



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Bachelor Thesis

Smartwatches in siot.net

Bachelor Thesis im Studiengang Informatik an der Berner Fachhochschule

Autor

Sathesh Paramasamy

Betreuer Dozent

Dr. Andreas Danuser

Version X1.0

12. Januar 2016



Kontaktpersonen

Autor

Sathesh Paramasamy
Kirchweg 54
3324 Hindelbank

E-Mail: sathesh.paramasamy@students.bfh.ch

Betreuerender Dozent

Dr. Andreas Danuser
Berner Fachhochschule
Technik und Informatik
Höheweg 80
2502 Biel

E-Mail: andreas.danuser@bfh.ch

Beurteilender Experte

Armin Blum
Burgunderweg 58
2502 Biel

E-Mail: armin.blum@bluewin.ch

Management Summary

An der Berner Fachhochschule konzipiert und entwickelt die Fachgruppe SIOT des Institutes RISIS (Research Institute for Security in the Information Society) mit Industriepartnern (AppModule und NetModule) die siot.net Plattform, welche Sensoren und Aktoren mit Internet-of-Things-Anwendungen verbindet.

Smartwatches und Smartphones, die grosse Anzahl Sensoren und Aktoren integrieren, sollen effektiv in der siot.net Plattform genutzt werden. Es gibt eine Vielzahl von Smartwatches auf dem Markt mit verschiedenster Betriebssystemen und Funktionalitäten. Um eine Auswahl zu treffen, wurden mehrere Geräte betrachtet und bewertet.

Inhaltsverzeichnis

1 Einleitung	1
1.1 Ausgangslage	1
1.2 Problemstellung	1
1.3 Zielsetzung	1
1.4 Abgrenzung	2
1.5 Projektmanagement	2
2 Grundlagen	3
2.1 Internet of Things (IoT)	3
2.2 Smartwatches	3
2.2.1 Kurzübersicht von Funktionen	3
2.2.2 Fachbereich Informatik	3
2.3 MQTT	4
2.4 Android und Android Wear	4
2.5 siot.net	5
2.5.1 IoT-Center	5
3 Marktsegmente	6
3.1 Marktsegmente im Internet of Things	6
3.1.1 Industrie	6
3.1.2 Automobil	7
3.1.3 Städte und Verkehr	8
3.1.4 Heimautomation	9
3.1.5 Detailhandel	9
3.1.6 Mensch	9
3.1.7 Natur	9
3.2 Marktsegmente für Smartwatches	11
3.2.1 Mensch	11
3.2.2 Zeit	11
3.2.3 Benachrichtigung	12
3.3 Marktsegmente für Smartwatches im Internet of Things	13
3.3.1 Mensch	13
3.3.2 Benachrichtigung	13
3.3.3 Heimautomation	13
3.3.4 Detailhandel	14
3.3.5 Ortsbezogen	14
4 Bedürfnisanalyse	15
4.1 Smartwatch Applikationen	15
4.1.1 Gesundheit	15
4.1.2 Smart Home	15
4.1.3 Sport	16
4.1.4 Ortsbezogen	16
4.1.5 Authentifikation	17
4.1.6 Finanztechnologie - FinTech	17
4.2 Smartwatch Applikationen für siot.net	17
4.2.1 siot.net Gateway Library	17

4.2.2	siot.net Sensorcenter	18
4.2.3	siot.net Dashboard App	18
4.2.4	Herzfrequenz Überwachung	18
4.2.5	Steuerung von Modellen	18
5	Technische Anforderungen	19
5.1	Allgemeine Applikationen Anforderungen	19
5.1.1	Gesundheit	19
5.1.2	Smart Home	19
5.1.3	Finanztechnologie - FinTech	20
5.2	siot.net Applikationen Anforderungen	20
5.2.1	siot.net Gateway Library	20
5.2.2	siot.net Sensorcenter	20
5.2.3	siot.net Dashboard App	21
5.2.4	Herzfrequenz Überwachung	21
5.2.5	Steuerung von Modellen	22
6	Technologiewahl	23
6.1	Allgemeine Eigenschaften	23
6.2	Kandidaten	24
6.3	Entscheid Smartwatch	25
7	Architektur	26
7.1	Android Wear Paket Architektur	26
7.2	siot.net Applikationsarchitektur	26
7.3	Netzwerk-Architektur	27
7.3.1	geplante Architektur	27
7.3.2	effektive Architektur	29
8	Anforderungsdokumente	31
8.1	Requirements - siot.net Gateway Library	31
8.1.1	Allgemeine Beschreibung	31
8.1.2	User Requirements	31
8.1.3	Funktionale Anforderungen und Anwendungsfallbeschreibungen	32
8.1.3.1	Anwendungsfall #1	32
8.1.3.2	Anwendungsfall #2	33
8.1.3.3	Anwendungsfall #3	34
8.1.3.4	Anwendungsfall #4	35
8.1.3.5	Anwendungsfall #5	36
8.1.3.6	Anwendungsfall #6	37
8.1.4	Nicht-funktionale Anforderungen	37
8.2	Requirements - siot.net Sensorcenter	38
8.2.1	Allgemeine Beschreibung	38
8.2.2	User Requirements	38
8.2.3	Funktionale Anforderungen und Anwendungsfallbeschreibungen	39
8.2.3.1	Anwendungsfall #1.1 und #1.2	39
8.2.3.2	Anwendungsfall #2.1 und #2.2	40
8.2.3.3	Anwendungsfall #3.1 und #3.2	41
8.2.3.4	Anwendungsfall #4.1 und #4.2	41
8.2.4	Nicht-funktionale Anforderungen	42
8.3	Requirements - siot.net Dashboard	43
8.3.1	Allgemeine Beschreibung	43
8.3.2	User Requirements	43
8.3.3	Funktionale Anforderungen und Anwendungsfallbeschreibungen	44
8.3.3.1	Anwendungsfall #1	44

8.3.3.2	Anwendungsfall #2	44
8.3.3.3	Anwendungsfall #3	44
8.3.3.4	Anwendungsfall #4	45
8.3.3.5	Anwendungsfall #5	46
8.3.3.6	Anwendungsfall #6	46
8.3.4	Nicht-funktionale Anforderungen	47
9	Konzepte	48
9.1	Konzept - siot.net Gateway Library	48
9.1.1	Packagediagramm	48
9.1.2	Domänendiagramm	49
9.1.3	Sequenzdiagramme	50
9.1.3.1	App starten	50
9.1.3.2	Login zu siot.net	51
9.2	Konzept - siot.net Sensorcenter	52
9.2.1	Domänendiagramm	52
9.2.2	Sequenzdiagramm	53
9.3	Konzept - Kommunikation Smartphone-zu-Smartwatch	54
9.3.1	Request-Response	54
9.3.2	Fire-and-Forget	54
10	Implementation	55
10.1	Abgrenzung	55
10.1.1	Ziele - siot.net Gateway Library	55
10.1.2	Ziele - siot.net Sensorcenter	55
10.2	Implementation - siot.net Gateway Library	55
10.2.1	Klassendiagramm	55
10.2.2	Logik	57
10.3	Implementation - siot.net Sensorcenter	57
10.3.1	Klassendiagramm	58
10.3.2	Logik	58
10.3.3	Graphical User Interface	59
10.4	Plattformen und Libraries	60
11	Schlussfolgerung	61
12	Ausblick	62
A.	Abbildungsverzeichnis	63
B.	Tabellenverzeichnis	63

1 Einleitung

1.1 Ausgangslage

Die Fachgruppe SIOT des Instituts RISIS der BFH konzipiert und entwickelt zusammen mit Industriepartner die Plattform siot.net, welche Sensoren und Aktoren weltweit mit IoT-Anwendungen (IoT: Internet of Things) verbindet. Smartwatches, welche eine rasante Marktakzeptanz geniessen, spielen eine grosse Rolle im Bereich IoT, denn sie integrieren eine Anzahl von Sensoren und können am Handgelenk Informationen anzeigen. Allerdings gibt es betreffend Funktionalität eine grosse Spannweite bei den Smartwatches, was deren mögliche Einsatzgebiete schliesslich definiert.

1.2 Problemstellung

Die Projektarbeit 2 erlaubte Android Smartwatch zu analysieren. Diese Erkenntnisse sollen genutzt werden um in einer praktischen Umsetzung konkretisiert werden. - Dabei sollend folgende Themen genauer betrachtet werden:

- Welche Anwendungsklassen kann man für Smartwatches erkennen?
- Wie werden Smartwatches am weltweiten Internet angebunden?
- Welche GUI-Elemente werden bereit gestellt?
- Welche Sensoren und Aktoren stehen zur Verfügung?

Die Bachelorarbeit beinhaltet eine Markt- und Bedürfnisanalyse welche die Marktsegmente und die Bedürfnisse aus Sicht IoT für Smartwatch aufzeigen. Für die identifizierten Anwendungen werden Smartwatches evaluiert.

Als weitere Aufgabe wird eine generische System-Architektur definiert, mit welcher Software für Smartwatches für IoT Anwendungen im siot.net Umfeld realisiert werden kann.

In einem formalen Teil werden die Anforderung bzw. technischen Anforderung, einer bestimmten Smartwatch an eine Anwendung gestellt, untersucht und aufgezeigt. Hierbei sollen auch Genauigkeiten und Zuverlässigkeit genauer betrachtet werden.

Es wird ein Software Design erstellt mit welchem 2 bis 3 konkrete Anwendungen implementiert werden könnten.

Daraus wird mindestens eine konkrete Anwendung implementiert. Zur Implementation wird eine Dokumentation erstellt welche von Ingenieuren gelesen wird.

Schlussendlich werden in diesem Dokument alle Ergebnisse berichtet.

1.3 Zielsetzung

Mit einer Bedürfnisanalyse sollen Anwendungsfälle für Smartwatches erarbeitet werden. Mit den entdeckten Use-Cases werden aktuelle Smartwatches evaluiert und mindestens eine wird genauer betrachtet. Um eine geeignete Plattform für die Softwareentwicklung der gewählten Uhr aufzubauen wird eine Entwicklung-, Build und Testumgebung angeschaut und gewählt.

Die erstellten Grundlagen helfen die eine generische Softwarebibliothek zu erstellen. Mit diesem Stück Software wird ermöglicht, Smartwatches und Smartphones schnell an die siot.net Plattform anzubinden. Entwickler von Apps erleichtert dies die Arbeit, denn für die Verbindung an das IoT System kann die Bibliothek verwendet werden. Bei der generischen Anbindung wird das Hauptaugenmerk auf Sensor-, Ortungsdaten und Aktorenaktionen gelegt. Programmierern wird ein Entwicklungshandbuch bereitgestellt, dieses erläutert die Möglichkeiten (JavaDoc) und die Grenzen.

Einige Funktionen werden mit einer Applikation gezeigt, welche die siot.net Anbindungsbibliothek integriert.

1.4 Abgrenzung

Smartwatches im allgemein gibt es von vielen verschiedenen Anbietern. In dieser Arbeit werden aller Arten Smartwatches analysiert. Der Schwerpunkt liegt für das Software Designs und die Implementationen auf Smartwatches, die mit dem Betriebssystem Android ausgeliefert werden. Diese Abgrenzung findet auch statt um die Programmiersprache hauptsächlich auf Java zu beschränken.

Als Prozesssteuerung wird Kanban verwendet. Da dies Arbeit von einer Person durchgeführt wird dämmen die schlanken Zeit- und Prozessplanungsmethoden starken Overhead ein.

1.5 Projektmanagement

1.5.1 Zeitplan

Für die Zeitplanung wird ein tabellarischer Zeitplan verwendet. Der Plan ist im Anhang beigelegt.

1.5.2 Prozesssteuerung - Kanban

Für die Prozesssteuerung wird das Kanban Modell verwendet. Das Kanban Modell kommt ursprünglich aus Japan und heißt Signalkarte. Der Begriff Signalkarte, weil auf einer Tafel der Fortschritt des Projektes sichtbar ist. Entwickelt wurde das System durch den Toyota Konzern, welches für die Fertigung ihrer Produkte dienen soll. David J. Anderson (www.djaa.com) hat das Modell im Jahre 2007 für die Informationstechnologie adaptiert. Bei dieser Prozesssteuerungsart wird eine Prozesskette definiert und dann Aufgaben, welche Tickets genannt werden, ins Backlog erfasst. Das Kanbanboard besteht aus den Spalten aus der Prozesskette und den Status der Tickets. Zur Bearbeitung dieser Bachelor Thesis sind folgende Prozessspalten definiert worden: Backlog, Bereit (Ready), In Arbeit (In Progress), Erledigt (Done). Die Backlog Tickets sind dem tabellarischen Zeitplan zu entnehmen. Typischerweise wird bei einem Kanbanprojekt auch immer eine maximale Anzahl an Ticket ins Bearbeitung definiert. Bei einer Einzelarbeit, wurde darauf verzichtet um den Arbeitsfluss nicht zu hindern. Das Kanbanboard wurde nicht physisch geführt sondern mit einer Webapplikation¹. Diese Applikation bildet den Backlog aus der Versionisierungsablage² ab in die definierte Prozesstafel. Nach beenden eines Tickets werden diese nach 5 Tagen aus dem Prozesssteuerungssystem entfernt um die Übersichtlichkeit hoch zu halten. In der Versionierungsablage werden diese dauerhaft gespeichert mit dem aktuellen status.

¹Kanbanboard: <https://waffle.io/paras1/sw-siot>

²Github Backlog: <https://github.com/paras1/sw-siot/issues>

2 Grundlagen

2.1 Internet of Things (IoT)

Das Internet der Dinge (Internet of Things / IoT) ist eine Struktur, bei der Tiere, Menschen oder Objekte mit einer unverwechselbaren Identität bezeichnet sind. Weiterhin ist damit die Möglichkeit verbunden, Daten über ein Netzwerk ohne Interaktionen Mensch-zu-Mensch oder Mensch-zu-Computer zu übertragen. Das Internet der Dinge hat sich aus der Konvergenz der drahtlosen (wireless) Technologie, MEMS (Micro-Electromechanical Systems) und dem Internet entwickelt.

Ein Ding im Internet der Dinge kann zum Beispiel eine Person mit einem Herzschrittmacher, ein Nutztier auf einem Bauernhof mit einem Biochip-Transponder oder ein Automobil mit eingebauten Sensoren sein. Letzteres könnte eine Warnung auslösen, wenn der Reifendruck zu niedrig ist. Im Prinzip ist jedes vom Menschen geschaffene Objekt ein Kandidat, das sich mit einer IP-Adresse ausstatten lässt und Daten via Netzwerk übertragen kann. Bisher wurde das Internet der Dinge am häufigsten mit M2M-Kommunikation (Maschine-zu-Maschine) bei der Fertigung, sowie der Strom-, Gas- und Öl-Versorgung in Verbindung gebracht. Sind Produkte mit M2M-Kommunikation ausgestattet, werden sie häufig als intelligent oder smart bezeichnet.¹

2.2 Smartwatches

Smartwatches sind kompakte Computersysteme, welche vom Benutzers am Handgelenk getragen werden kann. Diese können viele verschiedene Funktionalitäten mit einem Gerät abdecken. Die Minicomputer sind meist mit einer oder mehreren drahtlos Technologie und verschiedenen Sensoren (Bewegungssensor, Lichtsensor, Herzfrequenzmesser) Aktoren (Bildschirm, Vibrationsmotor) ausgerüstet.

Diese Uhren unterstützen den Träger beim alltäglichen Leben. Gehören zur Gruppe der Wearables und damit zu einem essentiellen Bereich des IoT. Abbildung 2.2 zeigt Modelle der zwei grössten Player dieser Branche. Es sind zur Zeit Apple mit der Apple Watch und Google mit den Android Wear Geräte verschiedener Hersteller.

2.2.1 Kurzübersicht von Funktionen

- | | |
|----------------|--|
| Pulsmessung: | Überwachung des persönlichen Pulses |
| Bewegungen: | Mögliche Bewegungen welche über das Handgelenk ermittelt werden können analysieren |
| Fitness: | Genaue Bewegungen können registriert und in Kombination von Weg und ausgewertet werden |
| Informationen: | Der Träger kann Informationen empfangen welche auf seinem Smartphone ersichtlich sind |

2.2.2 Fachbereich Informatik

Smartwatches gehören in den Bereich der Wearables. Dies ist ein fachübergreifendes Gebiet der Informatik, einige Fachgebiete:

- Ubiquitous Computing, die Rechnerallgegenwärtig
- Pervasive Computing, die Vernetzung von Alltagsgegenständen
- Mobile Computing, mobile Mensch zu Maschinen Kommunikation
- M2M, Machine-to-Machine, Informationsaustausch zwischen Zielgeräten
- IoT, Internet of Things, dass auf den vorhergehenden Fachbereichen basiert

¹<http://www.searchnetworking.de/definition/Internet-der-Dinge-Internet-of-Things-IoT>, 23.10.2015

2.3 MQTT

MQTT wurde im Jahre 1999 von Andy Stanford-Clark (IBM) und Arlen Nipper (damals Eurotech) entwickelt um eine Ölipeline quer durch die Wüste zu überwachen. Das Ziel war ein Protokoll zu erhalten, welches Bandbreiteneffizient ist und wenig Energie konsumiert. Dies musste erreicht werden, weil die eingesetzten Geräte über Satelliten verbunden waren und dies wäre sehr teuer zu dieser Zeit.

Das Protokoll benutzt die publish/subscribe Architektur, im Gegensatz beansprucht HTTP request/response. Publish/subscribe ist ereignisgesteuert und erlaubt, Nachrichten an den Empfänger zu pushen. Der zentrale Kommunikationspunkt ist der MQTT Vermittler, auch MQTT Broker genannt. Dieser hat die Verantwortung die Mitteilung zwischen den Sendern und den richtigen Empfängern zu verteilen. Das Routing geschieht anhand von sogenannten Themen (Topics). Topics sind vergleichbar mit Ordnerstrukturen. Sie beginnen mit einem Thema und werden mit Slashes unterteilt in Unterthemen. Jede Nachricht, beinhaltet ein Topic, dieser dient für den Broker als Verteilschlüssel. Jeder Empfänger abonniert die gewünschten Topics und der Broker stellt ihnen die Meldungen mit dem passenden Wert zu. Das hat den Vorteil, dass Sender und Empfänger gegenseitig nicht bekannt sein muss. Diese Architektur erlaubt in hohen Massen skalierbare Lösungen ohne Abhängigkeiten zwischen Datenproduzent und Konsument.

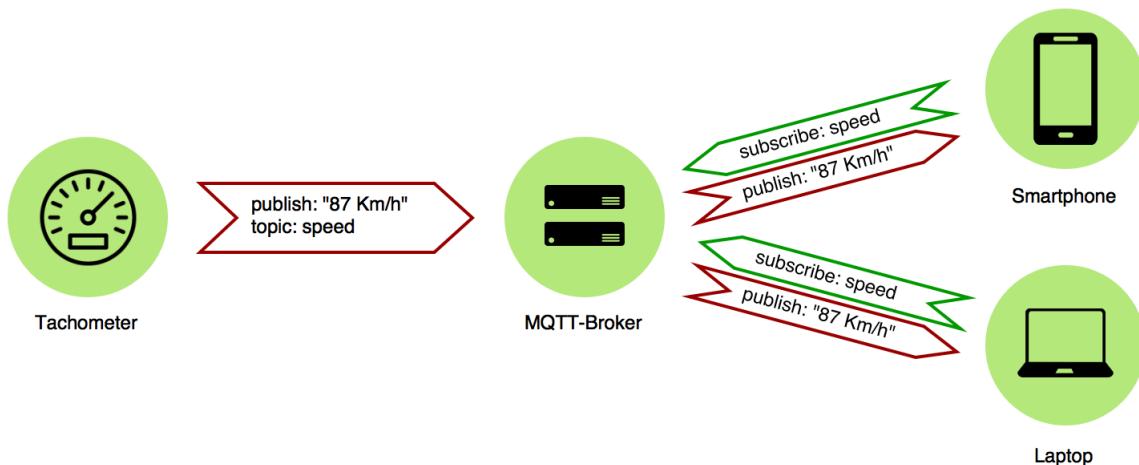


Abbildung 2.1: Die Publish/Subscribe Architektur von MQTT

Der Unterschiede zum HTTP Protokoll ist, dass ein dienstanforderndes Gerät die Informationen nicht holen muss, sondern direkt liefert bekommt. Voraussetzung dafür ist eine immer offene TCP Verbindung vom Client zum Server. Falls diese Konnektion unterbrochen werden sollte, kann der MQTT Broker Nachrichten zwischenlagern und diese dann pushen, wenn dieser wieder Verfügbar ist².

2.4 Android und Android Wear

Google's Android ist das marktführende Betriebssystem³ welches in aller Munde ist. Es betreibt Smartphones und Tablets unterschiedlicher Hersteller und ist ein Open-Source Produkt. Android Wear ist das OS von Google für Wearables, wie Smartwatches. Es basiert auf Android, ist optimiert für kleine Geräte mit weniger Leistung und kurzer Akkulaufzeit.

²vgl. <http://www.searchnetworking.de/definition/Internet-der-Dinge-Internet-of-Things-IoT>, 20.11.2015

³vgl. <http://de.statista.com/statistik/daten/studie/182363/umfrage/prognostizierte-marktanteile-bei-smartphone-betriebssystemen/>, 10.01.2016

2.5 siot.net

Die Fachgruppe SIOT des Instituts RISIS der BFH konzipiert und entwickelt zusammen mit Industriepartner (AppModule AG) die Plattform siot.net, welche Sensoren und Aktoren weltweit mit IoT-Anwendungen verbindet. Das Ziel dieser Plattform ist es sie zu industrialisieren um Klein und Mittlere Unternehmen (KMU) eine Möglichkeit zu geben ihre Sensoren und Geräte zu vernetzen. Dies soll vorteilhafterweise zu einem günstigen Preis möglich sein. siot.net bietet für eine kleine Gebühr ein IoT-Center und die IoT Infrastruktur. Konfiguration, Analyse, Verwaltung und Zuweisungen werden alle im IoT-Center durchgeführt. Über das IoT-Center können alle Sensoren und Aktoren verknüpft und konfiguriert werden. Um die Daten auszuwerten und darzustellen wird ein Dashboard zur Verfügung gestellt. Die IoT-Infrastruktur beinhaltet einen MQTT Broker und die Definition, wie Meldungen auf den siot.net Broker angeliefert werden sollen.



Abbildung 2.2: siot.net Plattform Logo

2.5.1 IoT-Center

3 Marktsegmente

Im Kapitel Marktsegmente werden die aktuellen Bereiche von Internet of Things, Smartwatches und Smartwatches im Internet of Things aufgezeigt. Es werden nur Bereich aufgezeigt in welchen die erwähnten Themen eine Rolle spielen. Es ist keine strategische Marktsegmentierung durchgeführt worden und es ist keine abschliessende Auflistung.

3.1 Marktsegmente im Internet of Things

Industrie:	Maschinensteuerung, Automatisierte Roboter, Lagerüberwachung
Automobil:	Telemetrie, Geografische Strecke, Fahrverhalten, Nutzungsverhalten, Verkehrsbericht
Städte/Verkehr:	Touristisches Informationen, Dynamische Straßen, Verkehrsregulierung, Navigation, Lageberichte
Heimautomation:	Nutzung und Überwachung von Haushaltsgeräte, Steuerung, Fernbedienungen
Detailhandel:	Produktebezeichnung, Kasse, Geldüberweisung, Geldbörse
Mensch:	Blutdruck, Puls, Bewegungen, Schlaf Überwachung, Körperanalyse (z.B. Gewicht, Fettanteil, Wasseranteil usw.)
Natur:	Erdplatten Bewegung, Wasserspiegel Überwachung, Temperatur, Wind, Licht, Luft

Wie in der Tabelle aufgelistet, kommt das Internet der Dinge in sehr vielen verschiedenen Marktsegmenten zum tragen. Es hat noch grösseres Potential den Mensch zu unterstützen und ihre Aufgaben zu erleichtern, als es jetzt schon tut.

3.1.1 Industrie

Das Internet der Dinge kommt in der Industrie soweit zum tragen, dass man von Industrie 4.0 spricht. Dies soll die vierte industrielle Revolution zum Ausdruck bringen. Die Fertigungstechnologie soll informatisiert werden. Auch die Logistik soll ihre Automatisierung erleben. Erreicht wird dies weil Maschinen untereinander kommunizieren können. Möglichst alle Sektoren einer Fabrik sollen vernetzt sein. Das Ziel der Industrie 4.0 ist die intelligente Fabrik.

3.1.2 Automobil

IoT kann in vielen Bereichen der Automobilbranche eingesetzt werden. Es können wichtige Daten des Fahrzeugs ausgelesen werden, z.B. die Telemetriedaten. Diese können dann verwendet werden um das Fahrverhalten vom Lenker festzustellen. Des Weiteren können diese auch benutzt werden um Probleme beim Auto auszumachen und direkt Fahrer und Mechaniker zu alarmieren. Interessant, für die Autobauer wie auch Autobesitzer, ist auch die Ortung der Fahrzeuge. Mit den aufgezeichneten geografischen Punkten kann analysiert werden, wie das Automobil verwendet wird und aktuelle verkehrsnahe Verkehrsberichte können genutzt werden. Die Vollendung der Vernetzung von Fahrzeugen ist das selbstfahrende Auto, welches alle nötigen Informationen empfängt, analysiert und verwendet um das Ziel zu erreichen.

Mercedes-Benz hat ein selbstfahrendes Forschungsfahrzeug entwickelt (Siehe Abbildung 3.2). Die Abbildung 3.3 zeigt wie das Fahrzeug durch die Sensoren einen Fußgänger erkennt und die Laserprojektionstechnik als Aktor verwendet, um dem Überquerenden die Fußgängerstreifen aufzuzeigen.



Abbildung 3.1: Das selbstfahrende Forschungsfahrzeug von Mercedes-Benz (Modell F015 In Motion)

Quelle:

http://www.mercedes-benz.ch/content/media_library/f_015_luxury_in_motion_layer-gallery_1_01__710x396_01-2015.jpg,
Stand: 05.11.2015

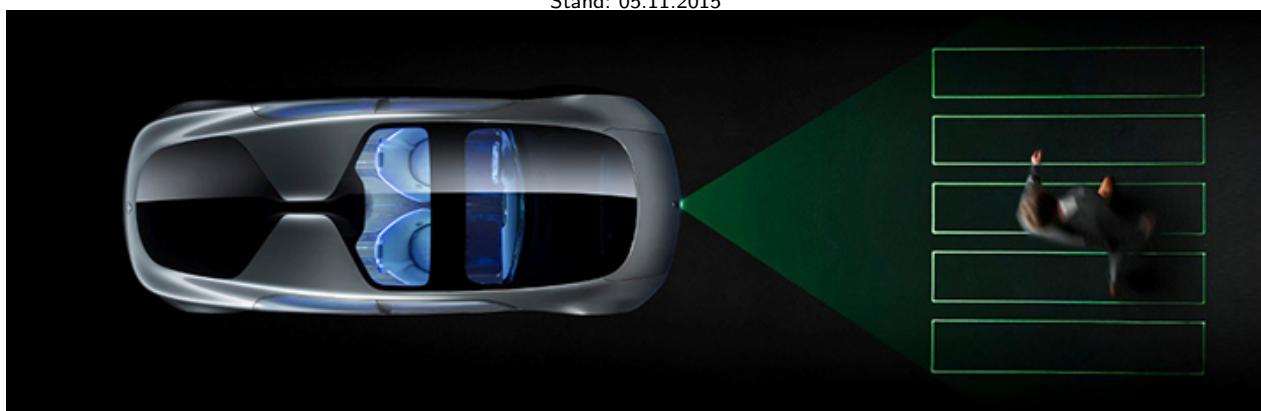


Abbildung 3.2: F015 In Motion erkennt Fußgänger mit seinen Sensoren

Quelle: http://www.mercedes-benz.ch/content/media_library/f_015_luxury_in_motion_gallery_05_715x230_01-2015.jpg, Stand:
05.11.2015

3.1.3 Städte und Verkehr

In Städten gibt es sehr viele Möglichkeiten, in Verbindung mit dem Verkehr geht dies ins unermessliche. Ein sehr interessantes Thema ist die Touristik. Um ein Beispiel zu nennen, Beacons welche nötige Information an ein Smartgerät publizieren um Daten von der Sehenswürdigkeit abzurufen. Dazu könnte auch gleich Empfehlungen in der Umgebung notifiziert werden. So würde für die meisten Reisenden der Reiseführer wegfallen.

Ein spektakuläres Projekt ist die dynamische Strasse: **Solar Roadways**.

Das sind kleine, feste Platten mit Photovoltaik-Elementen, Elektronik, verschiedenen Sensoren und LEDs integriert (Siehe Abbildung 3.4). Die Platten können wie Pflastersteine verlegt und miteinander verbunden werden. Durch die Sonneneinstrahlung sind sie permanent und umweltschonend mit Strom versorgt. Die LEDs können zentral gesteuert werden, um so die Fahrbahnmarkierungen anzuzeigen und z.B. aus zwei breiten Spuren drei schmale zu machen oder spontane Parkflächen oder Verkehrszeichen oder was auch immer. Die Sensoren können feststellen, wenn Tiere darüber laufen und die Fahrer blitzschnell schon ein paar hundert Meter vorher über die LEDs warnen. Und die Platten sind beheizbar. Kein übermassiges Ausbringen von Streusalz mehr, keine Glatteisunfälle mehr, keine Asphalterneuerungen mehr und vermutlich viel weniger Baustellen. Austauschen defekter Platten ist ohne weiteres möglich. Es ist von einer Privatpersonen initiiert und gecrowdfunded. Das Vorhaben ist auf Indiegogo (www.indiegogo.com) im Juni 2014 deutlich überfinanziert abgeschlossen worden. In Holland hat man im Jahr 2014 begonnen, Radwege auf diese Weise zu bauen.¹



Abbildung 3.3: Das Solar Roadway verändert die Strasse für die aktuelle Verkehrssituation

Quelle: <http://www.solarroadways.com/images/intro/Downtown%20Sandpoint%20-%20small.jpg>, Stand: 05.11.2015

¹vgl. <https://www.holisticon.de/2014/11/die-sache-mit-dem-horizont-iot-blogserie-episode-5>, Stand: 05.11.2015

3.1.4 Heimautomation

Die Heimautomation ist auch besser bekannt als Smart Home. Smart Home dient als Oberbegriff für technische Verfahren und Systeme in Wohnräumen und -häusern, in deren Mittelpunkt eine Erhöhung von Wohn- und Lebensqualität, Sicherheit und effizienter Energienutzung auf Basis vernetzter und fernsteuerbarer Geräte und Installationen sowie automatisierbarer Abläufe steht.

Unter diesen Begriff fällt sowohl die Vernetzung von Haustechnik und Haushaltsgeräten (zum Beispiel Lampen, Jalousien, Heizung, aber auch Herd, Kühlschrank und Waschmaschine), als auch die Vernetzung von Komponenten der Unterhaltungselektronik (etwa die zentrale Speicherung und heimweite Nutzung von Video- und Audio-Inhalten).

Von einem Smart Home spricht man insbesondere, wenn sämtliche im Haus verwendeten Leuchten, Taster und Geräte untereinander vernetzt sind, Geräte Daten speichern und eine eigene Logik abilden können. Geräte sind teilweise auch getagged, was bedeutet, dass zu den Geräten im Smart Home Informationen zum Beispiel über Hersteller, Produktnamen und Leistung hinterlegt sind. Dabei besitzt das Smart Home eine eigene Programmierschnittstelle, die (auch) via Internet angesprochen und über erweiterbare Apps gesteuert werden kann.

Eng verwandt mit diesen Verfahren und Systemen sind solche des Smart Metering, bei denen der Schwerpunkt auf dem Messen und einer intelligenten Regulierung des Energieverbrauchs liegt.

Neben „Smart Home“ haben sich Begriffe wie Intelligentes Wohnen, „eHome“, „Smart Living“ und weitere Bezeichnungen etabliert, die sich teils nur in Bedeutungsschattierungen unterscheiden. Zudem verwenden Hersteller von Smart-Home-Anlagen und -komponenten weitere, speziell auf deren individuelles Marketing abgestimmte Begriffe.²

3.1.5 Detailhandel

Im Verkauf hat die Revolution schon teilweise begonnen. Die grossen Player in der Schweiz beginnen alle ihre Filialen mit WLAN auszustatten. Momentan bieten diese den WiFi Zugang gratis den Kunden an. Somit steht der Kommunikationskanal für das IoT im Markt bereit und die Kunden sind zur gegebenen Zeit bereits verbunden damit. Weiter werden bereits Selbstbezahlkassen eingesetzt. Momentan werden zwei Modelle verfolgt. Eine Einkaufsart ist der Kunde wählt seine Produkte und scannt diese selber ein, bezahlt mit Karte oder Bar und verlässt das Geschäft. Die dem IoT nähren Methode registriert sich die einkaufende Person sich beim Eingang an einem Terminal und rüstet sich mit einem mobilen Strichcodeleser des Marktes aus oder nimmt sein Smartphone als Scanner. Die Person liest alle Produkte mit dem Scanner ein und bezahlt beim verlassen des Ladens beim Bezahlterminal. Beim abmelden des Scanners wird der Kunde ermittelt und das Total eingefordert.

Ein weiterer Schritt ist hier alle Produkte mit einer RFID zu taggen, somit könnte der Kunde nur seine Kreditkarte registrieren, die Waren in den Warenkorb legen und die Filiale verlassen. Beim verlassen wird, durch die Information auf dem RFID Tag, gemerkt welche Waren mitgenommen wurden und die Kreditkarte wird automatisch belastet.

3.1.6 Mensch

Der Mensch ist ein wichtiges Marktsegment, hierbei können Sensoren aller Arten den Menschen analysieren. Dieser Punkt wird bei der Marktsegmentierung Smartwatches und Smartwatchen im Internet of Things genauer betrachtet.

3.1.7 Natur

Viele verschiedene Anwendungsfälle gibt es auch in der Natur. Es können Sensoren eingesetzt werden um Temperaturen, Luftdruck, Luftfeuchtigkeit oder Windstärke zu messen. Mit Kombinationen von Sensoren, welche miteinander kommunizieren, können Frühwarnsysteme von Naturkatastrophen erschaffen werden. Dieses Segment hängt sehr nahe mit dem Marktsegment des Menschen zusammen.

²Quelle: https://de.wikipedia.org/wiki/Smart_Home, Stand: 05.11.2015

Earthquake Early Warning Basics

1 In an earthquake, a rupturing fault sends out different types of waves. The fast-moving P-wave is first to arrive, but damage is caused by the slower S-waves and later-arriving surface waves.

2 Sensors detect the P-wave and immediately transmit data to an earthquake alert center where the location and size of the quake are determined and updated as more data become available.

3 A message from the alert center is immediately transmitted to your computer or mobile phone, which calculates the expected intensity and arrival time of shaking at your location.

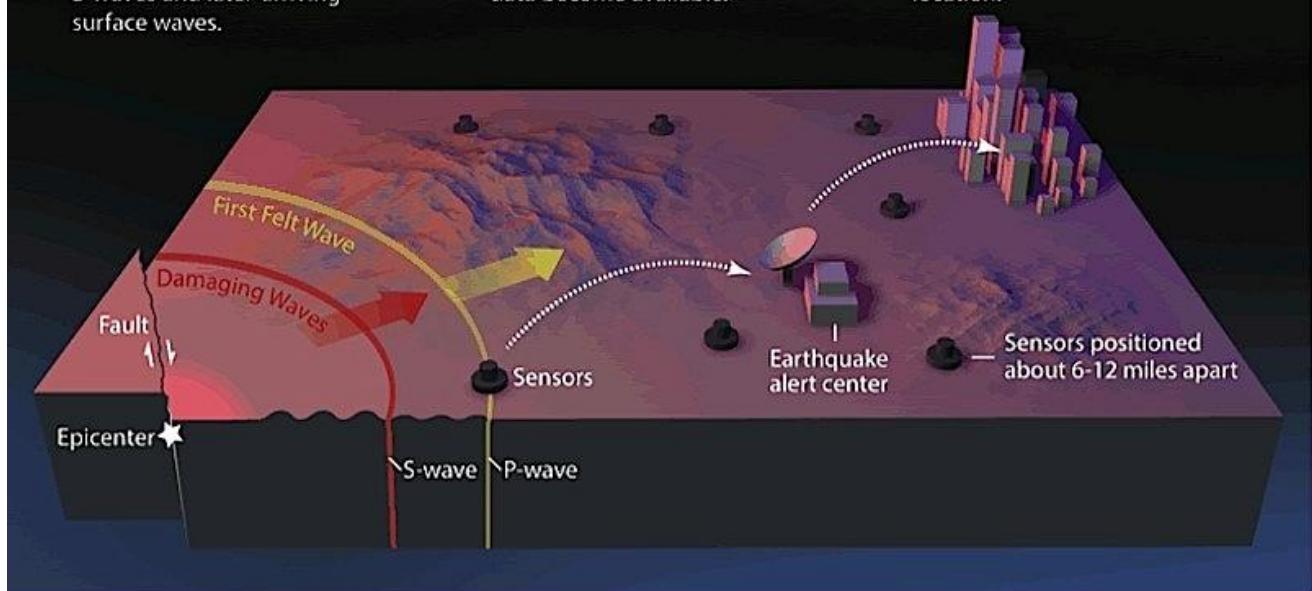


Abbildung 3.4: Wie ein Frühwarnsystem von Erdbeben funktioniert

Quelle:

http://www.ingenieur.de/var/storage/images/media/ingenieur.de/bilder/funktionsweise-fruehwarnsystems-shakealert/3666615-1-ger-DE/Funktionsweise-des-Fruehwarnsystems-ShakeAlert_image_width_884.jpg, Stand: 05.11.2015

3.2 Marktsegmente für Smartwatches

Mensch:	Blutdruck, Puls, Bewegungen, Schlaf Überwachung, Lebensüberwachung, Sportbeobachtungen, Sporttracking
Zeit:	Individuelle Zeitansichten, Zeitfunktionen
Benachrichtigung:	Informationen am Handgelenk, Kommunizieren
Momentan werden Smartwatches hauptsächlich zu Notifikationszwecke und Analysen des Menschen genutzt. Noch ist das Potenzial unerschöpft.	

3.2.1 Mensch

Heute werden Smartwatches verwendet, um den Menschen bei Aktivitäten überwachen zu können. Diese Wearables verfügen viele eingebaute Sensoren die die Bewegungen des Trägers analysieren und dem interessierten die Daten zur Verfügung stellen. Zu den Sensoren gehören z.B. ein Bewegungssensor, Schrittzähler, Herzfrequenzmesser und viele mehr. Viele Hersteller von Smartwatches rüsten Ihre Produkte mit Sensoren und auch gleich Auswertungsapplikationen (z.B. Google Fit, Apple Health oder Motorola Moto Body). Mit diesen Apps kann der User seine Daten während dem Training auf der Uhr verfolgen oder später auf dem Smartphone auswerten. Dies macht zusätzliche Sport-/Pulsuhren überflüssig.

3.2.2 Zeit

Die Hauptaufgabe einer Uhr sollte es immernoch sein die Zeit genau anzuzeigen. Die Smartwatches haben die Möglichkeit nicht nur die aktuelle Uhrzeit anzuzeigen sondern auch als Weltuhr, Stoppuhr und Countdown Rechner zu fungieren. Dabei hat der Träger der Computeruhr die Wahl wie das Ziffernblatt aussehen soll, die Abbildung 3.10 verdeutlicht dies.



Abbildung 3.5: Das Ziffernblatt der meisten Smartwatches kann individuell gestaltet werden

Quelle: <http://i1-news.softpedia-static.com/images/news2/>

Google-Launches-Watch-Face-API-You-Can-Customize-Your-Smartwatch-467130-2.jpg, Stand: 12.11.2015

3.2.3 Benachrichtigung

Eine Smartwatch wird neben der Uhrzeitfunktion auch als Notifikationsbildschirm verwendet. Alle relevanten Benachrichtigung an ein Smartphone können auch von der Smartwatch angezeigt werden. Dabei dient die Uhr meist als verlängerter Arm der Mobilgerätes. Es können Nachrichten empfangen, Telefonate geführt, Erinnerungen ausgelöst, der Wecker gestellt werden oder anzeigen was auf dem Smartphone ausgeführt wird (Siehe Abbildung 3.11).



Abbildung 3.6: Die Smartwatch zeigt an, welches Musikstück auf dem Smartphone abgespielt wird
Quelle: http://smartwatchpro.it/wp-content/uploads/2015/08/0B-YT760_smartd_M_20130904020012.jpg, Stand: 12.11.2015

3.3 Marktsegmente für Smartwatches im Internet of Things

Mensch:	Blutdruck, Puls, Bewegungen, Schlaf Überwachung, Gesundheitsbenachrichtigung
Benachrichtigung:	Alarne, Informationen
Heimautomation:	Fernbedienung, Statusanzeigen, Alarming
Detailhandel:	Geldbörse, Produktebezeichnung, Einkaufsliste
Ortsbezogen:	Navigation, Ortsspezifische Informationen, Ortung, Personen in der Nähe

3.3.1 Mensch

Um die Gesundheit eines Mensch zu überwachen, eignet sich eine Smartwatch sehr gut. Sie ist immer dabei und kann bei Unregelmässigkeiten Alarm schlagen. Durch die Vernetzung werden die Daten auch anderen Geräten zur Verfügung gestellt. Eine wichtige Benachrichtigung kann auch von einer anderen Smartwatch oder Smartphone verwendet werden. Ein sich vorstellbares Szenario: Bei einem Paar sind tragen beide eine smarte Uhr bei potenziellen gefahren, der andere alarmiert wird.

3.3.2 Benachrichtigung

Eine Smartwatch eignet sich genau so gut wie ein Smartphone oder andere Anzeigegeräte um Informationen darzustellen. Die im Internet der Dinge übermittelten und verwerteten Daten werden, für den Menschen, auf einem praktisch sichtbaren Display angezeigt.

Auf der Abbildung 3.12 sind aktuelle Herzfrequenz, Aktivität und die Dauer abgebildet. Diese Meldungen könnten auch auf einem anderen Bildschirm projiziert werden.



Abbildung 3.7: Die Smartwatch zeigt an, welche Herzfrequenz er hat und wie lange er eine Tätigkeit ausführt
Quelle: https://support.apple.com/images/en_US/applewatch/watch-indoor-workout-heartrate.png, Stand: 20.11.2015

3.3.3 Heimautomation

In Smart Home Bereich kann die Kombination Smartwatch und Internet of Things ihre stärken ausspielen. Durch den Zusammenschluss aller Haushaltgeräte, wie Fernseher, Lampen oder Herdplatte, sind alle Daten zentral erreichbar und verwaltbar. Nun bestehen viele Möglichkeiten die Computeruhr ins System einzubinden. Einige geeignete Anwendungsfälle sind, das Licht vom Handgelenk ein und auszuschalten, wenn Haushaltsgesäte nicht ausgeschaltet wurden diese alarmieren, die Geräte fernbedienen und dies unabhängig vom Ort wo der Träger der Uhr ist.

3.3.4 Detailhandel

Im Detailhandel sind besonders Finanztechnologie Applikationen schon stark vertreten mit Apple Pay, Google Wallet oder auch die schweizerische Lösung TWINT. Diese Anwendungen erlauben Geldüberweisungen mit Smartphones oder Smartwatches mittels drahtloser Verbindung über ein Terminal, welches die Bezahlung anfordert.

Des Weiteren können für Händler von Selbstbedienungsgeschäften Smartwatches als erweiterter Informationseinblender von Produkten sein. Durch die Anbindung ans Internet, hat man die Möglichkeit, jedes weitere Detail welches nicht auf der Ware beschrieben ist, auf dem Bildschirm am Arm anzuschauen. Ein sehr grosser Vorteil ist, dass verschiedene Medien benutzt werden können, z.B. detailliert Nährwertangaben, Anleitungsfilme aller Art, Explosionszeichnungen von Modellen und viele weitere.

3.3.5 Ortsbezogen

Applikationen für die Ortung oder zur Anzeige standortabhängige Daten wird fast überall benutzt. Dies auf die Smartwatch zu erweitern ist ein logischer Schritt. Ortsrelevantes Thema kann direkt am Handgelenk angeschaut werden ohne das Smartphone heranzuziehen (siehe Abbildung 3.13).



Abbildung 3.8: Landkarten anzeigen und Navigation ist auf Smartwatches möglich
Quelle: <http://icdn9.digitaltrends.com/image/heremaps-samsung-gear-s-720x480.jpg?ver=1>, Stand: 20.11.2015

4 Bedürfnisanalyse

Im Kapitel der Bedürfnisanalyse werden Anwendungsfälle ermittelt welche mit Smartwatches abgedeckt werden können. Zusätzlich wird eine Bedarfsanalyse durchgeführt für Applikationen welche in Verbindung zur siot.net Plattform stehen.

4.1 Smartwatch Applikationen

4.1.1 Gesundheit

Im Gesundheitssektor gibt es einige Anwendungsfälle welche mit einer smarten Uhr abgedeckt werden können.

Eine Smartwatch bietet die Möglichkeit sich zu überwachen. Da die Computeruhr mit vielen Sensoren, wie z.B. Bewegungssensor oder Herzfrequenzmesser, ausgerüstet ist, hat sie die Möglichkeit den Träger sehr genau zu analysieren.

Bei einem Sturz des Benutzers kann ein Alarm ausgelöst werden. Dieser würde in erster Instanz eine positive Gesundheitsbericht des Anwenders verlangen. Diese Bestätigung sollte in einem definierten Zeitrahmen statt finden. Falls dies nicht ausgeführt wird und die Uhr keine Bewegung registriert, kann ein Alarm an eine Vertrauensperson oder gar ein Notruf ausgelöst werden. Dieser Notruf kann wichtigen Daten angereichert werden, wie z.B. Pulswerte und die GPS Koordinaten. Statt des Sturzes kann hier der Auslöser des Alarms auch ein zu tiefer oder gar kein Puls sein.

Ein weiterer nützlicher Use-Case ist, Alarme von Patienten im Spital. Hier kann das Pflegepersonal mit Smartwatches, Patientenalarme erhalten. Die Alarme sollten möglichst nur empfangen werden, wenn der Patient in der Nähe des Patienten sich befindet. Wenn mehrere Pfleger/innen benachrichtigt werden, kann eine Pflegeperson den Alarm bestätigen und die Verantwortung für den Patienten übernehmen, so können Doppelalarme vermieden werden.

4.1.2 Smart Home

Für die Fernbedienung von Geräten im Haus oder Wohnung eignet sich die Smartwatch gut. Mit eingebauten Touchscreen und Vibrationsmotor, haben die kleinen Handgelenkrechner die Möglichkeit Informationen visuell wie taktil an die Person zu bringen.

Geräte im Haushalt können überwacht werden. Dies hilft Gefahren abzuwenden. Wenn eine Herdplatte noch läuft kann ein Alarm ausgelöst werden und es kann gleich mit der Uhr reagiert werden und die Platte ausschalten.

Für jeden einen Mehrwert gibt die Funktion Licht vom Handgelenk zu bedienen. Es ist bequem das Zimmer zu beleuchten ohne zum Lichtschalter gehen zu müssen. Dimmen mit dem Touchscreen und terminierte Lichtsteuerung. Auch eine automatische Beleuchtung durch erkennen der Helligkeit im Raum ist eine Funktion mit hohem Potenzial.

Ein weiterer Anwendungsfall ist die Waschmaschine. Die Restzeit des Waschganges kann auf den Bildschirm angezeigt werden und wenn er beendet ist, wird der Träger mit einem Vibrationsimpuls notifiziert.

Des Weiteren ist das Fernbedienen von allen Multimediageräten vom Handgelenk sehr praktisch. Es genügt eine Uhr und braucht nicht mehr verschiedene proprietäre Steuerungen. Dies wird heute bereits mit Smartphone Apps praktiziert. Mit Sprachsteuerung können Personen mit eingeschränkter Sehkraft die Uhr verwenden.

4.1.3 Sport

Heute werden Smartwatches hauptsächlich als Fitnesstracker verwendet¹. Das Praktische an den Uhren unter den Wearables ist, dass diese nicht nur für zum Sport treiben gekauft werden muss. Hier erhält der Endkunde ein Gerät für den Alltag und die Freizeit.

Im Sportbereich kann mit den vorhandenen Sensoren viele verschiedene Werte ermittelt und analysiert werden. Mit den nötigen Voreinstellungen, wie Körpermasse, Schrittlänge, Alter und Geschlecht, ist es möglich Bewegungsdaten genau aufzuzeichnen. Mit den Daten können für den Anwender interessante Informationen berechnet werden. Für Hobbysportler meist relevante Berechnungen sind Zeit, Schritte, Geschwindigkeit und Kalorienverbrauch. Für erfahrene Sportler verbessern ihre Fähigkeiten durch betrachten von Auswertungen der Körperbelastungen, z.B. Beschleunigung, Stärke, Drehmoment und weitere.

4.1.4 Ortsbezogen

Applikationen welche umgebungsorientiert arbeiten, sind geeignete Kandidaten für Smartwatches. Durch die permanente Anzeige am Handgelenk, können schnell ändernde Daten dauerhaft im Auge behalten werden.

Ein Anwendungsfall ist, das Smartphone zu überwachen. Die Uhr kann den Träger informieren, wenn das Sichtbarkeitsumfeld vom Mobiltelefon und des tragbaren Rechners sich nicht mehr überschneiden, was bedeuten würde die Geräte entkoppeln sich voneinander.

Die gleiche Methode bietet sich an, Personen mit Smartwatches in der Nähe zu scannen. Diese Funktion kann bei Partnervermittlungsapplikationen effektiv eingesetzt werden. Durch den vorhandenen Touchscreen können potenzielle Datingpartner angezeigt und kontaktiert werden. In der schnellen Welt sind sich schnell erschliessende Kontakte sehr willkommen.

Ein weiterer Punkt ist Geofencing. Mit Geofencing werden automatische Aktionen eingeleitet. Diese geschehen sobald bestimmte geografische Grenzen überschritten werden. Über eine Ortung des Gerätes kann ermittelt werden ob dieses sich in einer Geofencing Zone befindet. Mit Smartwatches kann dies effektiver genutzt werden, da durch den immer sichtbaren Bildschirm, die ortsrelevanten Daten ohne Zeitverlust angezeigt.

Eine Geofencing Anwendung wäre Sehenswürdigkeiten präsentieren. Eine App welche für jede Sehenswürdigkeit eine Geofencing Zone errichtet oder kennt und jeweils die Auskunft des Objektes darstellt.

Die Indoornavigation ist ein Bedürfnis welches mit Smartwatches nicht gelöst jedoch erweitern kann. In Zusammenarbeit mit Beacons/Eddystones und/oder Access Points können die Standorte von Smartwatchträger, im inneren von Räumen, ermittelt werden. Beacons und Eddystones sind Sender/Empfänger welche auf Bluetooth Low Energy (BLE, Bluetooth 4.0 oder Bluetooth Smart).

Grossfirmen können eine grossen Nutzen aus dieser Technologiekombination schöpfen. Die Mitarbeiter ihren Standort preisgeben, damit diese gefunden werden können ohne zu suchen. Es können Geofencing Zonen definiert werden um die Zeiterfassung zu automatisieren. Beim eintreten, des geografisch definierten Bereichs, welcher zu Arbeitszone gehört, wird Arbeitszeit erfasst. Verlässt der Mitarbeiter diesen Teil des Gebäudes, wird die Arbeitszeiterfassung gestoppt.

Zusätzlich kann das Problem mit den Shared-Desk Arbeitsplätzen gelöst werden. Bei diesem Arbeitsplatzmodell richtet sich der Mitarbeiter jeden Tag an dem Ort ein wo es einen freien Platz hat. Da dieser nicht im vorherein weiß wo der nächste freie Platz ist, führt dies zu Verlusten, wenn Zeit benötigt wird um einen Arbeitsplatz zu suchen. Mit der Smartwatch ausgetattet, ist der Angestellte in der Lage, vorgängig einen freien Arbeitsplatz zu reservieren und sich anzumelden, beim erreichen des Schreibtisches.

¹vgl. Studienband Smartwatch Umfrage eResult, Ausgabe: Juni 2015

4.1.5 Authentifikation

Im Authentifikationssektor gibt es einige Bedürfnisse welche mit Smartwatches gedeckt werden können

Türen entriegeln mit der Smartwatch ist ein geeigneter Anwendungsfall. Heute benutzen die meisten Automobilhersteller ein Keyless System für ihre Fahrzeuge. Mit diesen Systemen hat der Fahrer nur ein Sender/Empfänger, mit einem einzigartigem Zertifikat, welches sich mit dem Fahrzeuges korreliert. Türen werden geöffnet und Motoren werden gestartet, diese Funktion könnte auch die Smartwatch übernehmen. Damit hätte der Fahrzeugbesitzer ein Utensil weniger zu mitzutragen.

Die intelligente Uhr hat das Potenzial Personalausweise zu ersetzen. Wie bereits im Abschnitt Ortsbezogen erwähnt kann es zur Zeiterfassung genutzt werden. Das heisst Mitarbeiter muss nicht mehr an die Zeiterfassungsleser.

4.1.6 Finanztechnologie - FinTech

Die Möglichkeit zu haben mit der Uhr Zahlunge zu authorisieren ist ein Bedürfnis, welches erschaffen werden kann. Denn es scheint praktisch einzukaufen gehen ohne das Portmonee dabei zu haben. Es sind Lösungen vorhanden, welche mit Smartphones funktionieren (Abbildung 4.1 Apple Pay/Google Wallet/TWINT). Mit der Lancierung der Apple Watch, erreichte die erste Smartwatch mit einer Zahlfunktion, sie unterstützt Apple Pay, den Markt.

4.2 Smartwatch Applikationen für siot.net

Die siot.net Plattform bietet sich bestens als Kommunikationsschnittstelle an für die Applikationen, welche im vorherigen Abschnitt ermittelt wurden. Die meisten Bedürfnisse verlangen irgendeine Art von Kommunikation. Ob es nur übermitteln der Sensordaten ist oder die Abfrage eines Sicherheitstokens ist, die siot.net Plattform erlaubt sämtliche Informationen über eine Schnittstelle auszutauschen. Damit es einfacher wird sollten möglichst viele oder besser alle Geräte, welche miteinander kommunizieren sollen, an die siot.net Plattform angebunden werden.



Abbildung 4.1: Bekannte Zahlungslösungen für Smartphone: Apple Pay, Google Wallet und TWINT powered by PostFinance
<http://www.apple.com/apple-pay/> <https://wallet.google.com/> http://www.twint.ch/ueber_uns/medien/, 04.12.2015

4.2.1 siot.net Gateway Library

Um Verknüpfungen individueller Applikationen von Smartwatches oder auch Smartphones mit der Plattform zu ermöglichen, sollte es eine generische Bibliothek geben. Diese sollte eine einfache Schnittstelle implementieren, welche Applikation an die siot.net Plattform anbindet. Eine automatische Erkennung aller Sensoren vereinfacht die Entwicklung von Apps welche zum siot.net IoT-Center kommunizieren wollen.

4.2.2 siot.net Sensorcenter

Jedes Android Gerät kann mit den eingebauten Sensoren hervorragend als Sensorstation dienen. Um die Vielzahl von Sensor in Eigenregie zu im siot.net zu manifestieren und Sensordaten preiszugeben, sollte es ein App geben mit einer möglichst einfachen Benutzeroberfläche.

4.2.3 siot.net Dashboard App

Um Auswertung und Darstellung von den Daten zu haben, gibt es von siot.net bereits eine Dashboard Webapplikation. Um diese Funktionen einem Smartphone, in einer kompakten Form, zur Verfügung zu stellen, eignet sich eine App. Diese App sollte von Vorteil durch den Benutzer konfigurierbar sein.

4.2.4 Herzfrequenz Überwachung

Vielen Personen, welche ihren Puls im Griff haben wollen, leichte gesundheitliche Probleme haben oder jemanden alarmieren will bei Unregelmässigkeiten bei der Herzfrequenz, würden eine Herzfrequenz Überwachung begrüssen. Diese Anwendungsfälle kann mit einer Smartwatch und der siot.net Plattform abgedeckt werden.

4.2.5 Steuerung von Modellen

Modellbau und Smartphones ist keine Weltneuheit, es wird schon rege verwendet, wie bei der Parrot bebop Drohne² (siehe Abbildung 4.4). Das Potenzial mit einer Steuerung (Smartphone oder Smartwatch) mehrere Modelle simultan zu steuern, scheint jedoch noch nicht im grossen Stile abgerufen zu werden. Mit der Anbindung von Modellen an das Internet der Dinge, zusätzlich gekoppelt mit einem einer smarten Steuerungseinheit, kann dieses Bedürfnis abgedeckt werden. Die siot.net Umgebung bietet dazu eine günstige Ausgangslage.



Abbildung 4.2: Parrot bietet Drohnen an, welche mit dem Smartphone gesteuert werden können: Parrot bebop
https://s.gravis.de/p/z1/parrot-bebop-drone-kamera-drohne-fuer-smartphones-tablets-gps-blau_z1.jpg, 04.12.2015

²Quelle: https://s.gravis.de/p/z1/parrot-bebop-drone-kamera-drohne-fuer-smartphones-tablets-gps-blau_z1.jpg, Stand: 04.12.2015

5 Technische Anforderungen

In diesem Kapitel werden die Technischen Voraussetzungen, für die in der Bedürfnisanalyse ermittelten Anwendungen, definiert. Dabei wird unterschieden für Allgemeine Applikationen, welche nur eine kleine Auswahl beachtet wird, und für siot.net Applikationen.

5.1 Allgemeine Applikationen Anforderungen

Die Technischen Anforderungen für die Bedürfnisse Gesundheit, Smart Home und Finanztechnologie werden in diesem Abschnitt definiert.

5.1.1 Gesundheit

Bei den Gesundheitsapplikationen ist es wichtig, dass die Smartwatch über einen Herzfrequenzmesser verfügt und die Kombination aus Gyroskop, Rotationssensor und Bewegungssensor eingebaut ist. Diese Elemente sind erforderlich um Pulsraten eines Menschen zu messen, sowie Analysen von Bewegungen zu erstellen.

Benötigte Sensoren:

- Gyroskop
- Bewegungssensor
- Rotationssensor
- Herzfrequenzmesser

5.1.2 Smart Home

Für die Steuerung von Haushaltsgeräte, Heizung oder Licht wird vor allem ein Steuerungsanzeige benötigt. Dies ermöglicht die Apparate ein- und auszuschalten, die Beleuchtung zu dimmen und Benachrichtigungen (z.B. von offenem Fenster, beendetem Waschgang uvm.) anzuzeigen. Für eine automatische Helligkeitsregulierung sollte ein Lichtmesser eingebaut sein. Um für Alarne auszugeben ist ein Lautsprecher oder Vibrationsmotor hilfreich.

Bedienelement:

- Touchscreen

Aktoren:

- Lautsprecher
- Vibrationsmotor

Benötigter Sensor:

- Lichtsensor

5.1.3 Finanztechnologie - FinTech

Für Geld Transaktionen zu führen wird eine drahtlose Verbindungsart vorausgesetzt. Transaktionen werden Bereits erwähnte Applikationen wie Apple Pay und Google Wallet verwenden den NFC (Near Field Communication) Chip und TWINT verwendet Bluetooth. Die Kommunikationsschnittstelle ist notwendig um Zahlungsanforderungen zu erhalten und diese auch zu autorisieren. Zum auswählen von Zahlungsoptionen und bestätigen von Überweisungen ist ein Bedienelement notwendig. Wenn mehrere Kreditkarten hinterlegt wären, muss der Anwender die Möglichkeit haben, um eine davon auszuwählen für die Überweisung.

Auswahl von benötigten Kommunikationsschnittstellen:

- NFC Chip
- Bluetooth
- GSM/UMTS/LTE
- Wireless LAN (weniger geeignet)

Mögliche Bedienelemente:

- Touchscreen (beste Voraussetzung)
- Mikrofon zur Sprachsteuerung
- Physische Taste

5.2 siot.net Applikationen Anforderungen

Technische Voraussetzung für die siot.net Applikationen werden für alle in der Bedürfnisanalyse ermittelten Klassen definiert.

5.2.1 siot.net Gateway Library

Die Bibliothek, welche smarte Geräte an die siot.net Plattform anbinden soll, wird in erster Linie nur für das Android Betriebssystem entwicklet. Für die Benutzung von Apps welche ans siot.net angeschlossen werden, müssen diese minimal die Android Tools der SDK Version 22 beherrschen (ab Android 5.0 Lollipop). Um die Daten an den MQTT Broker übermitteln zu können, braucht es ein Netzwerkmodul, die eine Verbindung ins Internet erlaubt. Die Bibliothek sollte alle verfügbaren und bekannten Sensoren selber erkennen. Die Verwendung, ob Sensoren aktiviert werden oder nicht, wird durch den Appentwickler oder der Software selber definiert.

Auswahl von benötigten Kommunikationsschnittstellen:

- NFC Chip
- Bluetooth
- GSM/UMTS/LTE
- Wireless LAN

Betriebssystem:

- ab Android 5.0
- ab SDK Tools Version 22 (Android Tools)

5.2.2 siot.net Sensorcenter

Die Sensorcenter App für Android Geräte soll ermöglichen, die dem Gerät verfügbaren Sensoren, dem siot.net zur Verfügung zu stellen. Die Applikation muss sich mit dem siot.net koppeln, Sensoren manifestieren und Sensordaten senden können. Diese App kann vom Bestehen der siot.net Gateway Library profitieren, welche integriert wird. Technisch benötigt diese Applikation, zusätzlich zu den Voraussetzungen der siot.net Gateway Library, ein Touchscreen.

Auswahl von benötigten Kommunikationsschnittstellen:

- NFC Chip
- Bluetooth

- GSM/UMTS/LTE
- Wireless LAN

Bevorzugtes Bedienelement:

- Touchscreen

Betriebssystem:

- ab Android 5.0
- ab SDK Tools Version 22 (Android Tools)

5.2.3 siot.net Dashboard App

Um Sensordaten von der siot.net Plattform darzustellen eignet sich eine Dashboard App. Diese wird in dieser Arbeit vorraussichtlich nur für Android Geräte in betracht gezogen. Für die Darstellung einer derartigen digitalen Instrumententafel eignet sich ein Display, bevorzugt ein Touchscreen. Ein Berührbildschirm erlaubt es die gewünschten Anzeigen bequem einzublenden. Eine weitere Voraussetzung ist die Vernetzung. Das Gerät muss einen Zugang zum Internet herstellen können. Nur durch eine erfolgreiche Anmeldung an den siot.net MQTT Broker, können die Informationen empfangen werden.

Auswahl von benötigten Kommunikationsschnittstellen:

- NFC Chip
- Bluetooth
- GSM/UMTS/LTE
- Wireless LAN

Bevorzugtes Bedienelement:

- Touchscreen

Betriebssystem:

- ab Android 5.0
- ab SDK Tools Version 22 (Android Tools)

5.2.4 Herzfrequenz Überwachung

Die Herzfrequenz Überwachung ist eine schlanke Variante des Dashboards. Um eine Anzeige wie bei einem Elektrokardiogramm (EKG) darzustellen braucht es ein Display. Und um Daten zu erhalten braucht es einen Herzfrequenzmesser. Um bei Überschreitung von definierten Werten einen Alarm auszulösen, braucht es womöglich noch Lautsprecher um akustische Signale auszusenden oder einen Vibrationsmotor für lautlose Anmeldungen.

Auswahl von benötigten Kommunikationsschnittstellen:

- Bluetooth
- GSM/UMTS/LTE
- Wireless LAN

Bevorzugtes Anzeigen / Aktoren:

- Touchscreen
- Lautsprecher

Sensoren:

- Herzfrequenzmesser

Betriebssystem:

- ab Android 5.0
- ab SDK Tools Version 22 (Android Tools)

5.2.5 Steuerung von Modellen

Für die Kontrolle über ein Modellfahrzeug oder Modellflugzeug zu erhalten, benötigt dieses einen Hardwarekontroller, welcher sich mit dem siot.net MQTT Broker verbinden kann. Die Steuerungseinheit muss auf die MQTT Topic subscriben und die empfangenen Bewegungsdaten interpretieren und an die erforderlichen Antriebsmotoren und Servos weitergeben. Als Kontroller könnte ein Raspberry Pi Zero oder ähnliches zum Tragen kommen. Auf den Hardwarekontroller wird nicht weiter eingegangen. Als Sender und Ermittler der Bewegungs- und Beschleunigungsdaten kommt ein Android Device (mit siot.net Sensorcenter) zum Einsatz.

Auswahl von benötigten Kommunikationsschnittstellen:

- Bluetooth
- GSM/UMTS/LTE
- Wireless LAN

Bevorzugtes Anzeigen / Aktoren:

- Touchscreen
- Lautsprecher

Sensoren:

- Bewegungssensor
- Beschleunigungssensor
- Gyroskop

Betriebssystem:

- ab Android 5.0
- ab SDK Tools Version 22 (Android Tools)

6 Technologiewahl

Im Kapitel der Technologiewahl werden einige aktuell erhältliche Smartwatch Modelle mit den theoretischen Daten verglichen, allgemeine Eigenschaften aufgeführt und die Entwicklerfreundlichkeit bewertet. Aufgrund der Bewertung von aufgestellten Kriterien werden ein bis zwei Uhren ausgewählt für die Arbeit.

6.1 Allgemeine Eigenschaften

Heute (Januar 2016) sind weitgehends alle erhältlichen tragbaren Computer am Handgelenk vom Smartphone abhängig, deshalb bieten sie nur einen kleinen Mehrwert. Sie können zwar viel mehr als nur die Zeit anzeigen, jedoch sehr beschränkt und zum grössten Teil wird das Mobiltelefon benötigt. Nachrichten, Termine und einkommende Telefonate anzeigen, aber ein Gespräch kann nur mit den wenigsten Geräten geführt werden, wie z.B. der Apple Watch. Mit den meisten Smartwatches können Nachrichten gelesen, jedoch nicht beantwortet werden, da diese über keine virtuelle Tastatur verfügen. Falls doch geschieht dies hauptsächlich über die Spracheingabe. Mit dem integrierten Touchscreenkeyboard, stellt die Samsung Gear S dar die einzige Ausnahme dar.

Smartwatches wollen die Gesundheit fördern, etliche davon messen den Puls, zählen die Schritte und erfassen zurückgelegte Wegstrecke. Weitere Apps erlauben den Schlaf zu überwachen, zur Bewegung aufzufordern bei langer Inaktivität und den Kalorienbedarf und Verbrauch berechnen. Durch die genauen Daten haben Amateursportler eine praktische Hilfe am Handgelenk, um ihren Körper fit zu halten. Für ambitionierte Sportler ist der Funktionsumfang no zu gering.

Manche Uhren der Hersteller Apple, LG, Samsung und Alcatel messen den Puls fast Elektrokardiogramm-genau. Apple arbeitet bereits an einem Elektrokardiogramm (EKG) Armband, welches Daten per Ultraschall übermitteln soll. Das Band ermittelt die Herzströme über zwei angebrachte Elektroden. Dies ermöglicht der Apple Smartwatch als mobiles EKG bereitzustehen¹.

Ein grosse, fast alle betreffende, Schwachstelle von Smartwatches sind ihre leistungsschwachen Akkumulatoren. Durchschnittlich erreichen die Uhren eine Laufzeit von knapp 24 Stunden bei geringer Nutzung.

Applikationen welche einen grossen Mehrwert bringe gibt es no zu wenige. Zum heutigen Zeitpunkt stellt sich heraus, dass die Minicomputeruhren überwiegend als Erweiterungsbildschirm des Smartphones dient. Wenige Geräte werkeln autonom, wie zum Beispiel die Samsung, jedoch bleiben die meisten ein verlängerter Arm des dazugehörigen Mobiltelefons. Dies liegt daran, weil die smarten Uhren noch Heute in den Kinderschuhen stecken.

¹vgl. <http://www.smartwatch.de/news/neues-apple-watch-armband-mit-ultraschall-ekg-geplant>, 20.11.2015

6.2 Kandidaten

Smartwatch	Apple Watch ^a	LG G Watch ^b	LG Watch Urbane ^c
Betriebssystem	WatchOS	Android Wear	Android Wear
Funktionen	gut	befriedigend	befriedigend
- Nachrichten	empfangen möglich, senden per Sprachnachricht, Emojis senden	empfangen möglich, senden per Sprachnachricht	empfangen möglich, senden per Sprachnachricht
- Telefonieren	führen von Gesprächen	nur Notifikation	nur Notifikation
- Display	42mm 312x390px 38mm 272x340px	1.65 Zoll 280x280px	1.3 Zoll 320x320px
Ø Akkulaufzeit	19h	30h	20h
Datensendungsverhalten	verschlüsselt	verschlüsselt	verschlüsselt
Smartphone-Betriebssystem	ab iOS 8.2	ab Android 4.3 ab iOS 8.2	ab Android 4.3 ab iOS 8.2
Pulsmesser	optischer Pulsmesser	kein Pulsmesser	optischer Pulsmesser
Gesamteindruck	bestes Display viele Funktionen befriedigend - gut	Always-On Display befriedigender Akku befriedigend	rundes Display edle Verarbeitung befriedigend

Tabelle 6.1: Technische Daten von Smartwatches - Teil 1

^avgl. Stiftung Warentest Magazin, Ausgabe: 10/2015

^bvgl. Modul BTI7302 Evaluation Wearables, 27.06.2015

^cvgl. Stiftung Warentest Magazin, Ausgabe: 10/2015

Smartwatch	Motorola Moto 360 ^a	Samsung Gear S2 ^b	Alcatel Onetouch Watch ^c
Betriebssystem	Android Wear	Tizen	proprietäres Alcatel OS
Funktionen	befriedigend	gut	befriedigend
- Nachrichten	empfangen möglich, senden per Sprachnachricht	empfangen möglich, senden möglich	empfängt nur Kurznachrichten
- Telefonieren	nur Notifikation	nur Notifikation bei eSIM Variante möglich	nur Notifikation
- Display	1.57 Zoll 360x290px	1.2 Zoll 360x360px	1.3 Zoll 320x320px
∅ Akkulaufzeit	10h	38h	20h
Datensendungsverhalten	verschlüsselt	verschlüsselt	unverschlüsselt
Smartphone-Betriebssystem	ab iOS 8.2	ab Android 4.4	ab Android 4.3 ab iOS 7
Pulsmesser	optischer Pulsmesser	optischer Pulsmesser	optischer Pulsmesser
Gesamteindruck	bestes Display meiste Funktionen befriedigend	Intuitive Lünette gute Akkuleistung befriedigend - gut	schwaches Display mangelhafte Datenübertragung mangelhaft

Tabelle 6.2: Technische Daten von Smartwatches - Teil 2

^avgl. Modul BTI7302 Evaluation Wearables, Michael Fankhauser und Sathesh Paramasamy, Ausgabe: 27.06.2015

^bvgl. <https://www.androidpit.de/samsung-gear-s2-test>, Stand: 27.12.2015

^cvgl. Stiftung Warentest Magazin, Ausgabe: 10/2015

6.3 Entscheid Smartwatch

Wenn der Funktionsumfang betrachtet wird, wäre eine Apple Watch oder die Samsung Gear S2 die bevorzugte Wahl gewesen. Die Samsung ist erst seit Oktober 2015 erhältlich, folglich fiel diese Aufgrund des späten Erscheinungsdatum und dessen Samsung proprietären Betriebssystem Tizen aus dem Katalog. Die Apple Watch wurde nicht gewählt, weil die Entwicklung sich auf die Apple spezifische Programmiersprache Swift beschränkt. Ein weitere Entscheidungspunkt ist, dass der Markt aus einem Produkt besteht, wenn man die Modelle Watch und Watch Sport und die Größen, 38mm und 42mm, nicht unterscheidet, da diese sich nur optisch voneinander unterscheiden. Ausgewählt wurden zwei Produkte, welche nicht sehr aufgefallen sind: Die **LG G Watch** und die **Motorola Moto 360**. Als erste auf dem Markt erhältliche Android Wear Smartwatch, wurde die LG G Watch ausgewählt. Der eingebaute Snapdragon 400 Prozessor rechnet mit 1,2 GHz, hat 512MB RAM und hat ein rechteckigen Bildschirm, welcher die Funktion biete immer eingeschaltet zu sein. Wichtiger Pluspunkt ist, die LG Datenuhr kann über USB gedebugged werden. Dies erleichtert die Entwicklung von Applikation. Um ein Android Wear Vergleichsobjekt zu erhalten, kam die neuere Motorola Moto 360 zusätzlich in die Entscheidung. Diese arbeitet mit einem Texas Instrument OMAP 3 Prozessor mit 1 GHz, hat ebenfalls 512MB RAM und hat einen runden Touchscreen. Mehrwert der Moto 360 zur G Watch sind, ein eingebautes WLAN Modul, Lichtsensor, Pulssensor und eine physische Taste. Bedauerlicherweise kann sie nur über Bluetooth gedebuggt werden, was viel Zeit beansprucht um entwickelte Apps zu übertragen. Durch die Entscheidung für Android Wear liegt der Schwerpunkt auf der Entwicklungssprache JAVA.

7 Architektur

7.1 Android Wear Paket Architektur

Android Wear hat eine strikte Paket Architektur. Abbildung 7.1 veranschaulicht die Übersicht dieser Architektur. Eine Android Wear App muss immer zwingend eine dazugehörige Android App beinhalten. Wenn beide Apps gemeinsam verwendete Klassen haben, müssen diese in ein solches Paket angelagert werden. Android Klassen können nicht in einer Android Wear Applikation instanziiert werden. Üblicherweise wird bei der Installation der .apk auf ein Android Smartphone, parallel die Smartwatch App auf die Uhr mit installiert.

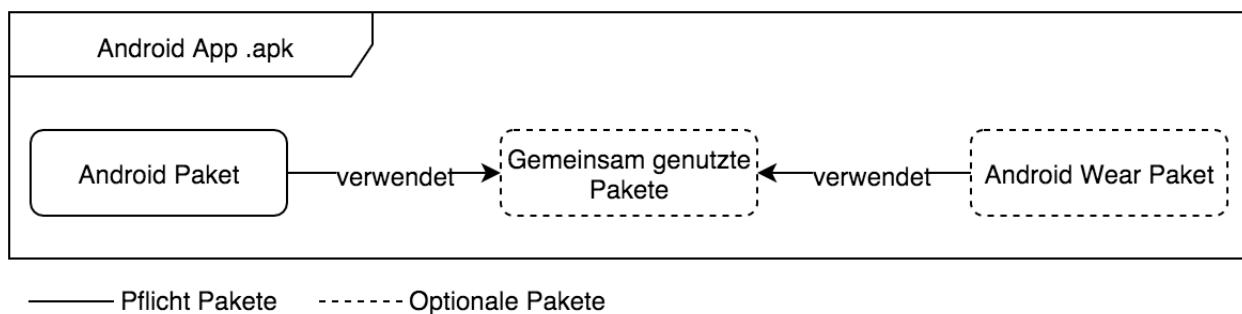


Abbildung 7.1: Übersicht über die einzuhaltende Paketstruktur

7.2 siot.net Applikationsarchitektur

Die Architektur auf der Abbildung 7.2 definierte und ersichtliche muss eingehalten. Android und Android Wear Applikation für siot.net können so am effizientesten konzipiert werden.

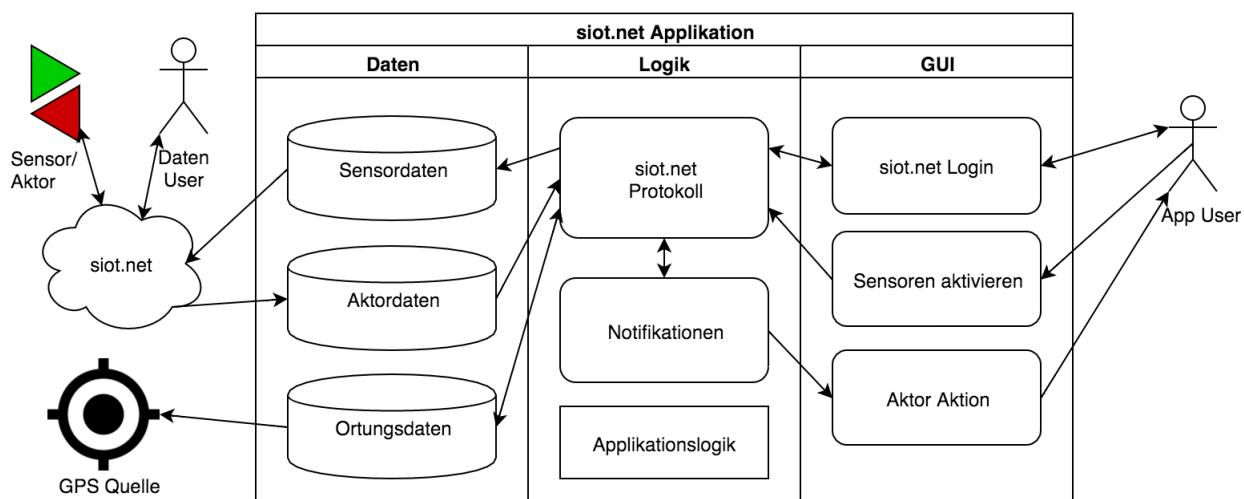


Abbildung 7.2: Applikationsarchitektur für siot.net Anwendungen

7.3 Netzwerk-Architektur

7.3.1 geplante Architektur

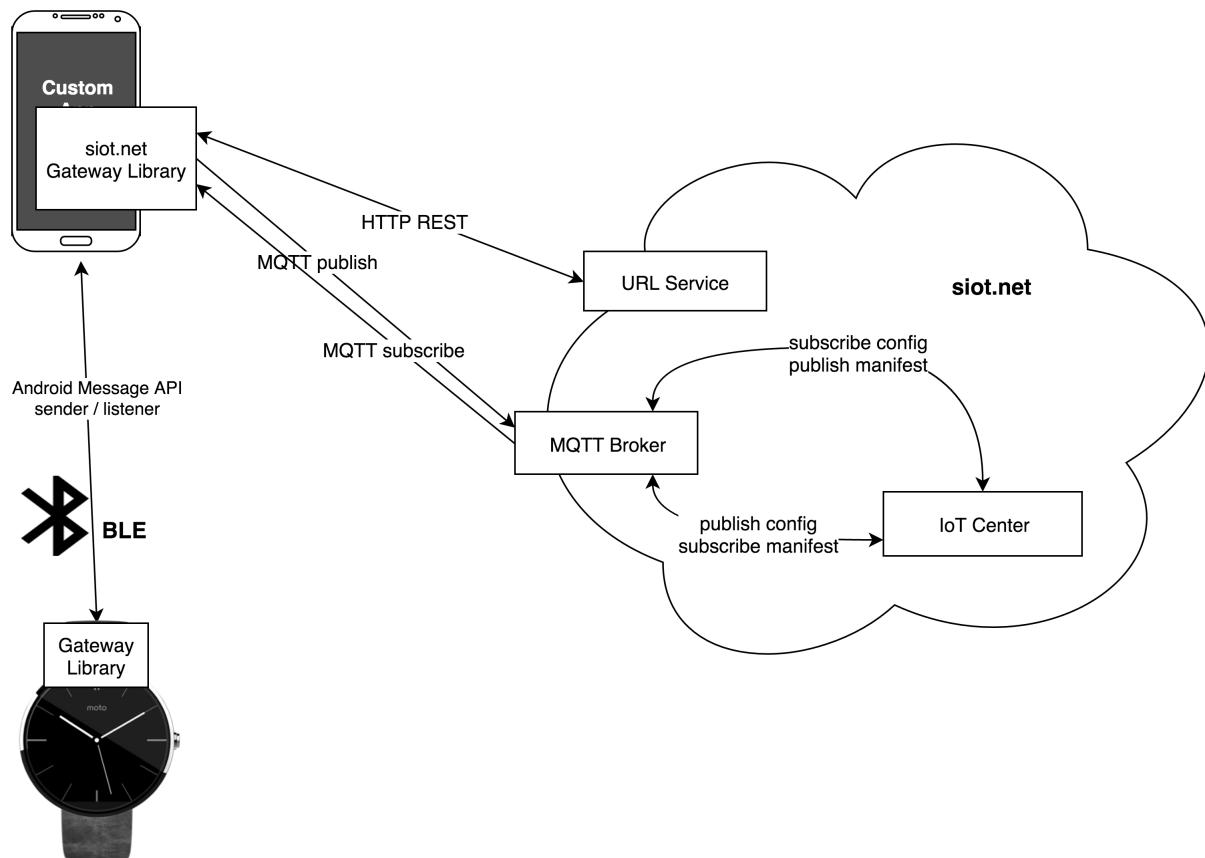


Abbildung 7.3: Geplante Kommunikation zwischen Smartwatch und Smartphone, sowie Smartphone und siot.net

In der Abbildung 7.3 visualisiert, wie die Kommunikation zwischen der Smartwatch, dem Smartphone und der siot.net Plattform stattfinden soll. Im Falle, dass die Smartwatch eine Bluetooth Low Energy (BLE) Verbindung mit dem Smartphone aufweist, wird dieses via REST und MQTT, mit siot.net Daten austauschen. Dabei dient das Mobiltelefon als Datenbrücke von der Smartwatch zur IoT-Cloud. Die Android Wear Uhr versendet die Daten, per Android Message API via BLE, zum Android Smartphone, das leitet die Pakete mit Hilfe des MQTT Protokoll weiter ans siot.net. Diese Variante der Datenübermittlung ist sehr viel stromsparender als die in Abbildung 7.4 gezeigte.

Das Bridging muss in der Smartphone Begleitapp (Android Paket) der Smartwatch Applikation (Android Wear Paket) implementiert sein.

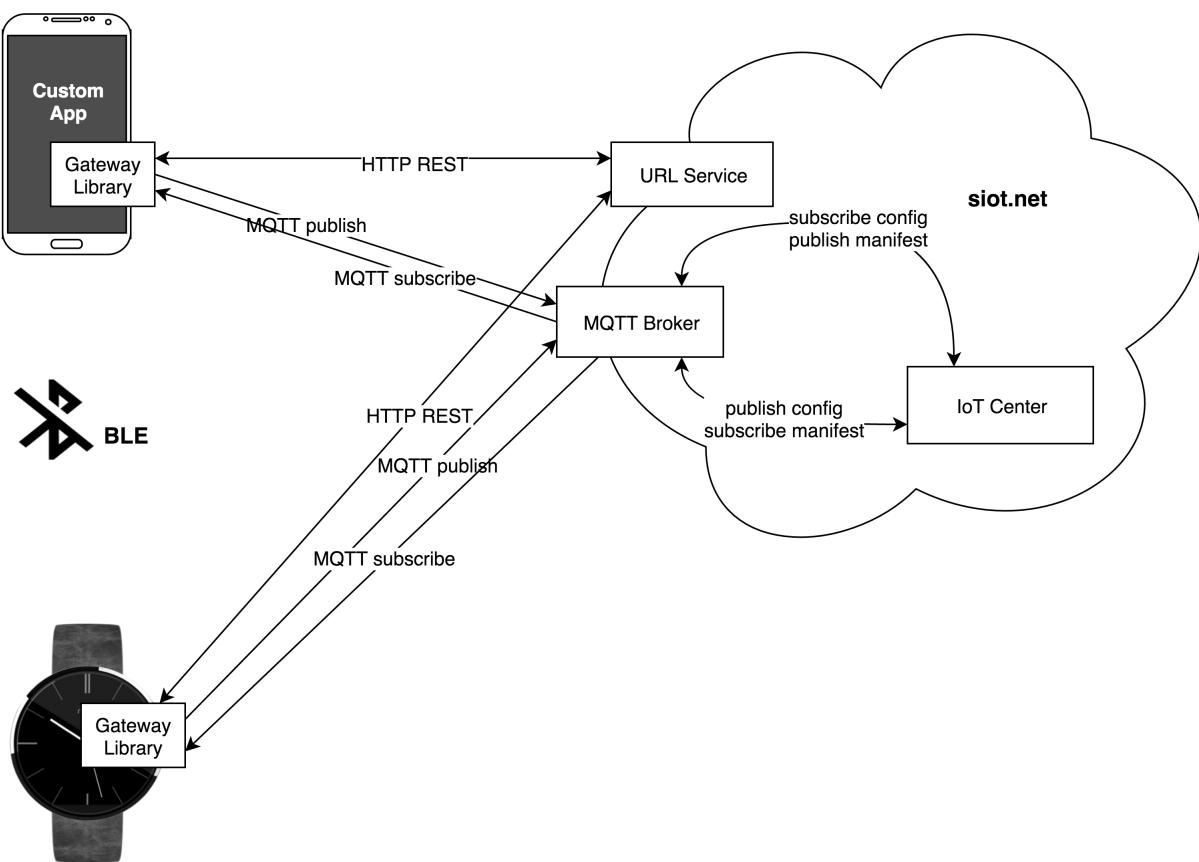


Abbildung 7.4: Geplante Kommunikation zwischen Smartwatch und siot.net, sowie Smartphone und siot.net

Bei Abbruch der Verbindung von Smartwatch zu Smartphone, soll die Computeruhr die Kommunikation zum siot.net selber übernehmen. Dazu muss zuerst über REST die URL des MQTT Broker ermittelt werden, um folgliche eine Kopplung durchzuführen. Diese Architektur ist genauer auf Abbildung 7.4 zu betrachten.

7.3.2 effektive Architektur

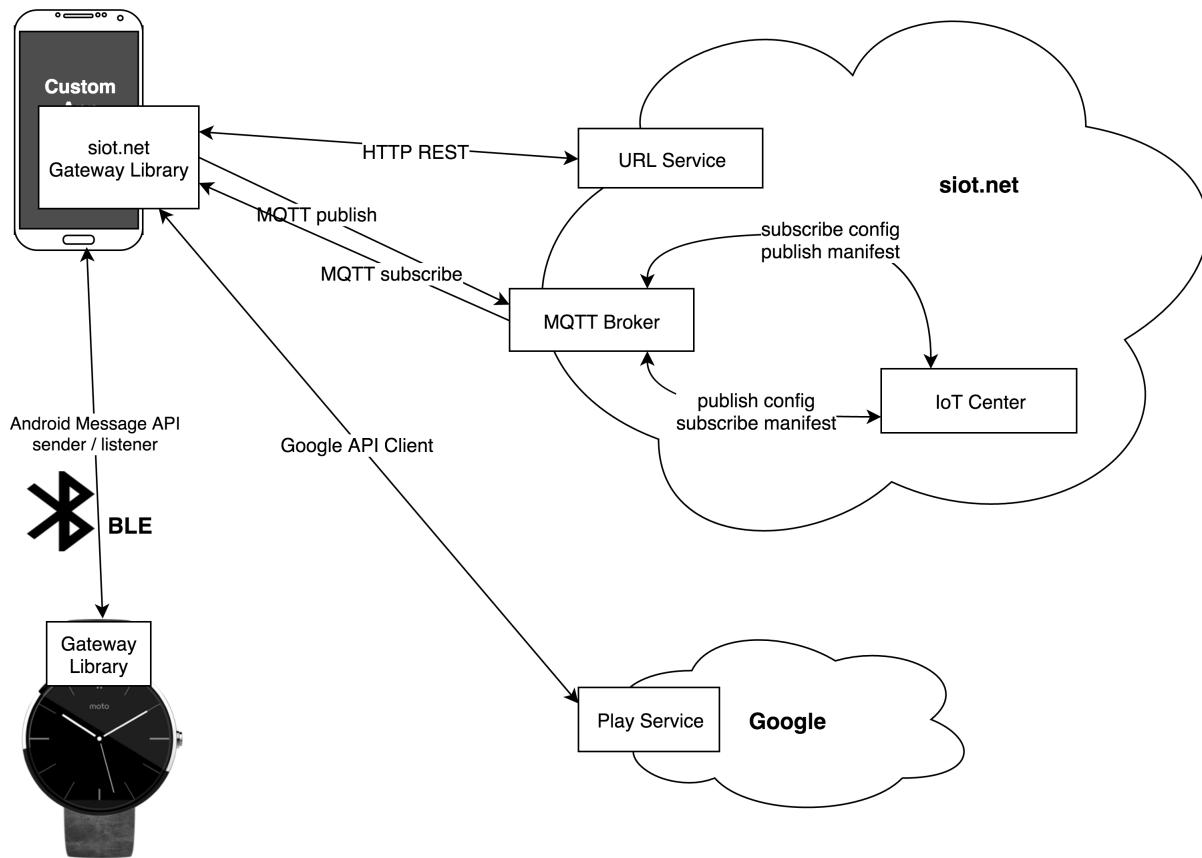


Abbildung 7.5: Effektive Kommunikation zwischen Smartwatch und siot.net, sowie Smartphone und siot.net

Diese Architektur (siehe Abbildung 7.5) unterscheidet sich geringfügig zu jener Abbildung 7.3. Hier kommt nun die Google Cloud dazu. Dies ist notwendig, da Android nicht erlaubt direkt über die Android Message API zu kommunizieren. Es muss vorhergehend eine Nodeld über den Google Play Service eingefordert werden. Dank dieser Nodeld dürfen die Geräte (Smartphone und Smartwatch benötigen eine eindeutige Identifikation) mittels Message API Nachrichten transferieren. Der Datenaustausch geschieht dann direkt via BLE. Dies ist die Netzwerk-Architektur kommt zum tragen, wenn die Kopplung von Smartwatch und Smartphone über Bluetooth Smart steht.

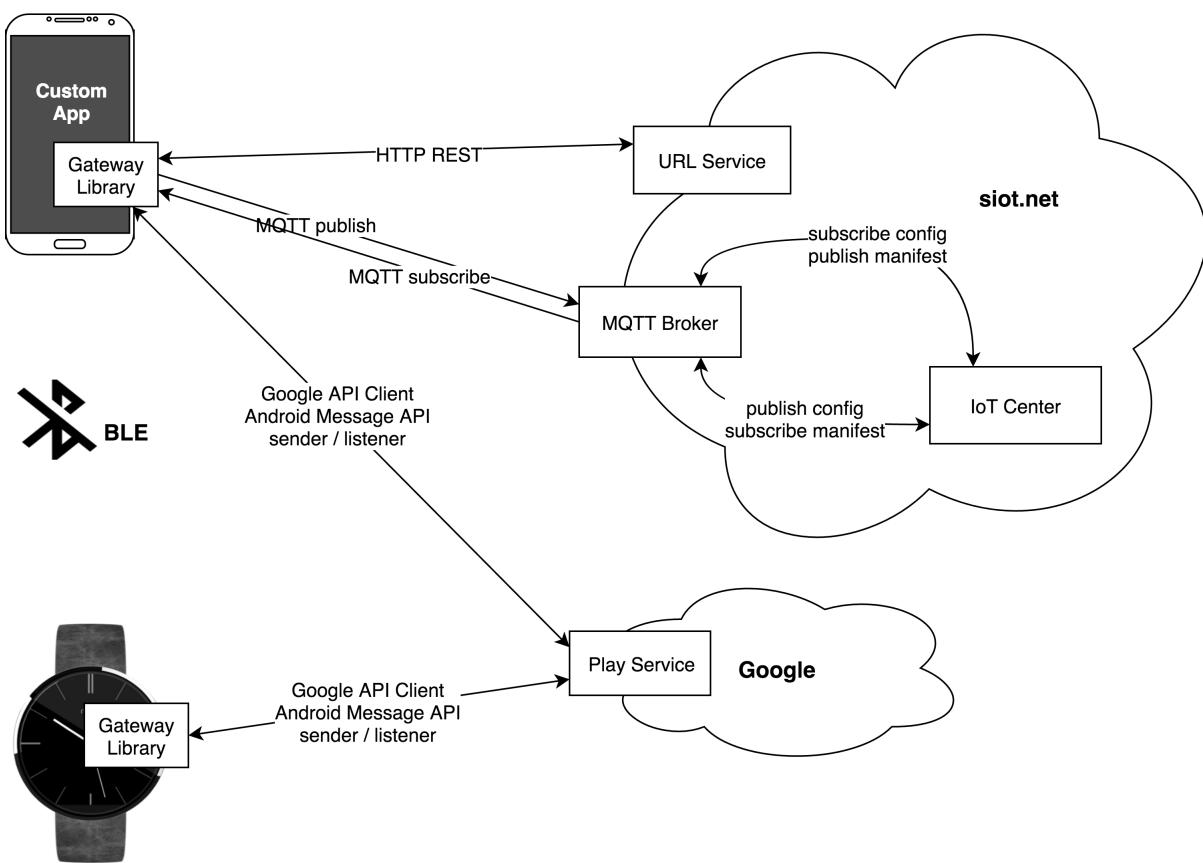


Abbildung 7.6: Effektive Kommunikation zwischen Smartwatch und siot.net, sowie Smartphone und siot.net

Abbildung 7.6 zeigt die funktionierende Architektur. Diese differenziert sich wesentlich von der geplanten Netzwerk-Definition. Auch hier spielt der Faktor Android eine grosse Rolle, denn es verbietet Android Wear Geräten den direkten Kontakt zum Internet. So ist hier eine Verbindung über den Google Play Service zum Smartphone nötig. Bei dieser Anbindungsart werden die Daten via Google Cloud an das Android Handy gesendet. Es muss in jedem Fall die Verbindung zu siot.net verwalten.

Der grosse Nachteil dieser Variante ist, dass die Smartwatch nicht als autonomes Gerät fungieren kann. Wenn das Smartphone nicht erreichbar ist, z.B. Akku leer, keine Netzwerkverbindung, ist die Computeruhr für siot.net nutzlos.

8 Anforderungsdokumente

Dieser Teil des Dokuments beinhaltet drei Anforderungsdokumente von der vorhergehend ermittelten Anwendungen. Requirements von der siot.net Gateway Library, siot.net Sensorcenter und siot.net Dashboard App sind aufgeführt.

8.1 Requirements - siot.net Gateway Library

In diesem Requirementsdokument werden die Anforderung an die siot.net Gateway Library (**sGWLib**) spezifiziert.

8.1.1 Allgemeine Beschreibung

Die siot.net Gateway Library soll Entwicklern von Android Apps für Mobile Geräte, wie Smartphones oder Smartwatches, erleichtern Ihre Software an die siot.net Plattform anzubinden. Programmieren von Applikationen muss immer einfacher und schneller werden, da Projekte immer mehr an höherem Zeitdruck leiden. Diese Bibliothek soll helfen Integrationen von Apps an das siot.net ohne grosses Vorwissen, schnell durchzuführen. Die Library übernimmt die Verwaltung von den Sensor den Gerätes, sowie die Kommunikation von Mobile Device zu siot.net und Smartwatch zu Smartphone.

8.1.2 User Requirements

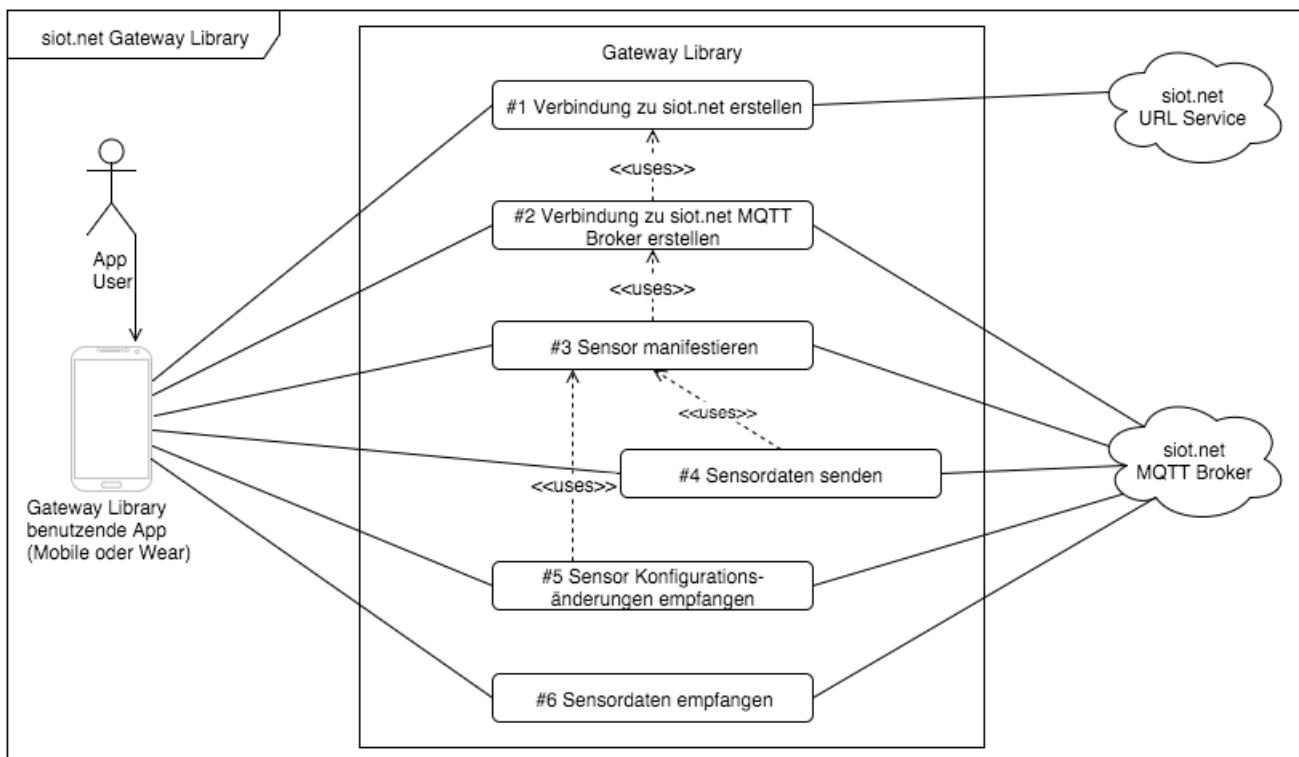


Abbildung 8.1: Das allgemeine Use Case Diagramm für die siot.net Gateway Library

8.1.3 Funktionale Anforderungen und Anwendungsfallbeschreibungen

8.1.3.1 Anwendungsfall #1

Nr. und Name:	#1 Verbindung zu siot.net erstellen.
Szenario:	Android Gerät wird mit dem siot.net verbunden.
Kurzbeschreibung:	Die Verbindung des Android Devices zum siot.net wird hergestellt, dies geschieht über den URL Service um die nötigen URLs zu erhalten (IoT Center und MQTT Broker).
Beteiligt Akteure:	Android App (User), siot.net Gateway Library, siot.net URL Service, siot.net MQTT Broker.
Auslöser und Vorbedingung:	User gibt über seine App den Befehl zu die Applikation zu einer IoT Cloud zu verbinden. Wenn siot.net als Ziel gewählt wird, muss die siot.net Gateway Library eingebunden sein. Für eine erfolgreiche Verbindung muss eine gültige siot.net Lizenz vorhanden sein.
Ergebnisse und Nachbedingung:	Der URL Service von siot.net kann die Lizenz richtig verifizieren und liefert die benötigten URLs.

Tabelle 8.1: siot.net Gateway Library: Übersicht Anwendungsfall #1

Ablauf:

1-1	App User	Startet die App.
1-2	App User	Gibt siot.net Lizenz ein und klickt verbinden zu siot.net.
1-3	App	Forder bei der siot.net Gateway Library (sGWLib) die Verbindung an.
1-4	sGWLib	Verfiziert sich beim siot.net URL Service mit dem Lizenzschlüssel.
1-5	sGWLib	Erhält URL vom siot.net MQTT Broker und verbindet sich (Anwendungsfall #2).
1-6	sGWLib	App über Verbindungsstatus benachrichtigen.
1-7	App	Benachrichtigung verarbeiten, z.B. Meldung an App User.

Tabelle 8.2: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #1

Ausnahmen und Varianten:

1-4.1	sGWLib	Ausnahme: siot.net URL Service nicht erreichbar, App benachrichtigen.
1-4.2	sGWLib	Ausnahme: Lizenzschlüssel nicht valide, App benachrichtigen.
1-5.1	sGWLib	Ausnahme: MQTT Broker nicht erreichbar, App benachrichtigen.

Tabelle 8.3: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #1

8.1.3.2 Anwendungsfall #2

Nr. und Name:	#2 Verbindung zu siot.net MQTT Broker erstellen.
Szenario:	Verbindung zum siot.net MQTT Broker herstellen.
Kurzbeschreibung:	Durch die vom URL Service erhaltene MQTT URL wird das Android Gerät wird mit dem Broker gekoppelt.
Beteiligt Akteure:	siot.net Gateway Library, siot.net URL Service, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Durch das erhalten einer gültigen MQTT Broker URL wird der Verbindungsversuch gestartet.
Ergebnisse und Nachbedingung:	Es kann erfolgreich ein MQTT Client Verbindung produziert werden.

Tabelle 8.4: siot.net Gateway Library: Übersicht Anwendungsfall #2

Ablauf:

2-1	sGWLib	Empfang der URLs vom siot.net URL Service.
2-2	sGWLib	MQTT URL wird verwendet um eine Verbindung zum Broker aufzubauen.
2-3	sGWLib	Verbindungsstatus wird überprüft und stellt die Verbindungsinstanz der App zur Verfügung.

Tabelle 8.5: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #2

Ausnahmen und Varianten:

2-2.1	sGWLib	Ausnahme: Authentifizierung fehlgeschlagen - Lizenzschlüssel ungültig, App benachrichtigen.
2-2.2	sGWLib	Ausnahme: Keine MQTT Broker URL erhalten, Information an App senden.
2-2.3	sGWLib	Ausnahme: MQTT Broker nicht erreichbar, App benachrichtigen.
2-2.4	sGWLib	Variante: Mehrere MQTT Broker URLs erhalten, absteigend der Reihe nacheinander Verbindungsversuche starten bei erstem Erfolg diesen Client verwenden und der App zur Verfügung stellen.

Tabelle 8.6: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #2

8.1.3.3 Anwendungsfall #3

Nr. und Name:	#3 Sensor manifestieren.
Szenario:	Manifestieren vom gewünschten Sensor beim siot.net
Kurzbeschreibung:	Jeder Sensor welcher an die siot.net Plattform angebunden wird, muss sich zuerst manifestieren. Dies geschieht durch eine MQTT Meldung, des Datentyps Manifest (definiert durch siot.net), an den siot.net Broker.
Beteiligt Akteure:	siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Wenn die App zum ersten Mal eine App ans siot.net anmelden will und dieser nicht dem IoT Center nicht bekannt ist.
Ergebnisse und Nachbedingung:	Der Sensor kann erfolgreich am IoT Center von siot.net manifestiert und danach verwaltet werden.

Tabelle 8.7: siot.net Gateway Library: Übersicht Anwendungsfall #3

Ablauf:

3-1	App	App meldet einen Sensor zur Anmeldung am siot.net dem siot.net Gateway Library.
3-2	sGWLib	Sensor wird instanziert, eindeutige GUID wird generiert und Manifest wird erstellt.
3-3	sGWLib	Manifest wird an siot.net MQTT Broker gesendet.
3-4	sGWLib	Gateway Library abonniert die Konfigurationstopic des MQTT Brokers des manifestierten Sensors.
3-5	sGWLib	Sensor ist nun für die App bereitgestellt.

Tabelle 8.8: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #3

Ausnahmen und Varianten:

3-3.1	sGWLib	Ausnahme: MQTT Verbindung nicht verfügbar, App informieren.
3-2.1	sGWLib	Variante: Gemeldete Sensor verwaltet mehrere physische Werte, pro Angabe wird ein Sensorsmanifest angelegt.
3-2.2	sGWLib	Variante: Sensor ist dem siot.net IoT Center bereits bekannt, es wird kein neuer Sensor angelegt sondern als bereits verfügbaren registriert, keine Aktion notwendig.

Tabelle 8.9: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #3

8.1.3.4 Anwendungsfall #4

Nr. und Name:	#4 Sensordaten senden.
Szenario:	Sensorwerte an den siot.net MQTT Broker senden.
Kurzbeschreibung:	Sensordaten werden vom Datentyp Sensordaten (spezifiziert von siot.net) an den Broker per MQTT gesendet.
Beteiligt Akteure:	Android System, siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Sensor meldet, dass neue Daten verfügbar sind.
Ergebnisse und Nachbedingung:	Werte können mit Erfolg an den MQTT broker publiziert werden.

Tabelle 8.10: siot.net Gateway Library: Übersicht Anwendungsfall #4

Ablauf:

4-1	Android	Sensor meldet an, dass neue Werte ermittelt wurden.
4-2	sGWLib	Sensordaten Datentyp wird angefertigt.
4-3	sGWLib	Sensordaten Meldung wird an siot.net MQTT Broker gesendet.

Tabelle 8.11: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #4

Ausnahmen und Varianten:

4-3.1	sGWLib	Ausnahme: MQTT Verbindung nicht verfügbar, App informieren.
--------------	--------	---

Tabelle 8.12: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #4

8.1.3.5 Anwendungsfall #5

Nr. und Name:	#5 Sensor Konfigurationsänderung empfangen.
Szenario:	Konfigurationsänderungen im siot.net IoT Center empfangen und verarbeiten.
Kurzbeschreibung:	Sensoren werden im siot.net IoT Center verwaltet und konfiguriert. Bei der Änderung einer Einstellung wird der Sensor per MQTT Message notifiziert. Die Gateway Library empfängt die Nachricht und stellt den Sensor den Werten entsprechend ein.
Beteiligt Akteure:	siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Konfigurationsnachricht von siot.net MQTT Broker erhalten.
Ergebnisse und Nachbedingung:	Sensor kann korrekt umkonfiguriert werden.

Tabelle 8.13: siot.net Gateway Library: Übersicht Anwendungsfall #5

Ablauf:

5-1	sGWLib	Erhält Message mit neuen Sensoreinstellungen.
5-2	sGWLib	Konfiguration wird geparsed und einem Sensor zugeordnet.
5-3	sGWLib	Sensor wird umkonfiguriert.

Tabelle 8.14: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #5

Ausnahmen und Varianten:

5-3.1	sGWLib	Variante: Konfiguration ist einem physischen Werte eines Sensors zugeordnet, welcher mehrere verwaltet, gesammten Sensor neu einstellen. Weiteres Verhalten muss in einem neuen Projekt genauer definiert werden (Sensorkombinationen sind im siot.net vorgesehen, jedoch noch nicht spezifiziert und freigegeben).
--------------	--------	---

Tabelle 8.15: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #5

8.1.3.6 Anwendungsfall #6

Nr. und Name:	#6 Sensordaten empfangen.
Szenario:	Sensordaten von anderen Sensoren im siot.net empfangen und bereitstellen.
Kurzbeschreibung:	Die siot.net Gateway Library kann auch Sensordaten empfangen welche beim siot.net IoT Center angemeldet sind. Sensordaten können beim MQTT Broker abonniert werden. Beim Empfang einer Informationennachricht wird diese interpretiert und der App zur Verwendung gestellt.
Beteiligt Akteure:	siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Sensordaten Message von siot.net MQTT Broker erhalten.
Ergebnisse und Nachbedingung:	Informationen sind interpretiert und verfügbar gemacht.

Tabelle 8.16: siot.net Gateway Library: Übersicht Anwendungsfall #6

Ablauf:

6-1	sGWLib	Erhält Message mit Sensordaten.
6-2	sGWLib	Informationen interpretieren.
6-3	sGWLib	Datenzugriff der App erlauben.

Tabelle 8.17: siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #6

Ausnahmen und Varianten:

6-2.1	sGWLib	Ausnahme: Empfangene Nachricht kann nicht interpretiert werden als Sensordaten, Information wird verworfen.
--------------	--------	---

Tabelle 8.18: siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #6

8.1.4 Nicht-funktionale Anforderungen

Produktanforderungen:

Benutzbarkeitsanforderungen	Die Bibliothek muss eine Java Library sein, um diese möglichst einfach in ein Android Projekt integrieren zu können.
Effizienzanforderungen	Ein übersichtliche und schlanke Dokumentation (z.B. JavaDoc oder Entwicklerhandbuch) soll dem Entwickler helfen, eine neue App zu programmieren.
Zuverlässigkeitssanforderungen	Die Daten, welche transferiert werden, verwenden das fire-and-forget Entwicklungsdesignmuster. Bei diesem Muster werden die gesendeten Nachrichten nicht mehr verfolgt. Wenn eine Message nicht ankommt, wird dies von keinem System bemerkt. Da Daten nur durch den vom Benutzer gewünschten Sensoren versendet werden, vereinfacht dieses Pattern die Kommunikation zwischen Smartdevice und siot.net. Der Sicherheitsaspekt wird in einem weiteren Release verfolgt. Technisch sollte die Applikation, die vorhandene Hardware nicht überfordern.
Portierbarkeitsanforderungen	Die Informationen werden in einem Format übertragen und empfangen wie sie von siot.net spezifiziert sind.
Performanceanforderung	An die Performance sind keine spezifischen Anforderungen gestellt. Daten sollten in nützlicher Frist gesendet werden (z.B. <5 Sekunden Verzögerung pro Nachricht)

Tabelle 8.19: siot.net Gateway Library: Nicht-funktionale Produktanforderungen

8.2 Requirements - siot.net Sensorcenter

In diesem Requirementsdokument werden die Anforderung an die Android App siot.net Sensorcenter Android (**sSC-A**) und siot.net Sensorcenter Android Wear (**sSC-AW**) spezifiziert.

8.2.1 Allgemeine Beschreibung

Das siot.net Sensorcenter für Android Geräte, ist die erste Applikation welche die siot.net Gateway Library (sGWLib) integrieren soll. Ein Android Device eignet sich sehr gut als eigenes Sensorcenter, weil die mobilen Telefone und Uhren haben grosse Mengen von Sensoren eingebaut. Diese App bietet die Möglichkeit alle vom User gewünschten Sensordaten dem siot.net preiszugeben. Der Benutzer kann die Sensoren selber aktivieren und deaktivieren. Die gemessenen Daten werden via MQTT, mit Hilfe der siot.net Gateway Library, übermittelt.

8.2.2 User Requirements

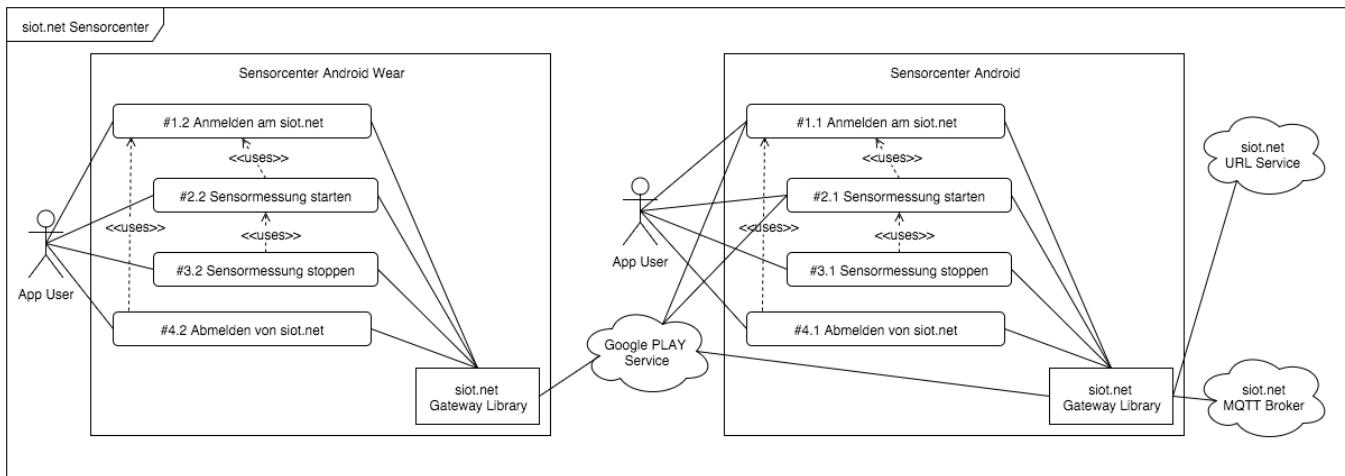


Abbildung 8.2: Das allgemeine Use Case Diagramm für das siot.net Sensorcenter

8.2.3 Funktionale Anforderungen und Anwendungsfallbeschreibungen

8.2.3.1 Anwendungsfall #1.1 und #1.2

Nr. und Name:	#1.1 und #1.2 Anmelden am siot.net.
Szenario:	Android Mobil oder Wear Gerät wird am siot.net angemeldet.
Kurzbeschreibung:	Die Verbindung zum siot.net wird hergestellt. Bei beiden Use Cases geschieht dies über die siot.net Gateway Library. Jedoch wird beim Anwendungsfall 1.2 die Verbindung über das gekoppelte Android Mobile Device hergestellt.
Beteiligt Akteure:	App User, siot.net Gateway Library, siot.net URL Service, siot.net MQTT Broker, Google Play Service (bei #1.2).
Auslöser und Vorbedingung:	User gibt siot.net Lizenzschlüssel ein und aktiviert den Verbinden Button.
Ergebnisse und Nachbedingung:	Mit den eingegebenen Daten kann die siot.net Gateway Library erfolgreich eine Verbindung aufbauen.

Tabelle 8.20: siot.net Sensorcenter: Übersicht Anwendungsfall #1.1 und #1.2

Ablauf #1.1:

1.1-1	App User	Sensorcenter App auf dem Smartphone starten.
1.1-2	App User	Lizenzschlüssel von siot.net eingeben und „Verbinden“ klicken.
1.1-3	sSC-A	Forder bei der siot.net Gateway Library (sGWLib) die Verbindung an.
1.1-4	sSC-A	Erhält die Antwort des Verbindungsversuches.
1.1-5	sSC-A	Zeigt die Informationen zur Verbindung an. Alle verfügbaren Sensoren werden mit Ein-/Ausschalter eingeblendet.

Tabelle 8.21: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #1.1

Ablauf #1.2:

1.2-1	App User	Sensorcenter App auf der Smartwatch starten.
1.2-2	sSC-AW	App auf dem Smartphone starten lassen.
1.2-3	sSC-A	siot.net Sensorcenter Android App auf dem Android Mobil Gerät starten.
1.2-4	App User	„Verbinden mit siot.net“ auf der Uhr klicken.
1.2-5	sSC-AW	Verbindung zu siot.net wird von sSCA verlangt.
1.2-6	sSC-A	sSCA fordert, unter Verwendung des bereits eingegebenen Lizenzschlüssels, die Kopplung mit siot.net von der Gateway Library.
1.2-7	sSC-A	Verbindet Smartphone zu siot.net und gibt die Verbindungsdaten an sSCAW weiter.
1.1-5	sSC-AW	Visualisiert Aktiverungs- und Deaktivierungsschalter aller verfügbaren Sensoren.

Tabelle 8.22: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #1.2

Ausnahmen und Varianten #1.1 und 1.2:

1.2-2.1	sSC-AW	Ausnahme: Smartphone nicht erreichbar, Sensorcenter kann auf der Smartwatch nicht verwendet werden.
1.2-2.2	sSC-AW	Ausnahme: Google Play Service nicht verfügbar, Sensorcenter kann auf der Smartwatch nicht verwendet werden.
1.1-3.1 1.2-6.1	sSC-A	Ausnahme: siot.net Gateway Library antwortet dass, siot.net URL Service nicht erreichbar, Verbindung kann nicht aufgebaut werden.
1.1-3.2 1.2-6.2	sSC-A	Ausnahme: siot.net Gateway Library antwortet dass, siot.net Lizenz nicht gültig, User benachrichtigen.
1.1-3.3 1.2-6.3	sSC-A	Ausnahme: siot.net Gateway Library antwortet dass, siot.net MQTT Broker nicht verfügbar, Verbindung kann nicht aufgebaut werden.

Tabelle 8.23: siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #1.1 und #1.2

8.2.3.2 Anwendungsfall #2.1 und #2.2

Nr. und Name:	#2.1 und #2.2 Sensormessungen starten.
Szenario:	Sensor wird auf Android oder Android Wear Gerät gestartet. Messungen werden an siot.net übermittelt.
Kurzbeschreibung:	Mit den eingebblendeten Ein- und Ausschalter kann jeder verfügbare Sensor einzeln aktiviert werden. Die gemessenen Daten werden bei #2.1 mittels siot.net Gateway Library via MQTT an den siot.net Broker weitergeleitet. Beim Anwendungsfall #2.2 werden diese über den Google Play Service ans Smartphone geleitet, welches dann die Informationen an den MQTT Broker sendet.
Beteiligt Akteure:	App User, siot.net Gateway Library.
Auslöser und Vorbedingung:	Ein oder mehrere Sensoren werden mit der App eingeschaltet.
Ergebnisse und Nachbedingung:	Sensor kann erfolgreich aktiviert werden.

Tabelle 8.24: siot.net Sensorcenter: Übersicht Anwendungsfall #2.1 und #2.2

Ablauf #2.1 und #2.2:

2-1	App User	Einen gewählten Sensor via Ein-/Ausschalter aktivieren.
2-2	sSC-A/AW	Aktiviert den Sensor in der siot.net Gateway Library (Verantwortung für das Manifestieren und die Verwaltung der Kommunikation wird an die Bibliothek übergeben).
2-3	sSC-A/AW	Sensor im GUI als eingeschaltet darstellen.

Tabelle 8.25: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #2.1 und #2.2

Ausnahmen und Varianten #2.1 und #2.2:

-	-	Keine Ausnahmen und Varianten
---	---	-------------------------------

Tabelle 8.26: siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #2.1 und #2.2

8.2.3.3 Anwendungsfall #3.1 und #3.2

Nr. und Name:	#3.1 und #3.2 Sensormessungen stoppen.
Szenario:	Sensor wird auf Android oder Android Wear Gerät deaktiviert.
Kurzbeschreibung:	Jeder aktivierte Sensor kann, durch die auf dem GUI sichtbaren Schalter, ausgeschaltet werden.
Beteiligt Akteure:	App User, siot.net Gateway Library.
Auslöser und Vorbedingung:	Ein oder mehrere Sensoren werden mit der App ausgeschaltet.
Ergebnisse und Nachbedingung:	Sensor kann erfolgreich deaktiviert werden.

Tabelle 8.27: siot.net Sensorcenter: Übersicht Anwendungsfall #3.1 und #3.2

Ablauf #3.1 und #3.2:

3-1	App User	Einen gewählten Sensor mittels Schalter deaktivieren.
3-2	sSC-A/AW	Aktiviert den Sensor in der siot.net Gateway Library (Verantwortung für das Manifestieren und die Verwaltung der Kommunikation wird an die Bibliothek übergeben).
3-3	sSC-A/AW	Sensor im GUI als ausgeschaltet darstellen.

Tabelle 8.28: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #3.1 und #3.2

Ausnahmen und Varianten #3.1 und #3.2:

-	-	Keine Ausnahmen und Varianten
---	---	-------------------------------

Tabelle 8.29: siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #3.1 und #3.2

8.2.3.4 Anwendungsfall #4.1 und #4.2

Nr. und Name:	#4.1 und #4.2 Abmelden von siot.net.
Szenario:	Alle Sensoren stoppen und die Verbindung zu siot.net trennen.
Kurzbeschreibung:	Die durch die App gestarteten Sensoren werden deaktiviert.
Beteiligt Akteure:	App User, siot.net Gateway Library.
Auslöser und Vorbedingung:	Ein oder mehrere Sensoren werden mit der App ausgeschaltet.
Ergebnisse und Nachbedingung:	Sensor kann erfolgreich deaktiviert werden.

Tabelle 8.30: siot.net Sensorcenter: Übersicht Anwendungsfall #4.1 und #4.2

Ablauf #4.1:

4.1-1	App User	„Abmelden“ Button geklickt.
4.1-2	sSC-A	Deaktiviert alle in der App aktivierten Sensoren. Meldung an sSC-AW, dass abgemeldet wird.
4.1-3	sSC-AW	Schaltet alle Sensoren aus, welche der User aktiviert hat. Die Verbindung wird getrennt.
4.1-4	sSC-AW	Anmelde Ansicht wird angezeigt.
4.1-5	sSC-A	Gibt der siot.net Gateway Library den Befehl die Verbindung zum MQTT Broker zu schließen.
4.1-5	sSC-A	Display zeigt das Login GUI.

Tabelle 8.31: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #4.1

Ablauf #4.2:

4.2-1	App User	„Abmelden“ Button geklickt.
4.2-2	sSC-AW	Schaltet alle Sensoren aus, welche der User aktiviert hat. Die Verbindung wird getrennt.
4.2-3	sSC-AW	Anmelde Ansicht wird angezeigt.

Tabelle 8.32: siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #4.2

Ausnahmen und Varianten #4.1 und #4.2:

4.1-2.1	sSC-A	Kein Gerät verbunden welche sSC-AW ausführt, keine Aktion notwendig.
----------------	-------	--

Tabelle 8.33: siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #4.1 und #4.2

8.2.4 Nicht-funktionale Anforderungen

Produktanforderungen:

Benutzbarkeitsanforderungen	Die Applikation muss mit einem Android Smartphone oder einer Android Smartwatch bedienbar sein. Die Smartwatch muss die Funktion Always-On Display oder Ambient Modus beherrschen.
Effizienzanforderungen	Die Bedienung muss einfach gehalten werden, erleichtert die Verwendung für aller Art Benutzer.
Zuverlässigkeitssanforderungen	Displayanzeigen müssen verzögerungsarm sein. Speicherung von mindestens einem Lizenzschlüssel muss gewährleistet werden. Verbindung zum Internet muss vorhanden sein.
Portierbarkeitsanforderungen	Es werden nur Daten gespeichert die für diese Applikation notwendig ist. Es gibt keine Portierbarkeitsanforderungen.
Performanceanforderung	Die Anzeige muss immer in einer Menschen brauchbaren Geschwindigkeit reagieren. Es braucht eine Internetverbindung, um Sensordaten zu senden braucht es keine grosse Bandbreite. Die Daten Pakete sind sehr klein (<1KB).

Tabelle 8.34: siot.net Sensorcenter: Nicht-funktionale Produktanforderungen

8.3 Requirements - siot.net Dashboard

In diesem Requirementsdokument werden die Anforderung an die Android App siot.net Dashboard (**sDB**) spezifiziert.

8.3.1 Allgemeine Beschreibung

Das siot.net Dashboard für Geräte mit dem Android Betriebssystem, ist eine App welche die siot.net Gateway Library (**sGWLib**) integrieren soll. Auf einem grossen Touchscreen können ermittelte Daten, übersichtlich dargestellt werden. Es soll mindestens die Anzeige von Graphen, Landkarten und Freitext Wertanzeige möglich sein.

8.3.2 User Requirements

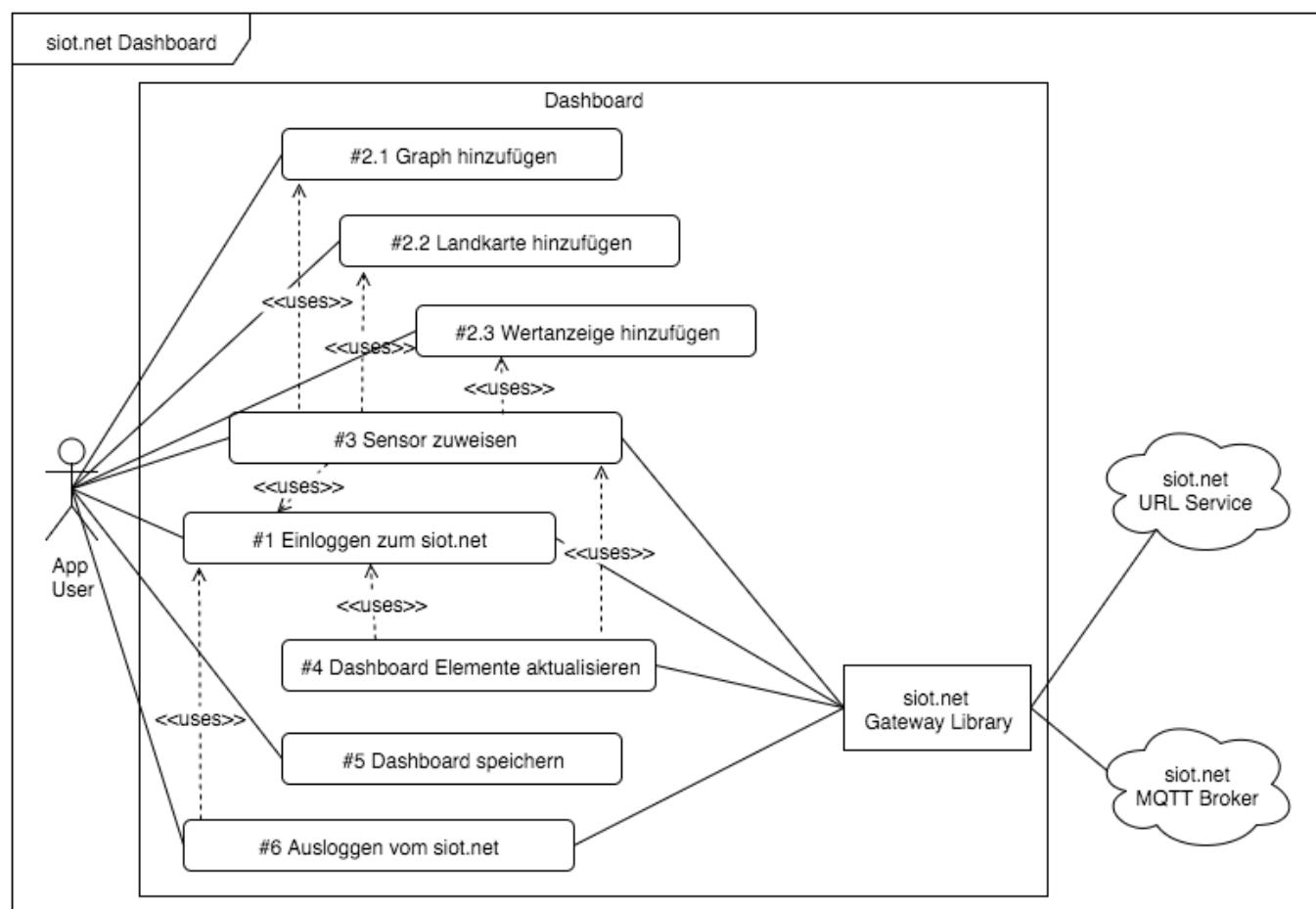


Abbildung 8.3: Das allgemeine Use Case Diagramm für das siot.net Dashboard

8.3.3 Funktionale Anforderungen und Anwendungsfallbeschreibungen

8.3.3.1 Anwendungsfall #1

Dieser Anwendungsfall #1 ist Deckungsgleich mit dessen im Kapitel 7.2.3, dem Anwendungsfall #1.1. Das Login verfahren wird in beiden Applikationen identisch behandelt. Das Dashboard soll nur auf grösseren Android Geräten zur Verfügung stehen. Auf einem kleinen Display, der Wearable Devices, wäre die Darstellung der Informationen nicht übersichtlich.

8.3.3.2 Anwendungsfall #2

Nr. und Name:	#2.1 Graph, #2.2 Landkarte und #2.3 Wertanzeige hinzufügen.
Szenario:	Ein Anzeigeelement dem Dashboard hinzufügen.
Kurzbeschreibung:	Der Applikation sollen während dem Betrieb, neue Objekte hinzugefügt werden können um Informationen anzuzeigen.
Beteiligt Akteure:	App User.
Auslöser und Vorbedingung:	User wählt ein neues Element aus.
Ergebnisse und Nachbedingung:	Das ausgesuchte Anzeigeelement ist auf dem App sichtbar.

Tabelle 8.35: siot.net Dashboard: Übersicht Anwendungsfall #2

Ablauf #2:

2-1	App User	Wählt einen Graph, Landkarte oder Freitext Wertanzeige aus.
2-2	sDB	Generiert das neue GUI Element und eine leere Auswahlliste.
2-3	sDB	Zeigt beide Objekte an.

Tabelle 8.36: siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #2

Ausnahmen und Varianten #2:

2-2.1	sDB	Variante: Dem siot.net Dashboard sind bereits die Sensoren vom IoT Center bekannt, Auswahliste kann bereits befüllt werden.
-------	-----	---

Tabelle 8.37: siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #2

8.3.3.3 Anwendungsfall #3

Nr. und Name:	#3 Sensor zuweisen.
Szenario:	Einer sichtbaren Informationsanzeige einen dazugehörigen Sensor zuweisen.
Kurzbeschreibung:	Die im Anwendungsfall #2 generierte Auswahliste soll mit passenden Sensoren gefüllt werden. Die Daten eines ausgewählten Sensors sollen auf der Anzeige dargestellt werden.
Beteiligt Akteure:	App User, siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	Neues Anzeigeelement wurde erzeugt, User will ein Sensor dazu referenzieren.
Ergebnisse und Nachbedingung:	Daten des ausgewählten Sensors werden auf dem Graph, der Landkarte oder der Wertanzeige dargestellt.

Tabelle 8.38: siot.net Dashboard: Übersicht Anwendungsfall #3

Ablauf #3:

3-1	App User	Wählt eine Auswahlliste eines Anzeigeobjekt.
3-2	sDB	Verlangt von der siot.net Gateway Library alle vorhanden Sensoren mit passendem Typ.
3-3	sDB	Zeigt die verfügbaren Sensoren an.
3-4	App User	Wählt einen Sensor an.
3-5	sDB	Abonniert die Informationen bei der siot.net Gateway Library.
3-6	sDB	Anzeige wird mit den Daten dargestellt und synchronisiert.

Tabelle 8.39: siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #3

Ausnahmen und Varianten #3:

3-2.1	sDB	Ausnahme: Keine Sensoren vorhanden, keine Aktion möglich, Auswahlliste bleibt leer.
3-6.1	sDB	Ausnahme: Keine Sensorendaten vorhanden, Anzeige wird erst angereichert wenn Daten eintreffen.

Tabelle 8.40: siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #3

8.3.3.4 Anwendungsfall #4

Nr. und Name:	#4 Dashboard Elemente aktualisieren.
Szenario:	Neu verfügbare Daten den Anzeigeobjekte zur Verfügung stellen.
Kurzbeschreibung:	Die im Anwendungsfall #3 abonnierte Daten müssen verarbeitet werden. Ein klassisches Model-View-Controller Pattern kommt hier zum Einsatz.
Beteiligt Akteure:	siot.net Gateway Library, siot.net MQTT Broker.
Auslöser und Vorbedingung:	siot.net Gateway Library empfängt neue MQTT Message und notifiziert das Dashboard.
Ergebnisse und Nachbedingung:	Graph, der Landkarte oder der Wertanzeige wird aktualisiert.

Tabelle 8.41: siot.net Dashboard: Übersicht Anwendungsfall #4

Ablauf #4:

4-1	sDB	Erhält die Benachrichtigung von der siot.net Gateway Library, dass neue Daten vorhanden sind.
4-2	sDB	Analysiert die Daten.
4-3	sDB	Aktualisiert die zu den Informationen gehörigen Anzeigen.

Tabelle 8.42: siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #4

Ausnahmen und Varianten #4:

3-2.1	sDB	Ausnahme: Keine Daten welche relevant sind, keine Aktion nötig.
--------------	-----	---

Tabelle 8.43: siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #4

8.3.3.5 Anwendungsfall #5

Nr. und Name:	#5 Dashboard speichern.
Szenario:	Das erstellt Dashboard speichern.
Kurzbeschreibung:	Speichern des Dashboardes um es ein anderes Mal wieder zu verwenden oder auch anderen Usern im gleichen IoT Center bereit zu stellen.
Beteiligt Akteure:	App User.
Auslöser und Vorbedingung:	User drückt „Dashboard speichern“ Taste.
Ergebnisse und Nachbedingung:	Dashboard kann in einer Datei persistiert werden.

Tabelle 8.44: siot.net Dashboard: Übersicht Anwendungsfall #5

Ablauf #5:

5-1	App User	Löst die Aktion aus durch betätigen des „Dashboard speichern“ Knopfes.
5-2	sDB	Struktur und Zuordnungen werden in eine Datei auf dem Gerät niedergeschrieben.
5-3	sDB	User erhält Bestätigung.

Tabelle 8.45: siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #5

Ausnahmen und Varianten #5:

5-2.1	sDB	Ausnahme: Datei kann nicht geschrieben werden, User benachrichtigen.
--------------	-----	--

Tabelle 8.46: siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #5

8.3.3.6 Anwendungsfall #6

Nr. und Name:	#6 Ausloggen vom siot.net.
Szenario:	Löst die Verbindung zu der IoT Cloud auf.
Kurzbeschreibung:	Dieser Schritt erlaubt eine saubere Abmeldung vom Netzwerk. Hier wird der Impuls der siot.net Gateway Library gegeben um alle Verbindungen Sauber zu trennen.
Beteiligt Akteure:	App User, siot.net Gateway Library.
Auslöser und Vorbedingung:	„Abmelden“ Schaltfläche wird aktiviert.
Ergebnisse und Nachbedingung:	Dashboard erfolgreich vom siot.net abgemeldet.

Tabelle 8.47: siot.net Dashboard: Übersicht Anwendungsfall #6

Ablauf #5:

5-1	App User	Abmelden wird ausgeführt.
5-2	sDB	Meldet dem siot.net Gateway zum trennen der Verbindungen.
5-3	sDB	Login Bildschirm wird angezeigt.

Tabelle 8.48: siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #6

Ausnahmen und Varianten #6:

-	-	Keine Ausnahmen und Varianten
---	---	-------------------------------

Tabelle 8.49: siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #6

8.3.4 Nicht-funktionale Anforderungen

Produktanforderungen:

Benutzbarkeitsanforderungen	Die Applikation muss mit einem Android Smartphone funktionstüchtig sein.
Effizienzanforderungen	Die Anwendung muss für aller Art Smartphone benutzen bar sein.
Zuverlässigkeitssanforderungen	Graphen, Landkarten und Werteanzeigen sollten möglichst flüssig angezeigt und aktualisiert werden.
Portierbarkeitsanforderungen	Speichern eines Dashboards soll möglichst in einer generischen Struktur geschehen. In einem weiteren Release wird in Betracht gezogen, dass Dashboards portierbar sind, das heißt sie können auch im siot.net IoT Center gespeichert werden.
Performanceanforderung	Die Anzeige muss flüssig reagieren um die Graphen brauchbar zu visualisieren. Es braucht eine Internetverbindung, um Sensorinformationen zu empfangen. Da die Datenpakete sehr klein sind (<1KB), ist keine sehr grosse Bandbreite notwendig.

Tabelle 8.50: siot.net Dashboard: Nicht-funktionale Produktanforderungen

9 Konzepte

Diese Kapitel dokumentiert das Softwaredesign der siot.net Gateway Library und des siot.net Sensorcenter.

9.1 Konzept - siot.net Gateway Library

9.1.1 Packagediagramm

Das auf Abbildung 9.1 visualisierte Paketdiagramm, zeigt die Grobeinteilung der Klassen und deren Abhängigkeiten.

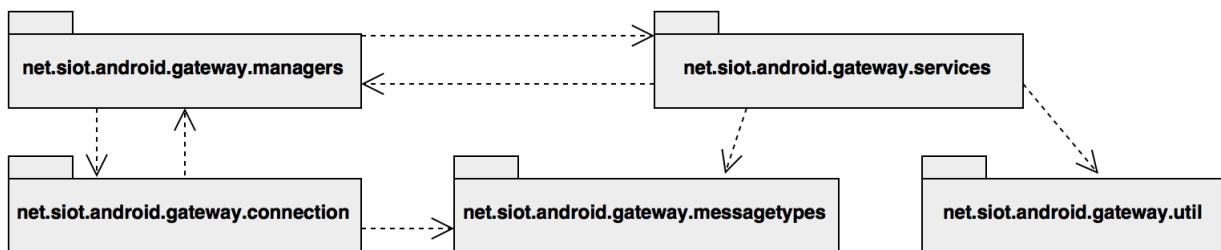


Abbildung 9.1: Java-Package Aufteilung der siot.net Gateway Library

Die Auflistung der einzelnen Packages mit den enthaltenen Klassen (Verwendung von Entwurfsmuster sind in Klammern).

net.siot.android.gateway.managers

- SiotNetGatewayManager (Abstrakte Fabrikklasse)
- SiotNetGatewayManagerMobile (Fabrikklasse)
- SiotNetGatewayManagerWear (Fabrikklasse)
- ReceivedMessageManager(Beobachtbare-Klasse)

net.siot.android.gateway.services

- SensorService (Abstrakte Fabrikklasse)
- SensorServiceMobile (Fabrikklasse)
- SensorServiceWear (Fabrikklasse)
- LocationService

net.siot.android.gateway.connection

- MQTTClient (Singleton Klasse)
- RestClient

net.siot.android.gateway.util

- GUIDUtil
- TopicUtil
- SensorTypeKeys

net.siot.android.gateway.messagetypes

- SensorActorManifest
- SensorConfig
- ActorConfig
- SiotUrl
- Urls
- WearableData

Tabelle 9.1: siot.net Gateway Library: Auflistung der Packageinhalte

9.1.2 Domäendiagramm

Auf der folgenden Abbildung 9.2, ist das gesamte konzeptionelle Domäendiagramm der siot.net Gateway Library aufgezeichnet.

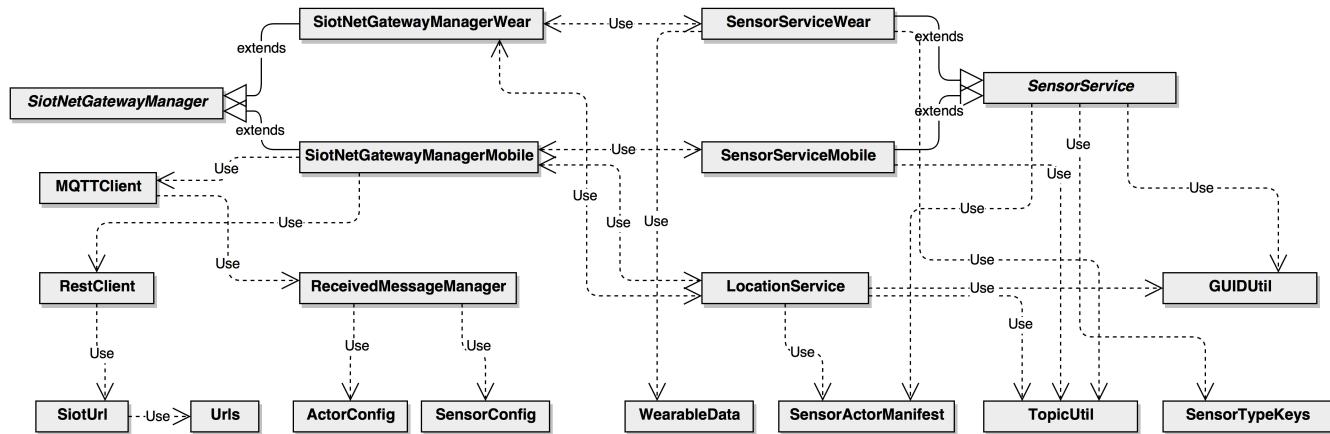


Abbildung 9.2: Die Gesamtübersicht aller Klassen der siot.net Gateway Library

Die Managers sind die zu instanzierenden Klassen einer App welche dieses Paket integriert. Die SiotNetGatewayManagers sind in Anlehnung des Designpatterns der Fabrikmethode¹ konzipiert, falls es no weitere Android Gerätearten hinzukommen, kann eine dazupassende Fabrik dazuprogrammiert werden (z.B. SiotNetGatewayManagerCar). Die ReceivedMessageManager Klasse ist für den Gebrauch in einem Beobachter-Entwurfsmuster² vorbereitet. Sie ist Observable und sollte von einem Observer Objekt instanziert werden. Von diesem Objekt darf nur eine Instanz erstellt werden, deswegen kommt hier das Singleton-Entwurfsmuster³ zur Anwendung.

Die Serviceklassen beinhalten die Verwaltung und die Kommunikation von den Sensoren und den Ortungsdaten. Die Sensorservices sind ebenfalls mit der Fabrikmethode konzipiert.

Das Connection-Paket ist für die Verbindungsobjekte. Der MQTT Broker wird gleichwohl, wie der ReceiverMessageManager, mit einer Singleton-Klasse angesteuert. RestClient implementiert die statische Methode zum konsultieren des URL-Services.

In Utils sind Helferklassen enthalten und in Messagetypes werden die Datentypen der Kommunikation definiert.

¹vgl. Design Patterns; Chapter 3. Creational Patterns; E.Gamma, R.Helm, R.Johnson, J.Vlissides

²vgl. Design Patterns; Chapter 5. Behavioral Patterns; E.Gamma, R.Helm, R.Johnson, J.Vlissides

³vgl. Design Patterns; Chapter 3. Creational Patterns; E.Gamma, R.Helm, R.Johnson, J.Vlissides

9.1.3 Sequenzdiagramme

Die grundlegenden Abläufe sind mit Sequenzdiagrammen spezifiziert.

9.1.3.1 App starten

Die Vorgänge welche durchlaufen werden, beim starten einer Smartphone und Smartwatch App, sind in der Abbildung 9.3 und 9.5 illustriert.

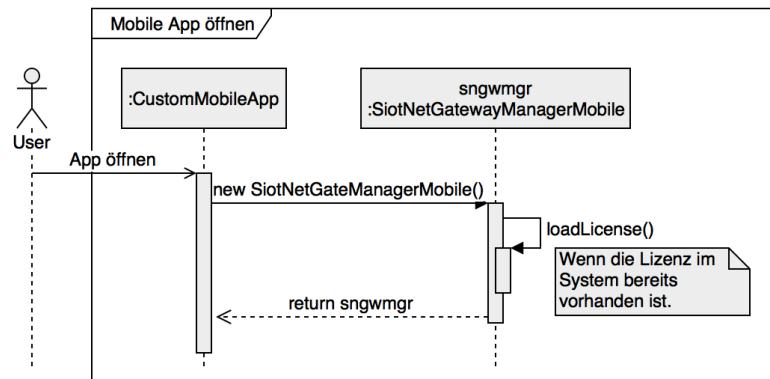


Abbildung 9.3: Sequenzdiagramm: Starten von Smartphone App

Das normale Starten einer Smartphone Applikation, welche die siot.net Gateway Library integriert, wird auf der Abbildung 9.3 geschildert.

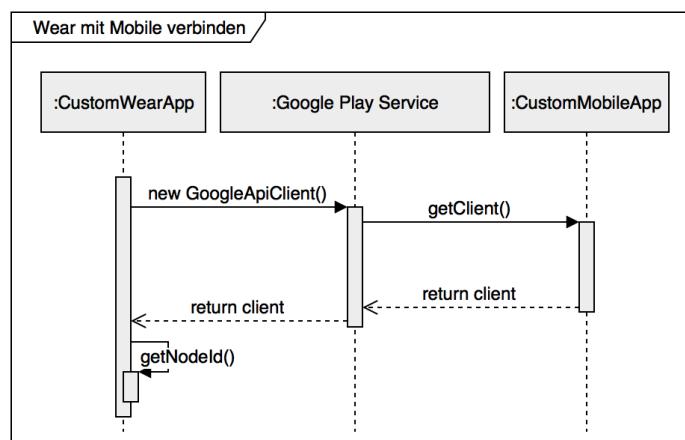


Abbildung 9.4: Sequenzdiagramm: Verbindungsaufbau von Smartwatch zu Smartphone

Wenn die zur vorgehenden Diagramm erwähnten App dazugehörige Smartwatch Anwendung aufgerufen wird, verbindet dieses sich zuerst via Google Play Service zum gekoppelten Mobiltelefon. Dazu die erklärende Abbildung 9.4. Das danach folgende Sequenzdiagramm auf Abbildung 9.5 hat eine Abhängigkeit darauf.

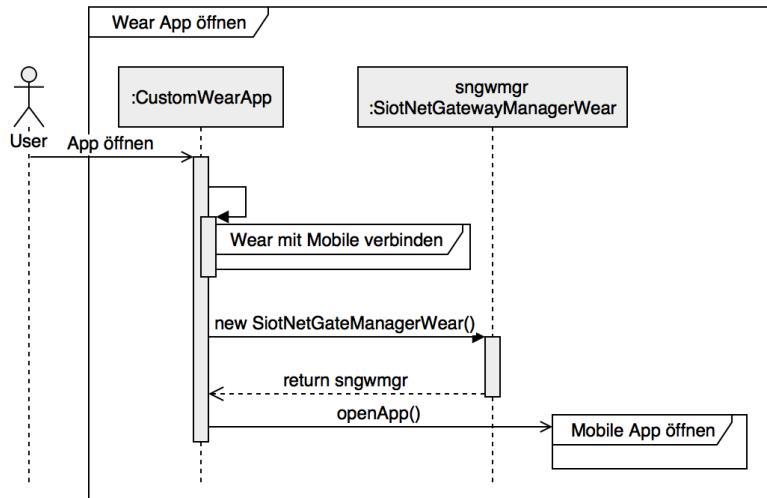


Abbildung 9.5: Sequenzdiagramm: Starten von Smartwatch App

Der Minicomputer am Handgelenk kann die siot.net Schnittstelle nur dann zur Verfügung stellen, wenn die Verbindung zum Smartphone aktiv ist. Zum erzeugen des Managerobjektes muss der vorher erhaltene GoogleApiClient mitgegeben werden.

9.1.3.2 Login zu siot.net

Die Kommunikationsabfolgen für das Login zum siot.net sind auf Abbildung 9.6 und 9.7 visualisiert.

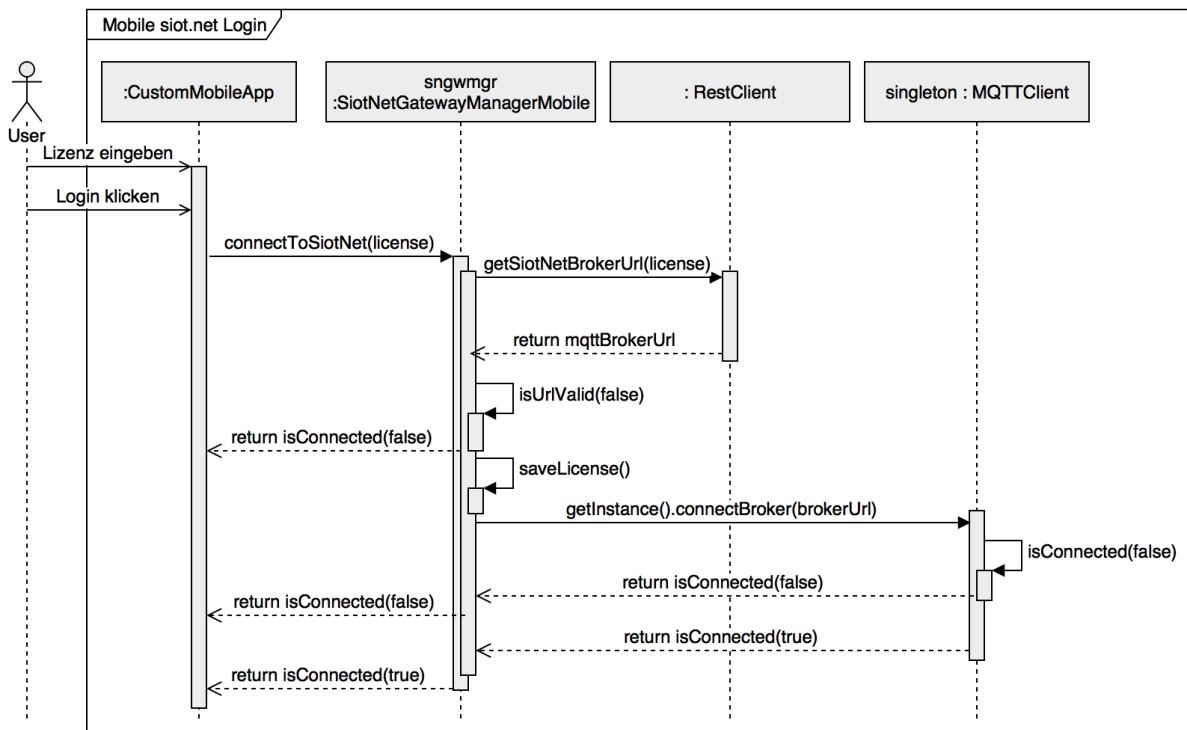


Abbildung 9.6: Sequenzdiagramm: Einloggen von einer Smartphone App

Abbildung 9.6 erläutert das Einloggenverfahren über ein Android Mobile Gerät. Für das Login benötigt es immer einen gültiger siot.net Lizenzcode. Nach einer erfolgreichen Authorisierung beim siot.net URL-Service wird der Schlüssel im System in einer Konfigurationsdatei abgelegt. Dieser wird bei jedem Start geladen (siehe Abbildung 9.3). Aus dieser positiven Bestätigung kann die Adresse des MQTT Broker gelesen werden. Der weitere Verlauf ist auf der Abbildung 9.6 ersichtlich.

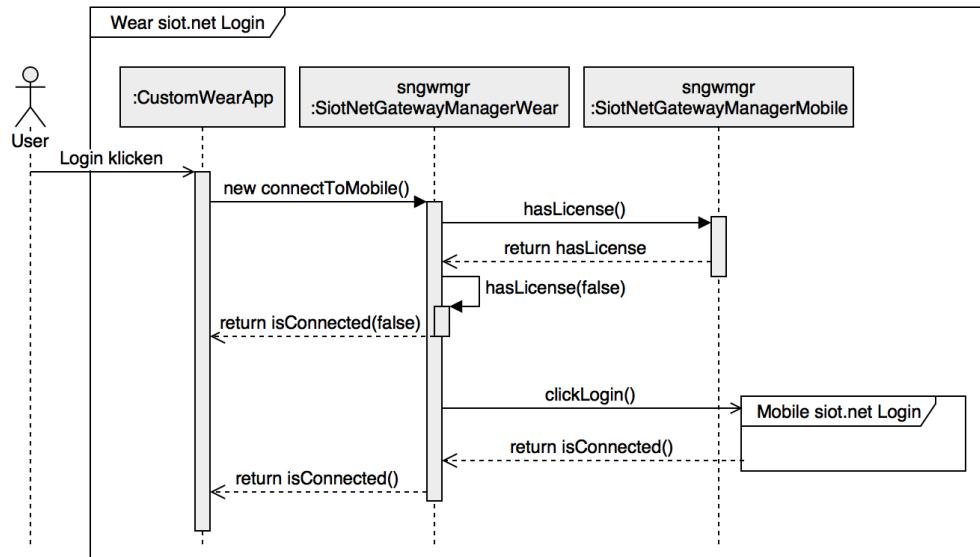


Abbildung 9.7: Sequenzdiagramm: Einloggen von einer Smartwatch

Die Authorisierung via Smartwatch geschieht nur indirekt, dieser verlangt, dass bereits ein valider Lizenzcode auf dem Smartphone existiert. Dies ist notwendig, da die Android Wear Geräte keine virtuelle Tastatur besitzen. Eine Eingabe wär nur per Sprachbefehl möglich, was die siot.net Gateway Library nicht unterstützt. Ist dieser vorhanden, fordert die Datenuhr das Mobiltelefon dazu auf sich mit dem siot.net MQTT Broker zu verbinden. Die Abbildung 9.7 stellt das genaue Sequenzdiagramm zu diesem Vorgang dar.

9.2 Konzept - siot.net Sensorcenter

9.2.1 Domänenendiagramm

Die Abbildung 9.8, ist das Domänenendiagramm vom siot.net Sensorcenter. Die Klassenstruktur der Endanwenderapp kann

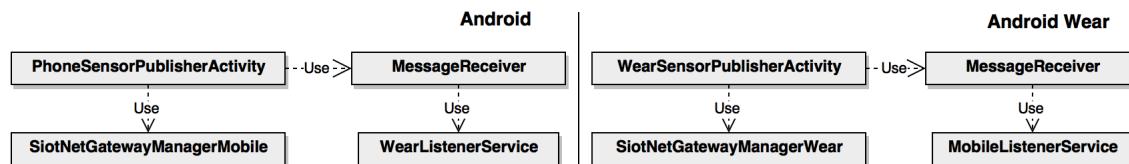


Abbildung 9.8: Die Klassenübersicht vom siot.net Sensorcenter im Domänenendiagramm

sehr schlank gehalten werden, da die meisten Aufgaben die siot.net Gateway Library den grossen Anteil bereits erledigt. Es ist notwendig, dass zwei fast identische Applikationsstrukturen in einem Projekt bestehen. Die Wearable App differenziert sich erheblich zur Smartphone App. Desweiteren ist von Android vorgegeben, dass für die Smartphone Activity und die Watch Activity eine separates Package erstellt wird⁴. Die Kernelemente jeweiliger Applikationen sind die Activityklassen. Diese sind zuständig für die grafische Benutzeroberfläche und instanzieren die siot.net Gateway Libraray. Das nötige

⁴vgl. <http://developer.android.com/sdk/installing/create-project.html>, 18.12.2015

Managerobjekt wird mit Hilfe der passenden Manager-Fabrik erzeugt. Der ListenerService und der MessageReceiver sind zuständig für die Kommunikation zwischen Smartphone und Smartwatch.

9.2.2 Sequenzdiagramm

Die folgenden Abbildungen 9.9 und 9.10 legen fest, wie die Sensorsteuerung, in Kombination mit der Gateway Bibliothek, von stattent geht.

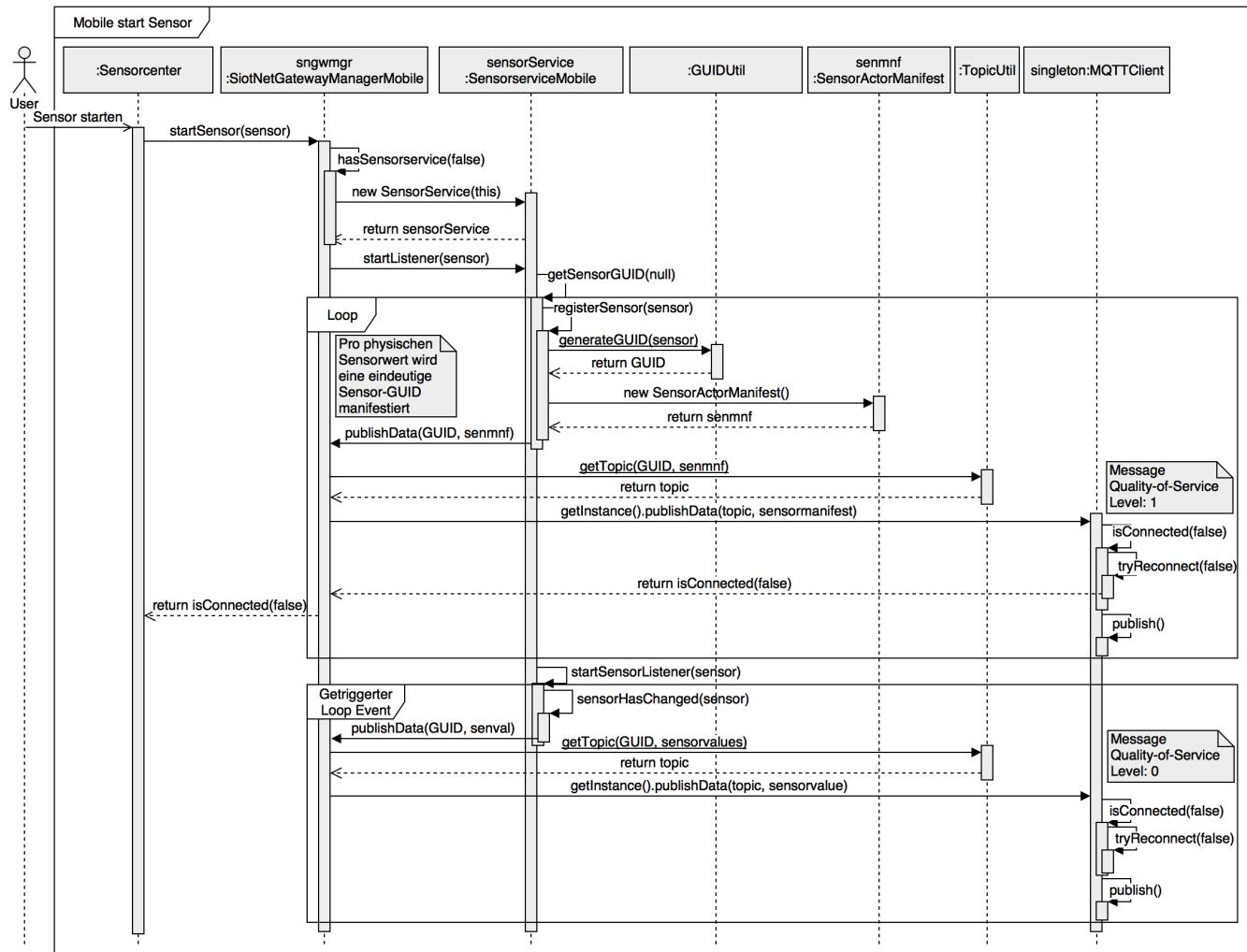


Abbildung 9.9: Der Ablauf beim starten eines Sensors im Sensorcenter

Die erste Spezialität in Abbildung 9.9 ist beim **Loop** Rahmen. Dieser Turnus kommt dann zum Zug, wenn der Sensor zum ersten Mal gestartet wird. Ein Sensor muss sich unter Umständen mehrfach manifestieren aufgrund mehreren physischen Werten. Beispielsweise hat ein Beschleunigungsmesser in einem dreidimensionalen System die Werte der X-Achse, Y-Achse und Z-Achse, d.H. pro Achse wird ein Manifest des Sensors erzeugt.

Nach der erfolgreichen Anmeldung des Sensors, wird der vom Androidsystem bereitgestellten Zuhörer des Sensor aktiviert. Diese gibt bei jeder Veränderung einen Messwert zurück, dass dann eine MQTT Nachricht an den siot.net Broker auslöst. Solange der Messwertegeber nicht ausgeschaltet ist, wird nach jeder Auslösung des Sensorlisteners die Sequenz im Rahmen **Getrigerter Loop Event** durchlaufen.

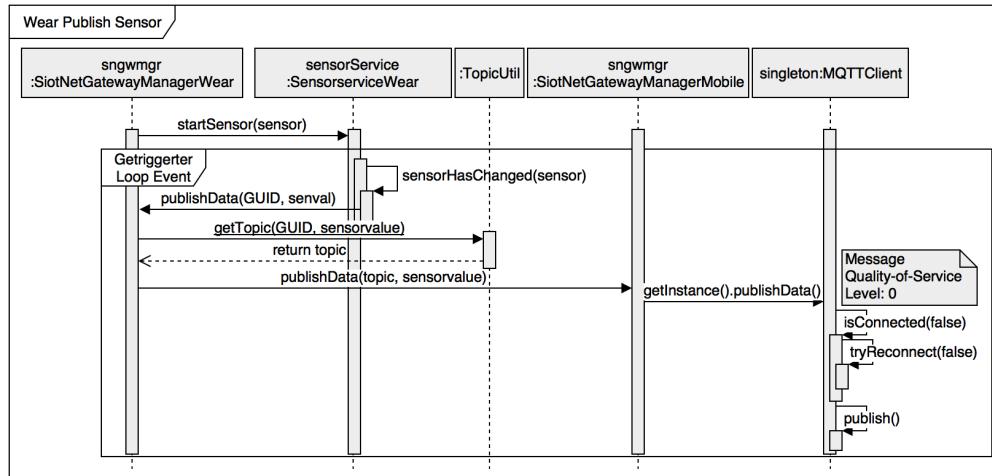


Abbildung 9.10: Kommunikationsverhalten beim senden von Daten bei Smartwatch

Das Auslösen von Nachrichten bei der Smartwatch benötigt noch einen weiteren Schritt. Durch die Beschränkung besteht, Daten nicht direkt ins Internet zu verbreiten, hier zum MQTT Broker, muss das Datenpaket zuerst zum Smartphone. Dieses wird als Datenbrücke verwendet, wie auf Abbildung 7.5 und 7.6 geschildert, um die Informationen durchzuschleusen.

9.3 Konzept - Kommunikation Smartphone-zu-Smartwatch

Für die Kommunikation von Smartphone zu Smartwatch und umgekehrt wurde ein Kommunikationskonzept zugewiesen. Die Kommunikation soll ausschliesslich über die Android MessageAPI verlaufen.

9.3.1 Request-Response

Anweisungen und Informationen welche zwischen den zwei Geräten ausgetauscht werden, müssen im Request-Response Verfahren ablaufen. Dies ist notwendig, weil Aktionen des einen Gerätes, Einfluss auf das andere haben kann. Kommunikation und Funktionalität auf dem Smartwatch sind stark gebunden mit dem Status des Smartphones. Darum müssen die Messages gegenseitig quittiert werden.

9.3.2 Fire-and-Forget

Bei den Sensormesswerten der Smartwatch ist dies anders geregelt. Diese werden im Fire-and-Forget verfahren verschickt. D.h. die Daten werden ausgesendet aber werden nicht verfolgt ob diese korrekt übermittelt wurden. Die wird angewendet, weil es System unkritische Informationen sind, dadurch kann Energie und Bandbreite gespart werden. Ausnahme: Wenn bei der MQTT Kommunikation, der Quality-of-Service > 0 wird (1 oder 2), müssen die Messwerte von der Smartwatch im Nachrichtenaustauschmuster Request-Response übergeben werden.

10 Implementation

Dieses Kapitel beleuchtet einige Gesichtspunkte der Umsetzung. Die Logik, Abgrenzungen, Einhaltung des Konzepts, verwendete Plattformen und Libraries gehören dazu.

10.1 Abgrenzung

Zu Beginn der Realisierung mussten die Ziele abgegrenzt werden. Der Umfang vom definierten Anforderungskatalog und Konzept ist sehr gross. Für die Umsetzung wurden einige Kernelemente ausgesucht.

10.1.1 Ziele - siot.net Gateway Library

- Die siot.net Gateway Library als Prototypen realisiert werden.
- Mindestens die Verbindung zum siot.net aufnehmen können.
- Sensoren nach dem siot.net Messagedesign¹ manifestieren.
- Daten an die IoT-Cloud im richtigen Format senden.
- In einer Applikation integriert werden.

10.1.2 Ziele - siot.net Sensorcenter

- Als Demoapplikation für die Gateway Library umsetzen.
- Für Smartphone und Smartwatch lauffähig.

10.2 Implementation - siot.net Gateway Library

Die Umsetzung erfolgte grösstenteils nach dem konzeptionierten Domänenmodell (siehe Abbildung 9.2). Nicht alle Funktionen wurden realisiert. LocationService und der ReceivedMessageManager sind nicht erfolgreich funktional umgesetzt worden. Der LocationService wurde ausgelassen, da dieser sich sehr dem SensorService ähnelt. Beim ReceivedMessageManager wurde bei der Umsetzung festgestellt, dass das Konzept nicht konsistent den Nutzen abdeckt. Dieser Manager muss überarbeitet und womöglich neu konzipiert werden.

10.2.1 Klassendiagramm

Auf der folgenden Seite ist, auf Abbildung 10.1, das Klassendiagramm der siot.net Gateway Library. Die realisierten Klassen sowie die entworfenen jedoch nicht funktional umgesetzten Objekte (weisser Hintergrund) sind auf dieser Grafik visualisiert.

¹vgl. siot.net Cookbook Version 1.0; siot.net; 2015

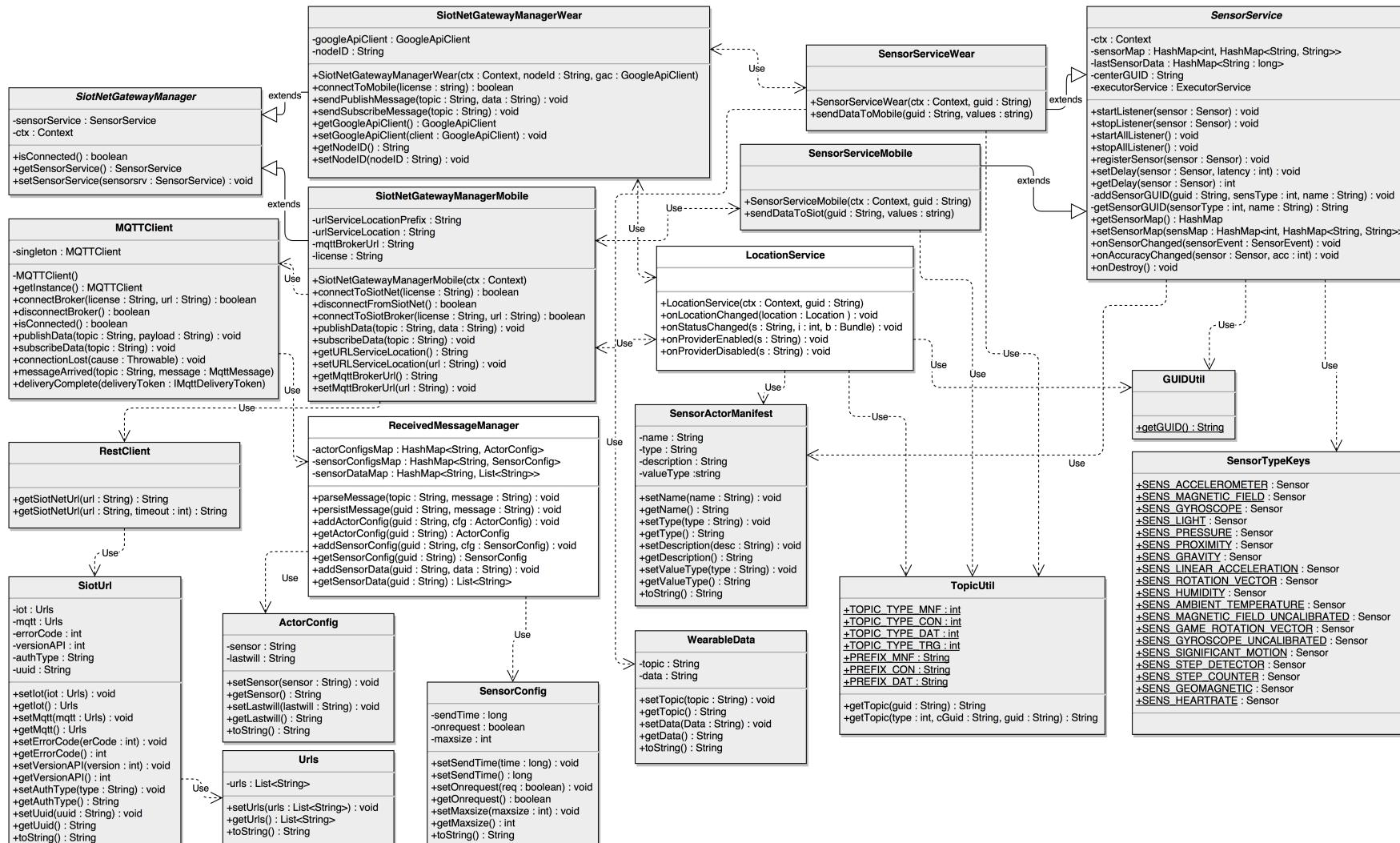


Abbildung 10.1: Die Gesamtübersicht aller Klassen der siot.net Gateway Library

10.2.2 Logik

Einige interessante Logikaspekte sind in den SiotNetGatewayManagerKlassen enthalten. Mit Hilfe der MobileManagers benötigt es für den Verbindungsaufbau zum MQTT Broker nur die siot.net Lizenz. Beim ausführen der connectToSiotNet(Lizenz) Methode, wird per REST API ein Authentifikationstoken angefordert. Um ein Solches zu erhalten wird ein HTTP GET request abgesetzt, welches eine Bestätigung oder Zurückweisung im JSON-Format retourniert. Diese ist in der RestClient Klasse implementiert. Die Antwort wird auf den MessageType SiotUrl geparsed. Die Broker URL wird aus der positiven Bestätigung gelesen.

Mit der aktuell erhaltenen Broker URL kann die Kopplung zum MQTT Broker starten. MQTTCClient Klasse dient für diese Angelegenheit. Der MQTTCClient Objekttyp ist eine nach dem Prinzip des Singleton-Entwurfsmuster² implementierte Klasse. Von der Library soll immer nur eine Objekt der Klasse MQTTCClient bestehen, da dieser die Verbindung zum Broker verwaltet, also die Meldungen sendet und auch empfängt. Das Empfangen von Nachrichten durch mehrere Clients würde redundante Daten schaffen, wenn mehrere Subscriber auf dieselbe Topic hören. Desweitern würde die Netzwerkverbindung und Stromversorgung unnötig belastet werden. Der Singleton eignet sich ideal, diese Probleme zu beseitigen.

Um ein Objekt aus der SiotNetGatewayManagerWear Fabrikklasse (Entwurfsmuster Fabrikmethode³) zu erzeugen, ist zwingend eine gültige Verbindung zum Smartphone nötig. Dieser wird in Form von einer GoogleApiClient Instanz und der Nodeld des Smartphone verlangt. Beim instanziieren muss die Überprüfung der gültigen Kopplung von Mobile und Wearable bestanden werden. Im Konzept wurde eine Überprüfung nicht in Erwägung gezogen. Die vorhandene Verbindung ist essentiel für die Funktionsweise der Smartwatch-App.

Nach erfolgreichen kreieren der Manager Elemente, können die Sensoren aktiviert werden. Die SensorenServiceKlassen sind auch nach der Fabrikmethode konzipiert und implementiert. Sensoren müssen bei der ersten Aktivierung beim siot.net manifestiert werden. Eine Besonderheit dabei ist, dass jeder physische Sensorwert einzeln angemeldet wird. Dies muss in der Applikation verwaltet werden, für dies wurde eine zweidimensionale Hashmap verwendet. Die erste Dimension enthält den Schlüssel des SensorTypes, welcher eindeutig vom System vergeben ist, verknüpft mit der zweiten Hashmap. Die zweite beinhaltet den Namen des Messwertes (z.B. X-Achse) als Schlüssel und die siot.net GUID des physischen Sensorswertes. Beim stoppen und starten eines Sensor kann die GUID anhand vorhandener Informationen (SensorType und Messwertenamen) und der Zuweisung in der Hashmap evaluiert werden. Die GUID wird, für das Manifestieren des Sensors, durch die GUIDUtil Klasse generiert. Jeder Senorservice ist auch für das versenden der Messwerte zuständig. Der MobileService sendet diese direkt per MQTT an den Broker. Der WearService sendet eine MessageAPI Nachricht im Format des MessageTypes WearableData an das Smartphone.

Bei ReceivedMessageManager Klasse wurde während der Entwicklung gestoppt, da erkannt wurde das, dass Konzept für die Bedürfnisse nicht genügend konstruiert war. Bei einer Umsetzung wie in Abschnitt 9.1 beschrieben, wäre dies zu Dateninkonsistenzen und Redundanzen gekommen. Ein erster Ansatz wurde in Betracht gezogen. Die Kombination vom Singleton und dem Observer-Pattern waren die Überlegungen, jedoch gibt es noch kein konkretes erneuertes Konzept dazu.

10.3 Implementation - siot.net Sensorcenter

Vom Domäendiagramm (siehe Abbildung 9.8) wurde leicht abgewichen. Die MessageReceiver Klasse wurde für Android und Android Wear als Lokale innere Klasse der Activity Klasse realisiert. Dies hat den Vorteil, die Objekte der äusseren Klasse kennt und diese selbst nicht nach aussen sichtbar ist. Das Kernelement dieser Applikation ist das Kommunikationsverhalten von Smartwatch und Smartphone. Weitere Informationen dazu im Abschnitt Logik.

²vgl. Design Patterns; Chapter 3. Creational Patterns; E.Gamma, R.Helm, R.Johnson, J.Vlissides

³vgl. Design Patterns; Chapter 3. Creational Patterns; E.Gamma, R.Helm, R.Johnson, J.Vlissides

10.3.1 Klassendiagramm

Abbildung 10.2 stellt klar, wie kompakt die Entwicklung einer externe Applikation sein kann. Sie zeigt alle Klassen der Android und Android Wear App.

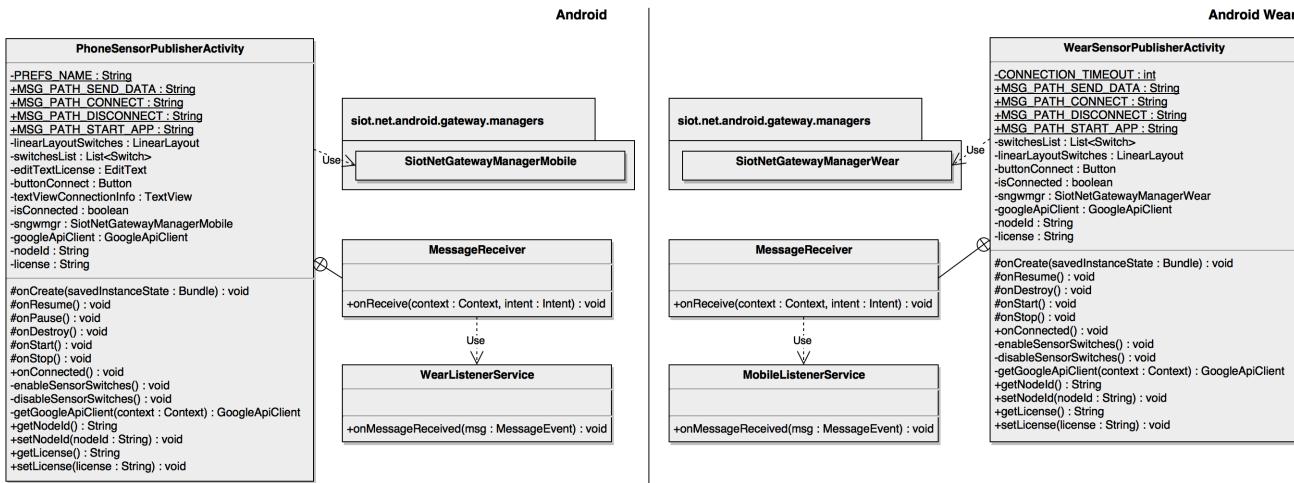


Abbildung 10.2: Klassendiagramme für das Android und Android Wear Sensorcenter

10.3.2 Logik

Diese Applikationen verwenden die siot.net Gateway Library und deren Logik. Erweiterte logische Elemente gibt es für die Handhabung der Verbindung von Smartwatch und Smartphone und das Verhalten der grafischen Oberfläche.

Die Kommunikation wurde wie im Konzept definiert umgesetzt. Wenn eine Smartwatch die Applikation startet, wird per GoogleApiClient ein Kommunikationstunnel zum Smartphone hergestellt. Dabei wird gleich die erste Nachricht via MessageAPI versendet, mit der Aufforderung, das Sensorcenter auf dem Smartphone zu öffnen. Die Klasse WearListenerService empfängt alle Messages welche durch diesen eröffneten Kanal gesendet werden. Auch wenn die App noch nicht läuft kann die Nachricht verarbeitet werden, das ist möglich weil der Listener als Service im Android System registriert wird. Zu erreichen ist dies durch einen Eintrag im AndroidManifest.

Nach dem die Applikation läuft, kann die Verbindung zu siot.net aufgebaut werden. Für die Wear App ist es notwendig, dass die Mobile App auch läuft. Der Aufbau der Kommunikationsleitung erfolgt wie auf Abbildung 9.6 (Smartphone) und Abbildung 9.7 (Smartwatch) illustriert. Die Android Wear Anwendung sendet hier eine CONNECT MessageAPI Nachricht ans Android Gerät. Wenn die Leitung zu siot.net noch nicht steht, wird dieser aufgebaut und eine positive Antwort an die Uhr verschickt. Nach erfolgtem Verbindungsaufbau, werden die Ein- und Ausschalter der Sensoren freigegeben. Es werden nur jene angezeigt, welche relevant sind für das Gerät.

Beim Abmelden verhält es sich ein wenig differenziert. Die Smartwatch meldet sich nur beim Smartphone als Sensorcenter ab. Es geschehen keine weiteren Aktionen. Durch Mobile App die Verbindung zu siot.net instand gehalten. Das Smartphone löst beim Abmelden eine DISCONNECT Nachricht, an die verbundene Smartwatch, aus. Es signalisiert, dass keine Verbindung mehr zum siot.net MQTT Broker besteht. Das Wearable reagiert umgehend und stoppt die Messungen und zeigt an das es nicht mehr verbunden ist.

Messwerte der Uhr können nicht direkt ins Internet verbreitet werden, weshalb das Smartphone diese per MessageAPI als WearableData MessageTyp erhält. Diese können nun mit Hilfe der öffentlichen Methode publishData(topic, Data) der SiotNetGatewayManagerMobile Instanz, an den MQTT Broker publiziert werden.

Aufgrund nicht lange ausdauernden Akkus, sind Android Wear Apps nicht eignen für den Betrieb im Hintergrund. Normalerweise wird die laufende Smartwatch Applikationen beendet, wenn das Display in den Standby-Modus geht. Um bei Verdunklung des Bildschirmes, die Arbeit des Sensorcenters nicht zu unterbrechen, muss das Gerät eines der folgenden Funktionen unterstützen: Always-On Screen oder den Ambient Mode. Erstere lässt die Uhr nie in den Ruhezustand kehren. Das Zweite lässt die Applikation laufen, verfinstert den Touchscreen um den Akku zu schonen. Um beide Methoden zu aktivieren, muss dies in der WearSensorPublisherActivity implementiert werden.

10.3.3 Graphical User Interface

Das GUI beider Softwareartefakte ist sehr simpel und geradlinig gehalten, da es hauptsächlich zu Demonstrationszwecke dienen soll.

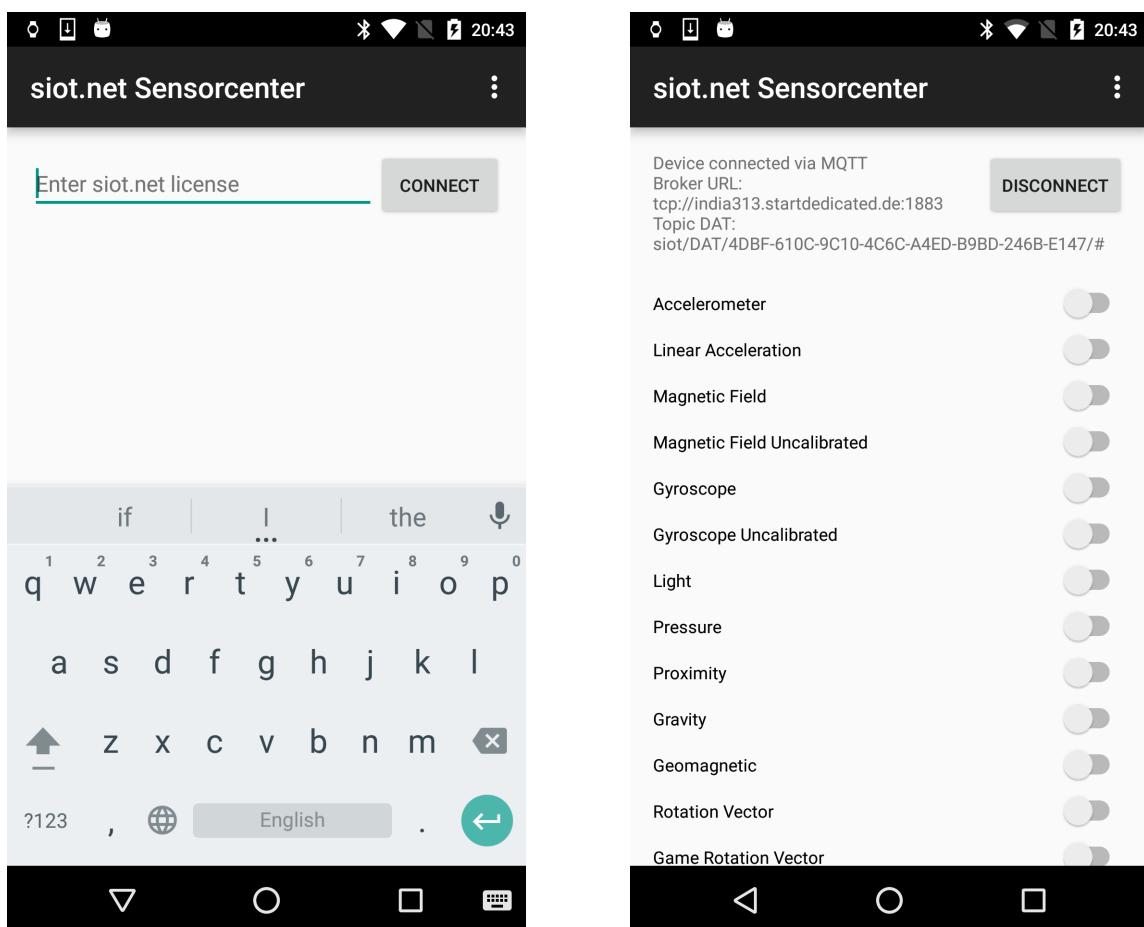


Abbildung 10.3: Die möglichen Ansichten des Sensorcenter eines Smartphones

Die Android Mobile App ist eine einfach zweisegmentige Ansicht. Oben mit dem Texteingabe oder Informationsbereich und dem Knopf. Die Abbildung 10.3 zeigt, dass das untere Teilstück die Schalter für die Sensoren darstellt, wenn der Benutzer eingeloggt ist.



Abbildung 10.4: GUI Elemente des Smartwatch Sensorcenters

Das GUI der Smartwatch Anwendung ist noch minimalistischer, dies visualisiert die Abbildung 10.4. Der Touchscreen der Uhr ist eher klein, infolge dessen werden nur die zwingendsten Informationen angezeigt.

10.4 Plattformen und Libraries

Für die Entwicklung der Android und Android Wear Bibliothek und App, wurde ausschliesslich Android Studio verwendet. Die Abbildung 10.5 zeigt, dass die Entwicklungsumgebung einige Abhängigkeiten hat. Für die eingesetzten Sprachen JAVA und XML genutzte Android Studio (A) basiert auf der IntelliJ IDEA Plattform. Es bietet Gradle als flexibles Build-Management-Automatisierungs-Tool. Dies nutzt eine Groovy(B) basierende domänen spezifische Sprache (DSL) zur Beschreibung der zu erstellenden Artefakte. Die benutzten Bibliotheken werden in dieser Beschreibungsdatei erfasst. Libraries können aus einem Lokalen Verzeichnis(D) kommen oder aus einer globalen Ablage, z.B. jCentral⁴(C).

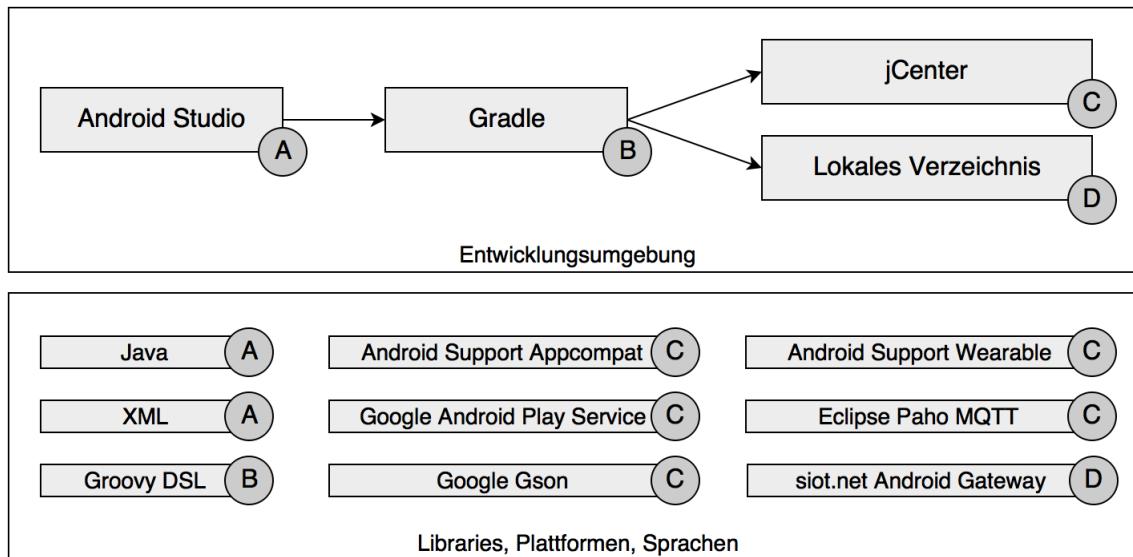


Abbildung 10.5: Genutzte technische Elemente und deren Abhängigkeiten

⁴vgl. <https://bintray.com/bintray/jcenter>

11 Schlussfolgerung

12 Ausblick

A. Abbildungsverzeichnis

2.1	MQTT Architektur	4
2.2	siot.net Logo	5
3.1	Selbstfahrendes Auto Mercedes-Benz F015 In Motion	7
3.2	Fussgängererkennung des F015	7
3.3	Solar Roadway, dynamische Strasse	8
3.4	Frühwarnsystem von Erdbeben	10
3.5	Smartwatch Ziffernblätter	11
3.6	Smartwatch Anzeige von Smartphone	12
3.7	Smartwatch Anzeige von Tätigkeiten	13
3.8	Smartwatch und Landkarten	14
4.1	Mobile Zahlungslösungen: Apple Pay, Google Wallet und TWINT powered by PostFinance	17
4.2	Smartphone gesteuerte Drohne: Parrot bebop	18
7.1	Übersicht Android Wear Paket Architektur	26
7.2	Übersicht siot.net Applikationsarchitektur	26
7.3	Geplante Netzwerk-Architektur mit BLE/ohne WLAN	27
7.4	Geplante Netzwerk-Architektur ohne BLE/mit WLAN	28
7.5	Effektive Netzwerk-Architektur mit BLE/ohne WLAN	29
7.6	Effektive Netzwerk-Architektur mit BLE/ohne WLAN	30
8.1	Use Case siot.net Gateway Library	31
8.2	Use Case siot.net Sensorcenter	38
8.3	Use Case siot.net Sensorcenter	43
9.1	siot.net Gateway Library Packagediagramm	48
9.2	siot.net Gateway Library Domänendiagramm	49
9.3	siot.net Gateway Library Sequenzdiagramm App starten 1	50
9.4	siot.net Gateway Library Sequenzdiagramm App starten 2	50
9.5	siot.net Gateway Library Sequenzdiagramm App starten 3	51
9.6	siot.net Gateway Library Sequenzdiagramm Login 1	51
9.7	siot.net Gateway Library Sequenzdiagramm Login 2	52
9.8	siot.net Sensorcenter Domänendiagramm	52
9.9	siot.net Sensorcenter Sequenzdiagramm Sensor starten 1	53
9.10	siot.net Sensorcenter Sequenzdiagramm Sensor starten 2	54
10.1	siot.net Gateway Library Klassendiagramm	56
10.2	siot.net Sensorcenter Klassendiagramm	58
10.3	siot.net Sensorcenter Smartphone Screens	59
10.4	siot.net Sensorcenter Smartwatch Screens	60
10.5	Plattforme, Libraries	60

B. Tabellenverzeichnis

6.1 Technische Daten von Smartwatches - Teil 1	24
6.2 Technische Daten von Smartwatches - Teil 2	25
8.1 siot.net Gateway Library: Übersicht Anwendungsfall #1	32
8.2 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #1	32
8.3 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #1	32
8.4 siot.net Gateway Library: Übersicht Anwendungsfall #2	33
8.5 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #2	33
8.6 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #2	33
8.7 siot.net Gateway Library: Übersicht Anwendungsfall #3	34
8.8 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #3	34
8.9 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #3	34
8.10 siot.net Gateway Library: Übersicht Anwendungsfall #4	35
8.11 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #4	35
8.12 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #4	35
8.13 siot.net Gateway Library: Übersicht Anwendungsfall #5	36
8.14 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #5	36
8.15 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #5	36
8.16 siot.net Gateway Library: Übersicht Anwendungsfall #6	37
8.17 siot.net Gateway Library: Chronologischer Ablauf von Anwendungsfall #6	37
8.18 siot.net Gateway Library: Ausnahmen und Varianten von Anwendungsfall #6	37
8.19 siot.net Gateway Library: Nicht-funktionale Produktanforderungen	37
8.20 siot.net Sensorcenter: Übersicht Anwendungsfall #1.1 und #1.2	39
8.21 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #1.1	39
8.22 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #1.2	39
8.23 siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #1.1 und #1.2	40
8.24 siot.net Sensorcenter: Übersicht Anwendungsfall #2.1 und #2.2	40
8.25 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #2.1 und #2.2	40
8.26 siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #2.1 und #2.2	40
8.27 siot.net Sensorcenter: Übersicht Anwendungsfall #3.1 und #3.2	41
8.28 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #3.1 und #3.2	41
8.29 siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #3.1 und #3.2	41
8.30 siot.net Sensorcenter: Übersicht Anwendungsfall #4.1 und #4.2	41
8.31 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #4.1	41
8.32 siot.net Sensorcenter: Chronologischer Ablauf von Anwendungsfall #4.2	42
8.33 siot.net Sensorcenter: Ausnahmen und Varianten von Anwendungsfall #4.1 und #4.2	42
8.34 siot.net Sensorcenter: Nicht-funktionale Produktanforderungen	42
8.35 siot.net Dashboard: Übersicht Anwendungsfall #2	44
8.36 siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #2	44
8.37 siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #2	44
8.38 siot.net Dashboard: Übersicht Anwendungsfall #3	44
8.39 siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #3	45
8.40 siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #3	45
8.41 siot.net Dashboard: Übersicht Anwendungsfall #4	45
8.42 siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #4	45
8.43 siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #4	45
8.44 siot.net Dashboard: Übersicht Anwendungsfall #5	46

8.45 siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #5	46
8.46 siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #5	46
8.47 siot.net Dashboard: Übersicht Anwendungsfall #6	46
8.48 siot.net Dashboard: Chronologischer Ablauf von Anwendungsfall #6	46
8.49 siot.net Dashboard: Ausnahmen und Varianten von Anwendungsfall #6	46
8.50 siot.net Dashboard: Nicht-funktionale Produktanforderungen	47
9.1 siot.net Gateway Library: Auflistung der Packageinhalte	48