÜTTER, Dietbert: *Highly-Distributed Systems: IoT, Robotics, Mobile Apps, Energy Efficiency, Security*. Cham : Springer International Publishing, 2022. <http://dx.doi.org/10.1007/978-3-030-92829-2>. <http://dx.doi.org/10.1007/978-3-030-92829-2>. – ISBN 978-3-030-92828-5 978-3-030-92829-2
(zitiert auf Seiten 3 und 5)
- [NMM⁺99] NIELSEN, Henrik ; MOGUL, Jeffrey ; MASINTER, Larry M. ; FIELDING, Roy T. ; GETTYS, Jim ; LEACH, Paul J. ; BERNERS-LEE, Tim: Hypertext Transfer Protocol – HTTP/1.1 / Internet Engineering Task Force. Version: Juni 1999. <http://dx.doi.org/10.1007/978-3-030-92829-2>

- 17487/RFC2616. 1999 (RFC 2616). – Request for Comments. – Num Pages: 176
(zitiert auf Seiten 5 und 6)
- [noa16] *ISO/IEC-20922-2016*. Juni 2016
(zitiert auf Seiten 6 und 7)
- [noa20a] *APR001 APPLICATION NOTE*. www.qorvo.com/products/d/da008433.
Version: 2020
(zitiert auf Seite 15)
- [noa20b] *IEEE Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*. <http://dx.doi.org/10.1109/IEEESTD.2020.9179124>. Version: Juni 2020. – ISBN: 9781504467988
(zitiert auf Seiten 13, 14, 15 und 16)
- [noa20c] *IEEE Std 802.15.4-2020, IEEE Standard for Low-Rate Wireless Networks*. Mai 2020
(zitiert auf Seite 15)
- [noa22] *IEEE Standard for Information Technology - Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks - Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 2022
(zitiert auf Seite 17)
- [noa23] *Ultra Wide Band*. <https://www.etsi.org/technologies/ultra-wide-band>.
Version: 2023
(zitiert auf Seite 15)
- [OHI04] OPPERMANN, Ian ; HÄMÄLÄINEN, Matti ; IINATTI, Jari: *UWB Theory and Applications*. West Sussex PO 19 8SQ : John Wiley & Sons Ltd, 2004. – ISBN 0–470–86917–8
(zitiert auf Seite 15)
- [SA11] SI, Hnin ; AUNG, Zaw M.: Position Data Acquisition from NMEA Protocol of Global Positioning System. In: *International Journal of Computer and Electrical Engineering* (2011), 353–357. <http://dx.doi.org/10.7763/IJCEE.2011.V3.340>. – DOI 10.7763/IJCEE.2011.V3.340. – ISSN 17938163
(zitiert auf Seite 18)
- [SF15] SIVERS, Mstislav ; FOKIN, Grigoriy: LTE Positioning Accuracy Performance Evaluation. Version: 2015. https://link.springer.com/10.1007/978-3-319-23126-6_35. In: BALANDIN, Sergey (Hrsg.) ; ANDREEV, Sergey (Hrsg.) ; KOUCHERYAVY, Yevgeni (Hrsg.): *Internet of Things, Smart Spaces, and Next Generation Networks and Systems* Bd. 9247. Cham : Springer International Publishing, 2015. – ISBN

978-3-319-23125-9 978-3-319-23126-6, 393-406. – Series Title: Lecture Notes in Computer Science
(zitiert auf Seite 20)

[TM17] TEUNISSEN, Peter J. (Hrsg.) ; MONTENBRUCK, Oliver (Hrsg.): *Springer Handbook of Global Navigation Satellite Systems*. Cham : Springer International Publishing, 2017. <http://dx.doi.org/10.1007/978-3-319-42928-1>. <http://dx.doi.org/10.1007/978-3-319-42928-1>. – ISBN 978-3-319-42926-7 978-3-319-42928-1
(zitiert auf Seite 18)

[Tre96] TRESCH, Markus: Middleware: Schlüsseltechnologie zur Entwicklung verteilter Informationssysteme. In: *Informatik-Spektrum* 19 (1996), Oktober, No. 5, 249-256. <http://dx.doi.org/10.1007/s002870050035>. – DOI 10.1007/s002870050035. – ISSN 0170-6012, 1432-122X
(zitiert auf Seiten 3 und 4)

[VK] VELAYOS, Hector ; KARLSSON, Gunnar: Limitations of range estimation in wireless LAN / KTH, Royal Institute of Technology,. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=de18e1dc40a6ce08ba2897b6f0d395b6e516e66e>. Stockholm, . – Forschungsbericht. – 6 S.
(zitiert auf Seite 17)

[YBMD14] YOZEVITCH, Roi ; BEN-MOSHE, Boaz ; DVIR, Amit: GNSS Accuracy Improvement Using Rapid Shadow Transitions. In: *IEEE Transactions on Intelligent Transportation Systems* 15 (2014), Juni, No. 3, 1113-1122. <http://dx.doi.org/10.1109/TITS.2013.2294537>. – DOI 10.1109/TITS.2013.2294537. – ISSN 1524-9050, 1558-0016
(zitiert auf Seite 18)

Hilfsmittel

Zur Erstellung der vorliegenden Abschlussarbeit wurden die folgenden Hilfsmittel verwendet:

- Dokumentenbetrachter (Evince, Version 42.3),
- draw.io (Version 20.6.2),
- Firefox Browser (Version 112.0.2, 64-Bit),
- Google-Maps
- Google-Scholar,
- Google-Suchmaschine,
- Kile (Version 2.9.93),
- Korrekturlesende (namentlich: Danny Schumann, Marcus Schütze, Steffi Preßler).
- MQTT-Explorer (Version 0.3.5)
- Okular (Version 21.12.3),
- Overleaf Desktop (Version 3.0.0, Snap Store)
- PyCharm 2022.3 (Professional Edition, Build #PY-223.7571.203)
- s-center (Version 7.0.1)
- Springer Link,
- Systemtestende (namentlich: Jan Sulimma),
- u-center (Version 22.07), und
- ZIGPOS Planning Tool,
- Zotero (Version 6.0.23)

*Der Unterschied zwischen Theorie und Praxis
ist in der Praxis größer als in der Theorie.*

— Ernst Ferstl [FS96]

Danksagung

Mit dieser Seite möchte ich mich bei all denjenigen bedanken, die mich während der Erstellung dieser Bachelorarbeit unterstützt haben.

Allem voran danke ich Danny Schumann für die Betreuung meiner Arbeit. Die zahlreichen hilfreichen Anmerkungen, das wiederholte Korrekturlesen und das unermüdliche Feedback zu meiner Ausarbeitung waren von unschätzbarem Wert.

Auch Prof. Dr habil. Andriy Luntovskyy möchte ich für sein Engagement bei der Betreuung dieser Arbeit danken. Ihre Gedanken und Impulse haben meine Arbeit in die richtige Richtung gebracht.

Ein weiterer Dank gilt der ZIGPOS GmbH, insbesondere meinen Kollegen. Danke für die geballte Expertise, die Geräte und die Unterlagen, die ihr mir zur Verfügung gestellt habt.

Bei meinem Korrekturlesern und Systemtestern Marcus Schütze, Steffi Preßler und Jan Sulimma möchte ich mich ebenfalls bedanken. Eure Hilfe hat mir Vieles erleichtert.

Zum Schluss gebührt meinen Freunden und meiner Familie Dank. Danke, dass ihr in den vergangenen Wochen so viel Geduld und immer ein offenes Ohr für mich hattet.

Erklärung an Eidesstatt

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen sowie die explizit aufgeführten Hilfsmittel verwendet habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden, ansonsten sind sie mit einem entsprechenden Quellennachweis versehen.

Diese Ausarbeitung ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Dresden, 2. August 2023

Nicolas Kohnert