Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 355

Eine sichere Schnittstelle zu Wearable Computers

Nicole Krawietzek

Studiengang:	Informatik
Prüfer/in:	PD Dr. Holger Schwarz
Betreuer/in:	Dipl.-Inf. Christoph Stach, Dipl.-Inf. Frank Steimle
Beginn am:	18. Juli 2016
Beendet am:	31. Januar 2017
CR-Nummer:	J.3 K4.1

Kurzfassung

Die sogenannten Wearable Computers befinden sich in einer Wachstumsphase. Zu ihnen zählen Smartwatches, Smartrings, aber auch diverse medizinische Geräte. Sie besitzen unterschiedliche Sensoren, wie zum Beispiel GPS, Beschleunigungs-, Licht- und Herzschlagsensoren. Sowohl im privaten als auch im medizinischen Bereich werden Wearables in immer größerem Umfang zur Datenerfassung genutzt. Diese Daten können mit Hilfe verschiedener Applikationen ausgewertet werden. Da es sich dabei um sensible Daten handeln kann müssen diese geschützt gespeichert und übertragen werden. Nutzer von Wearables sollen selbst entscheiden können, welchen Applikationen sie den Zugriff auf ihre Daten genehmigen und dadurch ihre Privatsphäre wahren.

Wearables können über unterschiedliche Technologien mit Smart Devices verbunden werden. Für Entwickler von Applikationen stellt die große Auswahl an Verbindungsstandards eine Schwierigkeit in der Flexibilität ihrer Applikationen dar. Einige Technologien werden in dieser Arbeit vorgestellt. Außerdem werden Kriterien definiert, die Verbindungsstandards erfüllen sollen.

In dieser Arbeit wird ein neues Konzept für eine sichere Schnittstelle zwischen Smartphones und Wearables entwickelt. Die Umsetzung dieses Konzepts und eine mögliche Implementierung anhand der Beschreibung eines möglichen Prototyps wird vorgestellt. Zum Vergleich der vorhandenen Schnittstellen und des neuen Konzepts wird eine Evaluation anhand der vorher definierten Kriterien durchgeführt, in der gezeigt wird, dass das neue Konzept die nötigen Anforderungen erfüllt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ACL Asynchronous Connectionless. 20

AIDL Android Interface Definition Language. 46

APK Android Package. 28

ATT Attribute Profile. 24

BLOB Binary Large Object. 29

GAP Generic Access Profile. 17

GATT Generic Attribute Profile. 17

HDP Health Device Profile. 21

ISM Industrial, Scientific and Medical Band. 19

LE Low Energy. 17

PMP Privacy Management Platform. 15

RPC Remote Procedure Call. 29

SCO Synchronous Connection-Oriented. 20

SDC Secure Data Container. 15

SDP Session Description Protocol. 20

SIG Special Interest Group. 19

SPP Serial Port Profile. 21

UART Universal Asynchronous Receiver Transmitter. 26

1. Einleitung

Schon Mark Weiser beschrieb 1991 in seiner Vision vom Computer des 21. Jahrhunderts allgegenwärtige Computer, die sich zu unterschiedlichsten Zwecken an allen möglichen Orten nutzen lassen [Wei91]. Diese Vision wurde durch die Einführung der Smartphones, die es erlauben überall und zu jeder Zeit Computer zu nutzen, verwirklicht.

Die sogenannten Wearable Computers, also Geräte, die „getragen“ werden können, wie beispielsweise Smartwatches oder Smartglasses, befinden sich in einer Wachstumsphase [ODo15]. In ihnen sind unterschiedliche Sensoren verbaut, wie zum Beispiel GPS, Beschleunigungs-, Licht- und Herzschlagsensoren. Durch diese können sie zu unterschiedlichen Zwecken eingesetzt werden. Einerseits können sie zur Erweiterung des Smartphones, indem Nachrichten und Anrufe auf die Smartwatches weitergeleitet und so möglicherweise nicht mehr direkt das Smartphone aus der Tasche gesucht werden muss, genutzt werden. Andererseits können sie aber auch zu neuen Zwecken, wie beispielsweise im Fitness- und Gesundheitsbereich, verwendet werden. Für diese Zwecke werden ständig neue Applikationen für Smartphones entwickelt, die immer mehr Daten erfassen [AVC15].

Die Applikation Google Fit [Goo16b] beispielsweise erfasst verschiedene fitnessrelevante Daten, wie die Geh- bzw. Laufgeschwindigkeit, GPS-Daten und erkennt mit Hilfe der Sensoren sportliche Aktivitäten. Außerdem führt sie auch die Daten anderer Apps zusammen und verknüpft diese. So wird ein Gesamtbild aus Ess- und Sportgewohnheiten eines Menschen erfasst.

Doch je mehr Funktionen von diesen Geräten übernommen werden, desto größer ist die Gefahr von Missbrauch. Vor allem im medizinischen Bereich kann ein Missbrauch schwerwiegende Folgen haben. Auch bei medizinischen Geräten kann es sich um Wearables handeln, wenn diese einen Computer besitzen und am Körper getragen werden [Wea16]. So können zum Beispiel Hörgeräte, Herzschrittmacher und Insulinpumpen zu Wearables gehören, wenn sie weiterführende Funktionen besitzen [Ärz14]. Im Januar 2017 wurde in Herzschrittmachern mit eingebauten Defibrillatoren die Existenz einer Sicherheitslücke veröffentlicht [US 17], bei der über Radiofrequenzen auf diese Geräte zugegriffen und so der Rhythmus des Herzschlags verändert werden konnte. Auch im Jahr 2012 zeigte der Hacker Barnaby Jack, dass er sich drahtlos in eine Insulinpumpe einhacken und die gesamte, sich in der Insulinpumpe befindliche, Insulindosis auf einmal injizieren konnte. Hätte er dies statt an einer Puppe bei einer lebenden Person gemacht, wäre diese an der verabreichten Dosis gestorben [Par12].

Jedoch können Wearables, wenn sie immer mer gesundheitsrelevante Funktionen übernehmen den Nutzern helfen gesundheitliche Probleme frühzeitig zu erkennen und gegebenfalls dadurch zu einer Reduzierung der Behandlungskosten führen [MVV+14].

Bei den meisten abgesetzten Wearables handelt es sich um Smartwatches oder Fitnessarmbänder [Bra15]. Diese besitzen keine Funktionalitäten, die medizinisch gefährlich für den Nutzer werden könnten. Jedoch stellen auch sie ein Ziel von Hackern dar [Sop13]. Denn auch sie sammeln Daten, die gesundheitsrelevant sein können. Dabei sind sich Nutzer von Smartphones bewusst, dass sensible und wertvolle Daten auf ihrem Gerät gespeichert sind [MBK+12]. Denn wie schon erwähnt sammeln viele Applikationen unterschiedliche Daten und tauschen sie untereinander aus, so dass sie zusammen an einer Stelle für den Nutzer verfügbar gemacht werden. Es wäre vorteilhafter, wenn viele einzelne Applikationen Daten von einer zentralen Stelle zur Verfügung gestellt bekommen und nicht einzeln alle Daten erfassen müssen. Ausserdem soll der Nutzer selbst entscheiden können, welche Applikation welche Daten überhaupt verwenden kann. So müssten sich Entwickler von Applikationen nicht mehr mit der Datenerfassung ausseinander setzen, sondern könnten sich auf andere Funktionalitäten und die Datenauswertung konzentrieren, während es für den Nutzer übersichtlicher wird, welche Daten überhaupt erfasst und mit welchen Applikationen diese geteilt werden.

Mit einem Standard, der sich flexibel auf viele Arten von Geräten erweitern lässt und an keinen Technologiehersteller gebunden ist, kann so ein positiver Nutzen für die Entwickler verschiedener Geräte und auch für deren Nutzer entstehen. Die Entwickler müssen sich nicht in verschiedene Technologien einarbeiten und können auf schon vorhandene Schnittstellen zurückgreifen, während die Nutzer Wearables unterschiedlicher Hersteller nutzen können und völlige Flexibilität genießen.

1.1. Motivation

In dieser Arbeit soll eine sichere Schnittstelle zwischen Wearables und Smartphones entwickelt werden. Dazu werden bisher vorhandene Technologien und Schnittstellen analysiert und erklärt, da bisher kein einheitlicher Standard vorhanden ist. Die neue Schnittstelle soll einen Standard festlegen, der von allen Geräteherstellern, Entwicklern und Anwendern genutzt werden kann. Es sollen alle Arten von Wearables unterstützt werden: von Sensoren, die lediglich Daten aufzeichnen und weiterleiten bis hin zu Wearables mit Betriebssystemen, wie Smartwatches. Die Schnittstelle soll Sicherheit für den Datenaustausch bieten. Es sollen keine unverschlüsselten Daten auf Wearables oder Smartphones gespeichert werden und somit niemandem unbefugten Zugang zu diesen gewähren. Nur der Nutzer des Smartphones und Wearables soll definieren können, welche Applikationen Zugriff auf welchen Teil der gespeicherten Daten erhalten. Dazu soll bei Wearables, auf denen bisher viele verschiedene Applikationen Daten gesammelt haben, eine zentrale Applikation die Daten sammeln und an das Smartphone schicken. Dort kann dann der Nutzer über die Weitergabe dieser Daten selbst

entscheiden. Auch soll er sich sicher sein können, dass die Daten, die übermittelt werden, plausibel sind. Diese Anforderungen werden in einem Kriterienkatalog definiert.

Im Anschluss daran wird ein neues Konzept vorgestellt und erläutert. Die technische Umsetzbarkeit des neuen Konzepts wird durch die Implementierung eines Prototyps gezeigt. Nachfolgend wird das neue Konzept im Vergleich mit den vorher analysierten Schnittstellen und Technologien bezüglich der geforderten Kriterien evaluiert.

Für die Umsetzung der neuen Schnittstelle wird die Privacy Management Platform (PMP) [SM13] genutzt. Diese liefert ein Berechtigungssystem anhand dessen der Nutzer über die Weitergabe von Daten an Applikationen und den Zugriff dieser auf Funktionen entscheiden kann. Dazu registrieren sich Applikationen bei der PMP und geben an auf welche Funktionen sie zugreifen möchten. Über sogenannte Privacy Settings kann der Nutzer entscheiden welche Funktionalitäten für die jeweilige Applikation freigegeben werden. Der Aufruf der Funktionen und Daten wird dabei über Ressourcen realisiert. Diese sind frei verfügbar und können von verschiedenen Applikationen wiederverwendet werden. Darüber hinaus liefert die PMP in Verbindung mit dem Secure Data Container (SDC) [SM15] eine Möglichkeit die übertragenen Daten verschlüsselt zu speichern und zu übertragen.

1.2. Gliederung der Arbeit

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: In diesem Kapitel werden Wearables und aktuelle Verbindungstechnologien erläutert. Dabei handelt es sich um Bluetooth, Bluetooth LE mit GATT und GAP, das HDP, Terminal I/O und Android Wear. Dieses Kapitel soll einen Überblick über diese Technologien und deren Funktionsweise schaffen.

Kapitel 3 – Kriterienkatalog für Verbindungsstandards: Hier werden Kriterien definiert, die von Verbindungsstandards erfüllt werden sollen.

Kapitel 4 – Konzept: Dieses Kapitel dient dazu ein neues Konzept für Verbindungen zwischen Wearables und Smartphones einzuführen. Es wird darauf eingegangen, wie derzeit Applikationen mit Smartphones interagieren und welche Funktionen mehr Sicherheit in eine Schnittstelle integrieren können.

Kapitel 5 – Umsetzung: In diesem Kapitel wird erklärt wie das zuvor erstellte Konzept umgesetzt werden kann. Es wird die PMP (Privacy Management Platform) vorgestellt und wie anhand von Ressourcen und des SDCs das zuvor vorgestellte Konzept umgesetzt werden kann.

Kapitel 6 – Implementierung: Ein Beispiel einer Implementierung zeigt, wie das Konzept implementiert werden kann und wie die Schnittstelle mit einer Applikation interagiert.

Kapitel 7 – Evaluation: In diesem Kapitel werden die zuvor beschriebenen Technologien und der PMP-basierte Ansatz anhand der zuvor definierten Kriterien evaluiert und verglichen.

Kapitel 8 – Zusammenfassung und Ausblick Dieses Kapitel fasst die Ergebnisse der Arbeit zusammen und erläutert welche zukünftigen Herausforderungen und Anknüpfungspunkte existieren.

2. Grundlagen

Den Einstieg in dieses Kapitel bildet eine Definition von Wearables, sowie eine Erklärung der Funktionalitäten und deren heutiger Nutzung. Anschließend wird, durch die Beschreibung von Bluetooth, die technische Seite einer Verbindung zwischen einem Wearable und einem Smart Device erläutert. Dazu wird das in der Medizin genutzte Bluetooth Health Device Profile erklärt, welches zum Zugriff und zur Verknüpfung von Daten medizinischer Wearables genutzt wird. Außerdem wird auf den heutigen Bluetooth Standard mit Bluetooth Low Energy (LE) und dessen Profilen Generic Access Profile (GAP) und Generic Attribute Profile (GATT) zum Verbindungsaufbau und zur Datenübertragung eingegangen. Als Weiterentwicklung des GATT Profils wird im Anschluss das Terminal I/O Profil erklärt, das neue Funktionalitäten in das Profil integriert. Zuletzt wird das Betriebssystem Android Wear erläutert, das für Wearables entwickelt wurde und eine Verbindung zu Android Smartphones bietet, indem es auf dem Bluetooth LE Standard aufbaut.

2.1. Überblick Wearables

Als Wearables bezeichnet man elektronische Geräte, die in Kleidung oder Accessoires integriert sind oder auch als Definition im weiteren Sinne handelt es sich bei Wearables um Microcomputer, die eine Funktion im Lifestyle-, Gesundheits- und Fitnessbereich haben [Wea16]. Eine andere Definition lautet, dass es sich um tragbare Technologien handelt, die einen bequemen und meist freihändigen Zugang zu Elektronik und Computern ermöglichen [TM14]. Sie verfügen im Vergleich zu herkömmlichen Computern (Notebooks, Desktop-PCs) meist über weniger Rechenleistung und können sowohl unabhängig von anderen elektronischen Geräten als auch in Abhängigkeit von diesen genutzt werden. Beispielsweise können Smartwatches mit dem Betriebssystem Android Wear nur in Verbindung mit einem Smartphone mit einem Android Betriebssystem genutzt werden, während Google Glasses auch eigenständig, ohne zusätzliche Endgeräte, genutzt werden können. Beispiele für Wearables sind Brillen, Uhren, sogenannte Smartwatches, Ringe und Armbänder, die durch diverse Sensoren und andere technische Eigenschaften verschiedene Aufgaben übernehmen können. So können Fitness-Armbänder durch Bewegungssensoren Bewegungen aufzeichnen und so dem Nutzer einen detaillierten Überblick über getätigte Bewegungen schaffen. Doch nicht nur Geräte zum reinen Freizeitvergnügen werden als Wearables bezeichnet, auch bei medizinischen Geräten kann es sich um Wearables handeln. Beispiele hierfür sind Insulinpumpen, die über Computer angesteuert werden können oder auch Herzschrittmacher, die Daten aufzeichnen und versenden können und ebenfalls über

(a) Samsung Gear S3 Smartwatch [Sam16]

(b) Smartring [Mar14]

Abbildung 2.1.: Zwei Beispiele für Wearables

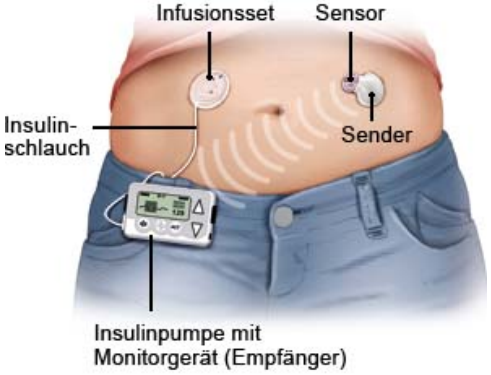

Abbildung 2.2.: Insulinpumpe [Ins15]

Computer angesteuert werden können. Laut einer von TNS Infratest und des Bundesverbands der Hörgeräte-Industrie (BVHI) durchgeführten Umfrage zählen 50 Prozent der Befragten auch Hörsysteme zu Wearables [Ärz14]. In Abbildung 2.1 und 2.2 sind Beispiele für Wearables in Form von einer Smartwatch, eines Smartrings und als medizinisches Wearable einer Insulinpumpe abgebildet.

Der Markt für Wearables befindet sich derzeit im Wachstum [Bra15]. Wie in Abbildung 2.3 erkennbar ist, soll der Absatz von Wearables im Jahr 2019 im Vergleich zum Jahr 2016 um 50 Prozent wachsen. Bei den meisten abgesetzten Wearables handelt es sich um Uhren oder Armbänder. Smartwatches gibt es in unterschiedlichen Varianten. Es gibt Modelle, die keinen Bildschirm besitzen, sondern lediglich ein Ziffernblatt und Zeiger. Über Vibrationen oder Beleuchtung können diese Uhren über auf dem Smartphone eingehende Benachrichtigungen und Anrufe informieren und als Wecker fungieren. Andere Modelle besitzen darüber hinaus Sensoren, mit denen sie Aktivitäten, wie z.B. Schritte, aufzeichnen und diese an das Smartphone weiterleiten können. Mit Hilfe von Applikationen können diese dann durch den Nutzer

Abbildung 2.3.: Prognose des weltweiten Wearable Markts [Bra15]

abgerufen und analysiert werden. Die umfangreichsten Smartwatches besitzen kein Zifferblatt, sondern einen Bildschirm und stellen kleine Computer mit eigenem Betriebssystem am Handgelenk dar. Für diese Smartwatches werden eigene Applikationen angeboten, die das Aussehen der Uhranzeige verändern können. Bei Android werden diese Anzeigen Watchfaces genannt. Außerdem können über diese Uhren auch Applikationen auf dem Handy gesteuert werden, sie reagieren auf Sprachbefehle und sind individuell anpassbar [Sma16b].

2.2. Bluetooth

Bei Bluetooth handelt es sich um eine Funktechnik zum Übertragen von Daten und Sprache. Diese ist für Kurzstrecken geeignet und dadurch, dass sie lizenzfrei und standardisiert ist, heute weit verbreitet.

Die Mobilfunkhersteller Ericsson und Nokia entwickelten 1994 die Bluetooth Technik, mithilfe derer kleine Geräte ohne großen Aufwand miteinander verbunden werden können. In den Bluetooth Core Specifications [Blu16b] wird Bluetooth durch die Bluetooth Special Interest Group (SIG), einen Zusammenschluss mehrerer tausend Firmen, standardisiert und spezifiziert. Die erste Spezifikation für Bluetooth 1.0 und 1.0B wurde 1999 erstellt. Die aktuellste Bluetooth Spezifikation beinhaltet die Version 5, die im Dezember 2016 offiziell verabschiedet wurde, deren Fokus auf schneller Geschwindigkeit und größerer Reichweite liegt [Blu16a].

Bluetoothgeräte senden auf einem lizenzfreien Industrial, Scientific and Medical Band (ISM)-Band zwischen 2,402 und 2,490 GHz und erreichen dabei Datenübertragungsraten von etwa 1 MBit pro Sekunde. Durch kleine Datenpakete und häufige Frequenzsprünge ist die Verbindungsqualität bei der Bluetoothtechnologie sehr hoch. Mit Bluetooth können bis zu acht Geräte gleichzeitig aktiv miteinander vernetzt sein, während weitere 248 Geräte angemeldet sein

können, aber passiv bleiben müssen. Für die Kommunikation ist hierbei kein Sichtkontakt notwendig, jedoch ist die Reichweite auf wenige Meter begrenzt.

Eine Verbindung von mindestens zwei Endgeräten wird als Piconet bezeichnet. Dabei handelt es sich um ein Personal Area Network, bei dem ein Gerät als „Master“ und das bzw. die anderen Geräte als „Slave“ agieren. Das Gerät, welches als Erstes ein Datenpaket verschickt und somit eine Datenverbindung herstellt wird dabei als „Master“ bezeichnet. Der Verbindungsaufbau beinhaltet drei Phasen [kio14]. Diese sind in Abbildung 2.4 aufgezeigt und werden im Folgenden erläutert. In der ersten Phase „Inquire“ (engl. für Anfrage) sendet das Mastergerät eine Anfrage an alle verfügbaren Geräte, die im aktuellen Netzwerk liegen. Die lauschenden Geräte, die diese Anfrage erhalten, antworten mit ihrer Adresse und weiteren Angaben, wie zum Beispiel ihrem Namen. Falls die Adresse der nötigen Geräte schon bekannt ist wird diese Phase übersprungen. In der zweiten Phase „Paging (Connecting)“ wird eine Verbindung zwischen zwei Bluetooth Geräten hergestellt. Hierbei werden die Uhren und die Frequenz mit dem Knotenpunkt der verbundenen Geräte synchronisiert. Danach besteht eine Verbindung der zwei Bluetoothgeräte und das Mastergerät kann über das Session Description Protocol (SDP) die Dienste des Knotenpunkts entdecken. Sobald dies erfolgte besteht ein Kommunikationskanal zwischen den zwei Bluetoothgeräten und die Geräte befinden sich in der dritten Phase „Connected“. Die Bluetoothgeräte können sich, während der Kommunikationskanal aktiv ist, in vier verschiedenen Zuständen befinden:

Active Mode: Das Gerät ist aktiv und erhält oder sendet Daten.

Sniff Mode: Das Gerät befindet sich in einem stromsparenden Modus und lauscht in definierten Abständen auf Übertragungen.

Hold Mode: Das Gerät befindet sich in einem stromsparenden Modus und wechselt zurück in den Active Mode, sobald eine vorher definierte Zeitspanne verstrichen ist. Das Mastergerät kann ein anderes Gerät dazu anweisen in diesem Modus zu wechseln.

Park Mode: Das Mastergerät kann ein anderes Gerät dazu anweisen zu „parken“ und somit inaktiv zu sein bis es wieder von ihm aufgeweckt wird.

Die Verbindung findet auf zwei verschiedene Arten statt. Dies ist ein großer Vorteil gegenüber anderen Funkstandards, die meist nur eine dieser Arten unterstützen. Mit dem Synchronous Connection-Oriented (SCO) Verfahren wird Sprache in festen, von der Masterstation reservierten Zeitschlitzen in Echtzeit übertragen. Im Vordergrund steht hier die Qualität der Verbindung, weshalb ein geschützter Kanal für die Sprachübertragung genutzt wird [ITW17b]. Neben der synchronen Übertragung für Sprache wird für die Datenübertragung ein asynchrones Verfahren, das Asynchronous Connectionless (ACL), genutzt. Bei diesem werden durch Fehlerkorrekturen und Datenpaket-Wiederholungen fehlerfreie Übertragungen gewährleistet [ITW17a].

Die Sicherheit des Verbindungsaufbaus wird bei Bluetooth über drei Sicherheitsstufen definiert [Sch03]:

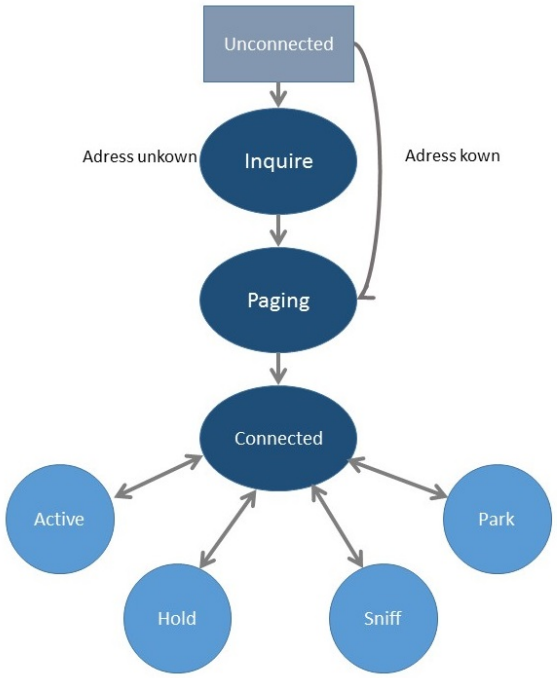

Abbildung 2.4.: Bluetooth Verbindungsaufbau

Niedrig
: Die Geräte können sich untereinander erkennen und ohne Authentifizierung Verbindungen zueinander herstellen.

Mittel
: Die Geräte können sich untereinander erkennen, können aber ohne Authentifizierung keine Verbindung zueinander herstellen (Dienste-Authentifizierung).

Hoch
: Die Geräte können sich untereinander nicht erkennen. Eine Verbindung ist nur mit einer Authentifizierung möglich (Verbindungs-Authentifizierung).

Eine zentrale Rolle spielen in der Bluetoothtechnologie Profile. Ein Ausschnitt der verfügbaren Profile befindet sich in Tabelle 2.1. In Profilen werden Regeln und Protokolle definiert, die zur Kommunikation zwischen Bluetoothgeräten nötig sind. Dadurch können Geräte verschiedener Hersteller direkt miteinander kommunizieren. Durch die Integration diverser Protokolle in verschiedene Geräte können diese an ihren jeweiligen Funktionsumfang angepasst werden und müssen nicht alle Funktionalitäten besitzen. Die Profile, die den Geräteherstellern zur Verfügung stehen, werden vom Bluetooth SIG standardisiert. So stellt das GAP ein standardisiertes Profil zum Verbindungsaufbau dar, das auch von weiterführenden Technologien genutzt wird.

Kürzel	Profil	Anwendung
GAP	Generic Access Profile	grundlegendes Verfahren zur Authentifizierung und Verbindungsaufnahme
A2DP	Advanced Audio Distribution Profile	drahtlose Stereoverbindung für Lautsprecher oder Kopfhörer
SDAP	Service Discovery Application Profile	Diensteabfrage, der gerade sichtbaren Nachbarn
HCRP	Hardcopy Cable Replacement Profile	Drucken
HFP	Hands Free Profile	herstellerunabhängige Kommunikation zwischen Handy und Freisprecheinrichtung
OPP	Object Push Profile	Termine und Adressen Übertragen
HDP	Health Device Profile	Übertragung von Medizindaten

Tabelle 2.1.: Ausschnitt der Übersicht über Bluetooth Profile [Sch03]

2.2.1. Bluetooth Health Device Profile

Mit dem Bluetooth Health Device Profile (HDP) [Blu09] wird die Kommunikation und Interoperabilität zwischen medizinischen, gesundheitlich relevanten und Fitnessgeräten ermöglicht und aufrecht erhalten. Es wurde 2008 entwickelt und ersetzte das bisher in der Medizin genutzte Serial Port Profile (SPP). Bei der Nutzung des SPP war es möglich, dass Geräte eines Herstellers nicht mit Geräten eines anderen Herstellers zusammenarbeiten konnten. Diese Interoperabilität sollte durch das HDP ermöglicht werden und wurde, unter anderem, durch das *IEEE 11073 Data Exchange Protocol* (vgl. Abbildung 2.6) erreicht. In diesem sind Spezifikationen zu Datenübertragungen im gesundheitlichen Bereich hinterlegt. HDP ist seit Bluetooth 3.0 möglich und führt ein neues Protokoll, das Multi-Channel Adaptation Protocol (MCAP), in den Bluetooth-Standard ein. Dieses ermöglicht es mehrere simultane Datenkanäle zur Verfügung zu stellen, wie man im unteren Teil der Abbildung 2.5 erkennen kann. Von HDP werden sowohl strombasierte Anwendungen, wie beispielsweise EKG Geräte, als auch nicht-strombasierte Anwendungen, z.B. Blutdruckmessgeräte unterstützt. Bei strombasierten Anwendungen ist nicht nur das reine Messergebnis sondern auch der Zeitstempel eine wichtige Angabe, die zur Auswertung der Daten benötigt wird. Das HDP erleichterte es die Daten verschiedener Geräte zusammen zu führen und gemeinsam auswerten zu können. So können Abhängigkeiten zwischen den Daten besser erkannt werden. Auch kann der Nutzer selbst einstellen, ob Daten kontinuierlich oder zu bestimmten definierten Zeitpunkten übertragen werden sollen. Einzelne Sensoren ausschließen und so Geräte mit verschiedenen Sensoren teilweise zu beschränken kann der Nutzer jedoch nicht. Im HDP Profil werden zwei Rollen definiert: die HDP *sink* und die HDP *source*. Beim HDP *sink* handelt es sich um das Gerät, das Daten von einem oder mehreren medizinischen Geräten erhalten und weiterverarbeiten bzw. an höhergestellte Systeme (bspw. eine Cloud) verteilen soll. Die HDP *source* ist das Gerät, das die Messungen durchführt und

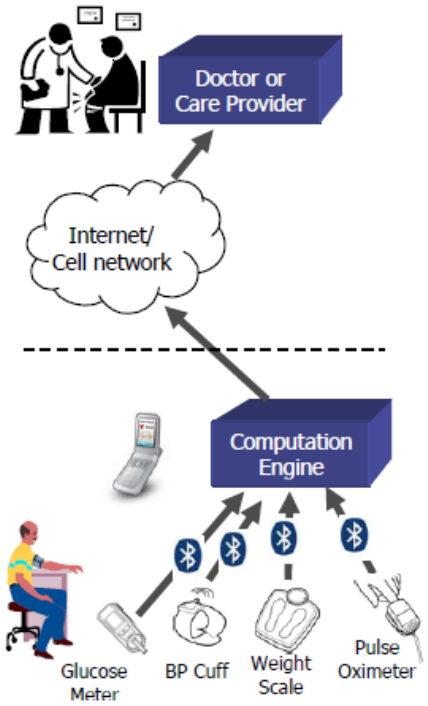

Abbildung 2.5.: Beispielhafte Nutzung des HDP Profils [Blu09]

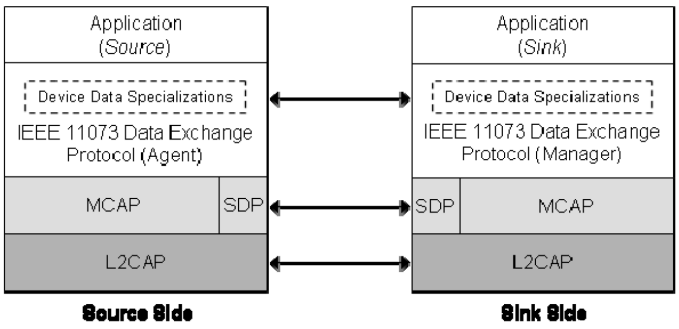

Abbildung 2.6.: Aufbau des HDP Profils [Blu09]

die Ergebnisse über eine Bluetoothverbindung an die HDP *sink* sendet [Blu09]. Auf Seiten des *sinks* findet keine Kontrolle der Plausibilität der Daten, sowie keine Verschlüsselung statt.

2.2.2. Bluetooth LE

Bei Bluetooth LE handelt es sich um die Versionen 4.0, 4.1 und 4.2, die auch als Bluetooth Smart bezeichnet werden. Bluetooth LE wurde speziell für Geräte mit geringer Akkuleistung

entwickelt. Dies wird durch Datenübertragung in größeren Intervallen mit geringen Datenmengen ermöglicht. Für die Übertragung großer Datenmengen wird weiterhin das gewöhnliche Bluetooth empfohlen. Die Reichweite beträgt bei Bluetooth LE etwa 10 Meter. Bluetooth Geräte werden in zwei Kategorien eingeteilt: Bluetooth Smart und Bluetooth Smart Ready. Bei Bluetooth Smart Geräten handelt es sich um Geräte, die lediglich Bluetooth LE unterstützen, wie es bei den in Kapitel 2.1 erläuterten Wearables der Fall ist. Zwei Bluetooth Smart Geräte können untereinander keine direkte Verbindung über Bluetooth herstellen, lediglich mit einem Bluetooth Smart Ready Gerät. Bei Bluetooth Smart Ready Geräten handelt es sich um Geräte, die sowohl Bluetooth LE als auch herkömmliches Bluetooth unterstützen. Beispielsweise fallen Smartphones in diese zweite Kategorie [Sch03]. Im Folgenden werden zwei Profile von Bluetooth LE näher erläutert, die für den Verbindungsaufbau zuständig sind und eine zentrale Rolle bei der Datenübermittlung spielen.

GATT und GAP

Das GAP definiert das Erkennen zwischen Bluetoothgeräten und das Verbindungsmanagement. Auch wenn Bluetoothgeräte keine anderen Profile unterstützen, so muss dieses immer integriert sein, um eine Verbindung zwischen Bluetoothgeräten gewährleisten zu können. Wenn sich Bluetoothgeräte im „discoverable mode“ befinden sind sie für andere Bluetoothgeräte erkennbar und können Verbindungen eingehen. Deren Namen und grundlegende Eigenschaften werden in diesem Profil definiert. Außerdem wird ebenfalls definiert wie sich Geräte verhalten, die sich im Standby und Connecting-Status befinden, so dass zwischen Bluetoothgeräten immer Verbindungen aufgebaut werden können. Durch GAP werden zwei Rollen für Bluetoothgeräte definiert: „Peripheral“ und „Central“ Geräte. Bei den Peripheral Geräten handelt es sich um Wearables oder andere kleinere Zusatzgeräte, die sich mit einem leistungsstärkeren Central Gerät, beispielsweise einem Smartphone oder Tablet, verbinden [Tow15a].

Bei GATT handelt es sich um ein Standardprofil, auf dem alle Low Energie Profile aufbauen. Während GAP den Verbindungsaufbau definiert, wird durch GATT der Datenaustausch zwischen zwei miteinander verbundenen Geräten definiert. Das Profil baut auf dem Attribute Profile (ATT) auf. Das Profil definiert den Austausch kleiner Datenpakete, sogenannter „Attribute“, zweier miteinander über Bluetooth Low Energy verbundener Geräte. Die verbundenen Endgeräte nehmen dabei die Rollen Server und Client ein. Der Client sendet Anfragen an den GATT Server und kann Attribute, die er beim Server findet, lesen und/oder schreiben. Die Hauptaufgabe des Servers ist es Attribute zu speichern und dem Server nach einer Anfrage zur Verfügung zu stellen [TCR14]. Wie das GATT Profil unterteilt ist zeigt Abbildung 2.7. Jedes Bluetooth Gerät kann mehrere dieser Profile enthalten. Im GATT Profil werden einzelne Profile definiert, die sich aus Services zusammensetzen. Diese Services wiederum bestehen aus Charakteristiken. Diese Charakteristiken setzen sich aus folgenden Werten zusammen [Wic16]:

- Der Beschreibung der Charakteristik: Hier werden die Rechte beschrieben, also ob gelesen, abonniert und/oder geschrieben werden darf

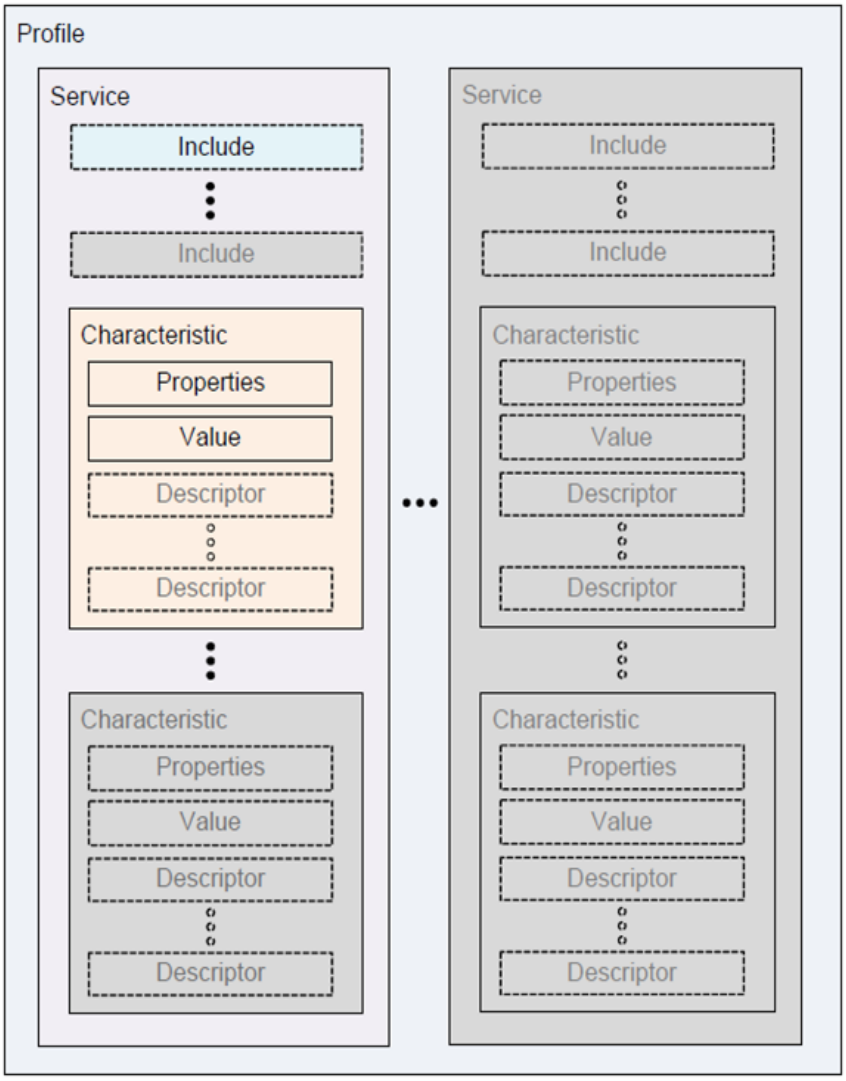

Abbildung 2.7.: Gatt-Profil [Blu16e]

- Der UUID: Eine eindeutige ID, die dieser Charakteristik zugeordnet ist.
- Der Wert der Charakteristik: Dieser kann je nach vergebenen Rechten gelesen und geschrieben werden.
- Optionale Deskriptoren: Diese können ebenfalls je nach vergebenen Rechten gelesen und geschrieben werden.

In Abbildung A.1 befindet sich ein Auszug der GATT Services, die vom Bluetooth SIG definiert werden. Unter diesen befindet sich beispielsweise das *Heart Rate*-Profil. Dieses setzt sich aus den drei Charakteristiken *Heart Rate Measurement*, *Body Sensor Location* und *Heart Rate Control Point* zusammen. Hierbei ist das *Heart Rate Measurement* als zwingend notwendig vorgesehen, die *Body Sensor Location* kann optional angegeben werden während die Implementierung der

Heart Rate Control Point-Charakteristik an das Energieaufwands-Feature geknüpft ist. Ist dieses vorhanden, so ist die Implementierung der *Heart Rate Control Point*-Charakteristik zwingend notwendig, andernfalls wird es nicht implementiert. In Abbildung A.2 ist ein Auszug der GATT Charakteristiken zu sehen. Hier ist unter anderem die eben erwähnte *Heart Rate Measurement*-Charakteristik aufgelistet, der die UUID 0x2A37 durch die Bluetooth SIG zugewiesen wurde [Tow15b].

Geräteentwickler können sich an diese Spezifikationen halten, so dass immer eindeutig bestimmt ist auf welche Sensoren zugegriffen wird. Jedoch können sie auch eigene UUIDs vergeben, wodurch sich Applikationsentwickler informieren müssen, wie sie auf Daten zugreifen können. Dies führt dazu, dass es keine Schnittstelle nach außen zum Betriebssystem gibt, in dem definiert wird auf welche Daten die Applikation zugreift. Somit kann der Nutzer die Berechtigungen für diese Applikation nicht zentral einschränken.

Daten können bei GATT auf zwei unterschiedliche Arten übertragen werden: mit und ohne Antwort. Wenn mit Antwort gesendet wird, wartet der Sender nach jedem Datenpaket auf eine Antwort des Empfängers und erst wenn diese erhalten wurde, wird das nächste Datenpaket gesendet. Dies führt zu einer geringeren Geschwindigkeit bei der Datenübertragung. Wird ohne Antwort gesendet, sendet der Sender alle Datenpakete nacheinander. Fehler bei der Datenübertragung bleiben dadurch unerkannt, jedoch ist die Geschwindigkeit der Datenübertragung ohne Antwort deutlich schneller als bei Datenübertragungen mit Antwort [Sto15]. Dies bedeutet aber auch, dass im Vornherein entschieden werden muss, was wichtiger ist: schnelle Datenübertragung oder korrekte Datenübermittlung. Jedoch wird bei beiden Datenübertragungsarten keine Plausibilitätsprüfung oder Verschlüsselung der Daten durchgeführt. Alle erfassten Daten werden unverschlüsselt direkt übermittelt.

2.2.3. Terminal I/O Profile

Beim Terminal I/O Profil handelt es sich um ein auf dem Bluetooth LE GATT Profil aufbauendes Profil, das von Stollmann entwickelt wurde [Sto15]. Die Rolle des „Peripheral“ (siehe Kapitel 2.2.2) übernimmt dabei ein von Stollmann entwickeltes und eingebautes Modul, während die Rolle des „Central“ auch von einem Smart Device eingenommen werden kann, wenn sie nicht ebenfalls von einem eingebauten Modul eingenommen wird. Durch die Nutzung einer Creditbasierten Übertragung werden beim Terminal I/O Profil die Nachteile der GATT-Übertragungen behoben. Sie ermöglicht sowohl eine schnelle, als auch korrekte Datenübertragung. Daten können nur übertragen werden, wenn der Empfänger dem Sender der Daten Credits zur Verfügung stellt und sobald diese aufgebraucht sind, müssen neue Credits gewährt werden, um weitere Datenübertragungen zu ermöglichen.

Der Verbindungsaufbau (siehe Abbildung 2.9) erfolgt über GAP und GATT und wird dann durch die „TerminalIO configuration“ erweitert. Der TerminalIO Server wird durch eine GATT Struktur in einem TerminalIO Service beschrieben (siehe Abbildung 2.8). Dieser enthält vier notwendige und weitere optionale Charakteristiken.

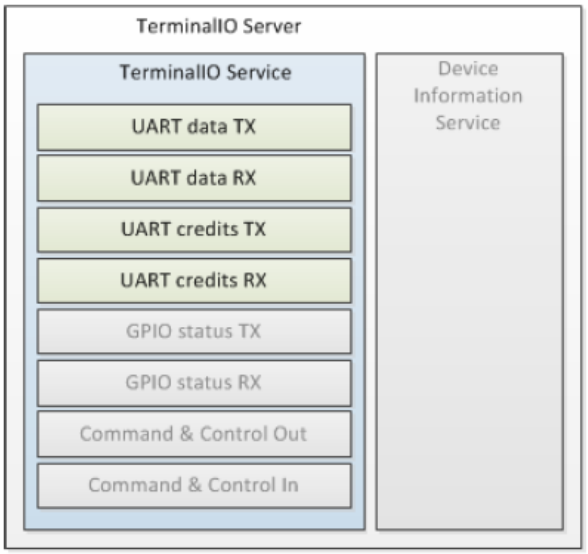

Abbildung 2.8.: GATT Struktur des Terminal I/O Servers [Sto15]

Beim Terminal I/O Profil versendet der Server Universal Asynchronous Receiver Transmitter (UART) Daten über Notifikationen und verwendet dabei die UART Data TX Charakteristik. Daten werden nur übertragen, wenn der Client den Server berechtigt Daten zu versenden und ihm eine ausreichende Anzahl an Credits über die UART credits RX Charakteristik zur Verfügung stellt. Ein Credit bezieht sich dabei auf eine UART data notification, unabhängig von ihrer Bytegröße. Dabei ist es die Aufgabe des Clients für eine angemessene Anzahl an Credits zu sorgen und einen flüssigen Datenaustausch zu ermöglichen.

Der Client nutzt die UART Data RX Charakteristik um Daten an den Server zu schicken. Dazu werden die UART Daten in den Wert der Charakteristik geschrieben. Der Client kann Daten nur senden, wenn er genügend Credits über die UART credits TX Charakteristik vom Server gewährt bekommt. Auch hier bezieht sich ein Credit auf eine Notifikation, unabhängig von ihrer Bytegröße. Hierbei ist es die Aufgabe des Clients die Anzahl der Credits zu verfolgen und in einem Zähler festzuhalten. Wenn dieser Zähler die Null erreicht kann der Client keine weiteren Daten an den Server schicken bis dieser ihm weitere Credits gewährt.

Da das Terminal I/O Profil auf GATT aufbaut, hat es teilweise auch dessen Nachteile übernommen. Auch hier werden Daten ohne weitere Überprüfungen übermittelt und ebenso ist keine Verschlüsselung der Daten vorgesehen.

2.3. Android Wear

Android Wear wurde im Jahr 2014 erstmals präsentiert. Es handelt sich dabei um ein Betriebssystem des Unternehmens Google Inc., das auf ihrem, auf Smartphones weit verbreiteten,

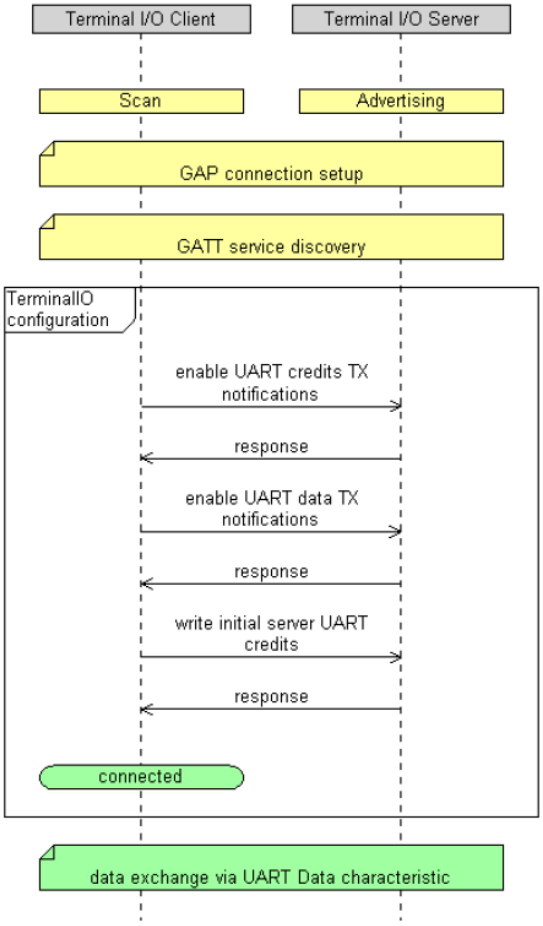

Abbildung 2.9.: Terminal I/O Verbindungsaufbau [Sto15]

Betriebssystem Android basiert. Android Wear wurde speziell für Wearables entwickelt und ermöglicht eine Verbindung von Smartphones mit dem Betriebssystem Android, mindestens ab Version Android 4.3 Jelly Bean, und Wearables mit dem Betriebssystem Android Wear über Bluetooth LE [Sma16a]. Es gibt keine eigenständigen Applikationen, die nur für Wearables entwickelt werden. Eine Nutzung eines Android Wear Endgerätes ohne Smartphone ist somit nicht möglich. Damit ein Smartphone mit einem Wearable mit dem Betriebssystem Android Wear verbunden werden kann ist die Android Wear App aus dem Google Play Store erforderlich [Goo17c]. Wird diese auf dem Smartphone installiert so können das Wearable und das Smartphone über eine Bluetooth LE Verbindung miteinander kommunizieren. Über die Android Wear App werden alle eingehenden Benachrichtigungen an die Uhr geschickt. Dabei kann es sich um SMS und Nachrichten von Messaging-Anwendungen handeln oder auch um die Steuerung eines Musik-Players.

Soll eine Applikation auf ein Wearable installiert werden, so wird diese über den Play Store des Smartphones auf dem Smartphone installiert. Beinhaltet diese App auch eine Applikation

Abbildung 2.10.: Integration Wear App in Android App [Uda16]

für das Wearable, so wird dieser Teil der Applikation auf der Uhr installiert. Abbildung 2.10 zeigt, dass das Android Wear Android Package (APK) im Android APK eingegliedert ist. Gibt es keine Verbindung zu einem Wearable so wird lediglich das Android APK auf dem Smartphone installiert. Android Wear unterstützt die Verbindung mehrerer Wearables zu einem Smartphone. Wird eine Nachricht empfangen, die durch Android Wear automatisch an ein Wearable Endgerät geschickt wird, so wird sie bei einer gleichzeitigen Verbindung zu mehreren Geräten automatisch an alle verbundenen Geräte verschickt [Uda16]. Möchte die Applikation auf interne Funktionalitäten der Smartwatch oder des Smartphones zugreifen, so wird seit Android 6.0 bei der ersten Nutzung der Funktionalitäten nach Berechtigungen gefragt. Dies ist sowohl beim Smartphone als auch bei der Smartwatch der Fall. Dabei handelt es sich um Funktionalitäten, wie z.B. die Nutzung der Kamera, Zugriff auf Körpersensoren oder Nutzung des GPS Sensors [Goo17b]. Eine Verschlüsselung der übertragenen Daten kann in der Applikation sowohl auf dem Wearable, als auch auf dem Smartphone durchgeführt werden. Jedoch handelt es sich hierbei um keinen Standard und es ist jedem Entwickler selbst überlassen die Daten zu verschlüsseln. Ebenso ist dies bei der Überprüfung der Plausibilität der zu übermittelnden oder übermittelten Daten der Fall. Der Entwickler einer Applikation kann dies sowohl auf Seiten des Wearables oder des Smartphones durchführen, verpflichtet dazu ist er jedoch nicht.

2.3.1. Wearable Data Layer API

Bei der Wearable Data Layer API handelt es es sich um eine Schnittstelle zum Wearable und dessen Applikationen. Sie beinhaltet verschiedene Datenobjekte, die die Datenübertragung ermöglichen. Sie wird über die Google Play services library zur Verfügung gestellt. Sie ist verfügbar, wenn auf den Geräten mindestens Android 4.3 installiert ist, eine Verbindung zu einem Wearable besteht und die Android Wear App [Goo17a] auf dem Gerät installiert ist [Goo16d].

Zugriff auf die Wearable Data Layer API erhält man, indem man eine Instanz des Google API Client erschafft. Der Google API Client stellt, wie in Abbildung 2.11 zu sehen ist, die Schnittstelle zwischen einer Applikation und den Google Play services dar. Im Google API Client kann dann über die Methode *addAPI()* auf die Wearable API zugegriffen werden.

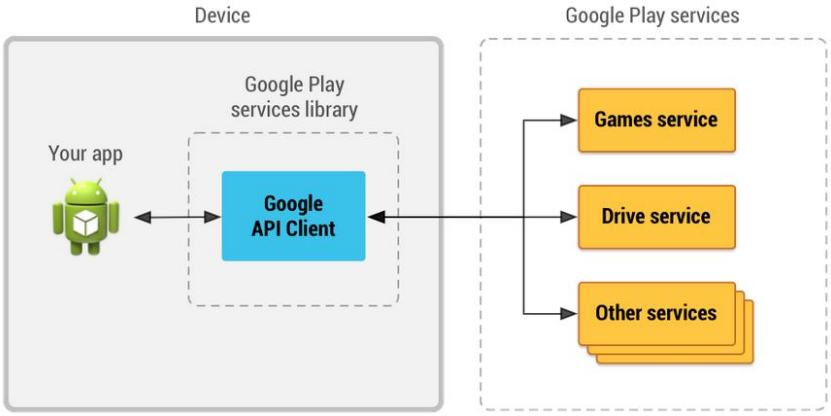

Abbildung 2.11.: Google API Client als Schnittstelle zu anderen Google Play services [Goo16d]

Es gibt mehrere Arten wie ein Wearable und ein Smartphone über Android Wear Daten verschicken können. Bei *Data Items* handelt es sich um Daten, die automatisch synchronisiert werden sobald die notwendigen Knoten in Reichweite sind. Dadurch wird ein Erreichen der Daten garantiert.
Bei *Messages*, die in der Message API definiert sind, handelt es sich um byte-arrays, die zu einem bestimmten vordefinierten Zeitpunkt gesendet werden. Hierbei ist das Erreichen der Daten nicht garantiert. Ist ein Wearable zu diesem Zeitpunkt nicht mit einem Smartphone verbunden so geht die Message verloren und es wird ein Fehler zurück gemeldet. Genutzt werden Messages für Remote Procedure Calls (RPCs), wie z.B. die Steuerung eines MediaPlayers auf dem Smartphone von einem Wearable aus.
Assets werden genutzt um Binary Large Objects (BLOBs), beispielsweise Bilddateien, zu verschicken. Assets werden an Data Items gehängt und das System sorgt automatisch für die Übertragung und speichert diese wenn nötig zwischen.
Der *WearableListenerService* ist dafür zuständig wichtige Events der Data Layer zu erkennen. Das System ist dafür zuständig den WearableListenerService zu den richtigen Zeitpunkten einzusetzen, wenn Data Items verschickt werden und ihn auszuschalten wenn keine Daten versendet werden.
Wenn aktiv auf Daten gewartet werden soll, weil der Nutzer beispielsweise eine Applikation aktiv nutzt, so kann statt des WearableListenerService der *DataListener* genutzt werden.
Um große Datenmengen, z.B. Musik, Filme,... zu übertragen, können *Channels* der ChannelAPI verwendet werden. Dies erfordert weniger Speicherplatz, da auf dem lokalen Gerät keine Kopie der zu versendeten Daten gespeichert wird.

Die gerade genannten APIs sind zur Kommunikation zwischen Wearables und Handhelds bei Android Wear entworfen und außer diesen sollten keine anderen Lösungen zur Datenübertragung bei Android Wear verwendet werden.

3. Kriterienkatalog für Verbindungsstandards

Im Folgenden wird ein Kriterienkatalog zusammengestellt. Die Reihenfolge der Kriterien spielt dabei im Hinblick auf ihre Relevanz keine Rolle. Diese Kriterien stellen Anforderungen dar, die Verbindungsstandards zur Datenübertragung zwischen Wearables und Smart Devices besitzen sollen. Als Grundlage dienen dazu die in Kapitel 2 vorgestellten Technologien. In diesem Kriterienkatalog sollen die Vorteile der vorgestellten Technologien enthalten sein, aber auch Kriterien, die in allen vorgestellten Technologien bisher nicht vorhanden sind. Diese Kriterien dienen als Grundlage für die in Kapitel 7 durchgeführte Evaluation.

Kriterium 1: **Flexibilität** (Krit1)

Wie in Kapitel 2 vorgestellt sind Wearables sehr vielfältig und die Entwicklung dieser schreitet schnell voran. Wichtig für eine Schnittstelle ist, dass sie viele unterschiedliche Wearables unterstützt. Dies kann von Uhren und Schmuck, die viele verschiedene Funktionalitäten besitzen, zu Sensoren reichen, die lediglich eine Aufgabe haben und nur diese durchführen können. Ein Nutzer soll viele unterschiedliche Wearables an einer Stelle gebündelt steuern und Auswerten können, ohne für jedes Wearable, das er sich neu zulegt, weitere neue Geräte zu benötigen, um dieses neue Wearable nutzen zu können. Außerdem wäre es wünschenswert, wenn dies nicht nur die Entwicklung neuer Wearables sondern auch weiterer smarter Geräte einschließen würde. Die Entwicklung smarter Fahrzeuge wird derzeit stark vorangetrieben, aber auch die Entwicklung smarter Geräte, z.B. Lautsprecher oder Kühlschränke, im Haushalt. Es sollte möglich sein alle diese Geräte ebenfalls, wie Wearables, ohne große Schwierigkeiten mit bisherigen Geräten nutzen und ansteuern zu können. Die Schnittstelle sollte nicht auf einzelne Technologien beschränkt sein, sondern keine Einschränkungen bei den unterstützen Geräten besitzen. Es soll keine technischen Beschränkungen geben, an die eine Schnittstelle gebunden ist.

Kriterium 2: **Interoperabilität** (Krit2)

Im Duden wird Interoperabilität als „Fähigkeit unterschiedlicher Systeme, möglichst nahtlos zusammenzuarbeiten“ [Bib16] definiert. Dieses Kriterium soll es Entwicklern von Applikationen ermöglichen ohne Anpassungen ihrerseits auf unterschiedliche Wearables zugreifen zu können. Das Wechseln des Wearables soll bei der Nutzung einer Applikation keine Konflikte

verursachen. Beispielsweise nutzt ein Entwickler für eine Applikation einen Beschleunigungssensor. Für die Applikation spielt es keine Rolle wo die Messung der Beschleunigung stattfindet. Ist das Smartphone mit einer Smartwatch verbunden, soll der Beschleunigungssensor des Wearables genutzt werden, befindet sich kein Wearable in der Nähe kann auf den Beschleunigungssensor des Smartphones zurückgegriffen werden. Der Entwickler definiert lediglich, dass seine Applikation auf einen Beschleunigungssensor zugreift.

Kriterium 3: **Sicherheit** (Krit3)

Die Sicherheit von Daten spielt in unserer Gesellschaft eine immer größere Rolle. Niemand möchte, dass private Daten, seien es Daten in Form von Nachrichten oder Daten, die Sensoren über uns erfassen, in falsche Hände geraten. Deshalb sollte eine Schnittstelle zu Wearables sicher sein. Diese Anforderung wird in diesem Kriterienkatalog so definiert, dass eine Übermittlung der Daten verschlüsselt vorgenommen wird und nur durch autorisierte Stellen entschlüsselt werden kann. Auch bereits übermittelte Daten sollen weiterhin verschlüsselt gespeichert werden können.

Kriterium 4: **Datenkontrolle** (Krit4)

Benutzt ein Nutzer Wearables, die Daten über ihn erheben, so soll er die Kontrolle über seine Daten besitzen. Er soll sich jederzeit informieren können, welche Apps auf welche Daten zugreifen und diesen Zugriff einschränken können. Außerdem soll er die Möglichkeit besitzen einer Applikation verfälschte Daten zukommen zu lassen, wenn er dies möchte. Keine Applikation soll ohne das Wissen des Nutzers Daten erheben und/oder auswerten können, wenn der Besitzer hiermit nicht einverstanden ist.

Kriterium 5: **Bündelung** (Krit5)

Wearables besitzen, wie in Kapitel 2 beschrieben, weniger Rechen- und Batterieleistung als andere technische Geräte. Deshalb werden von den meisten Applikationen, die auf Wearables laufen, wenige bis kaum Rechenaufgaben durchgeführt. Stattdessen werden alle Daten an das verbundene Smart Device weitergeleitet, welches dann die Rechenoperationen übernimmt. Um Batterieleistung zu sparen, aber auch um dem Nutzer einen leichteren Überblick über die erhobenen Daten zu ermöglichen, umfasst diese Anforderung eine Bündelung aller Daten, die auf dem Wearable erfasst und weitergeleitet werden und durch den Nutzer kontrolliert werden können, so dass weniger Applikationen aus unterschiedlichen Quellen redundante Informationen erfassen müssen. Wird eine Fitnessapplikation und eine Schrittzählerapplikation unterschiedlicher Entwickler genutzt, so sollen nicht beide Applikationen den Schrittsensor einzeln erfassen müssen. Sie sollen eine zentrale Stelle zur Datenübermittlung ansprechen, wenn sie die Daten übermittelt haben möchten. Dadurch wird der Sensor nur von der zentralen Stelle aus angesprochen und es findet vom Wearable an das Smartphone nur eine Datenübermittlung des gleichen Wertes statt.

Kriterium 6: **Plausibilität** (Krit6)

Verwendet ein Nutzer eine Applikation, kann er sich nicht sicher sein, ob die Daten die übermittelt werden ihre Richtigkeit haben. Nutzt er beispielsweise einen Herzfrequenzmesser und es werden ihm Werte, die größer als eintausend sind, angezeigt, so handelt es sich um Messfehler. Diese Daten sollen nicht in weiterführende Statistiken aufgenommen werden und Mittelwerte oder ähnliches beeinflussen. Solche Werte sollen herausgefiltert werden und in keine weiteren Berechnungen einfließen. Wenn möglich, sollen diese Werte schon vor der Datenübermittlung herausgefiltert werden, um so die zu übertragende Datenmenge so gering wie möglich zu halten.

4. Konzept

In diesem Kapitel wird ein Konzept für eine Schnittstelle zwischen Wearables und Smart Devices erarbeitet, welches möglichst die Anforderungen aus Kapitel 3 erfüllt. Als Smart Device wird hierbei ein Smartphone verwendet, da dieses häufig in Verbindung mit einem Wearable genutzt wird und es sich bei Smartphones um weit verbreitete Smart Devices handelt [Sta16b].

Wird heutzutage eine Applikation auf einem Wearable, speziell auf einer Smartwatch installiert, so ist diese im Allgemeinen nur in Verbindung mit einem Smartphone nutzbar. Eine Android-Smartwatch lässt sich nur in Verbindung mit einem Smartphone nutzen [RPP14], während eine Applewatch ohne iPhone zwar nutzbar ist, aber nicht ihren kompletten Funktionsumfang ausschöpfen kann [Tec16]. Der Großteil der Applikation wird auf dem Smartphone ausgeführt, während die Smartwatch dazu dient Benachrichtigungen anzuzeigen oder mithilfe ihrer eingebauten Sensoren Daten zu erfassen und diese an das Smartphone weiterzuleiten. Dies funktioniert über das Prinzip der zweigeteilten Applikation, wie in Kapitel 2.3 erläutert. Wird eine Applikation auf einem Smartphone installiert, die auch einen Teil für ein Wearable besitzt und es ist ein Wearable verbunden, so wird dieser Teil der Applikation auf dem Wearable installiert. So können Sensoren in einer Smartwatch beispielsweise Sportler unterstützen, die ihr Smartphone eine Zeit lang nicht bei sich führen. Wenn diese eine Runde joggen, erfasst der Teil der Applikation, der sich auf der Smartwatch befindet, über die GPS Sensoren den zurück gelegten Weg und überträgt diesen erst an den Teil der Applikation, die sich auf dem Smartphone befindet, wenn sich dieses im Bluetoothradius befindet. Auch andere Bewegungssensoren werden auf diese Art erst nach der sportlichen Betätigung auf das Smartphone übertragen. Besitzt ein Nutzer beispielsweise ein Smartphone und ein Wearable so können verschiedene Applikationen auf dem Wearable installiert sein, die alle die gleichen Daten erfassen. Ein Schrittzähler zählt die Schritte, während eine andere Fitness-Applikation ebenfalls die Schrittzahl erfasst. Das bedeutet, dass Applikationen auf der Smartwatch Daten erfassen und an „ihre“ Applikationen auf dem Smartphone senden, wie in Abbildung 4.1 erkennbar ist. Wenn beispielsweise die Teile der Applikationen 1 und 2, die sich auf dem Wearable befinden, lediglich einen und zwar den gleichen Sensor erfassen und diese Daten an das Smartphone verschicken, so sind diese Teile der Applikation identisch. Trotzdem mussten sowohl die Entwickler der Applikation 1 als auch der Applikation 2 jeweils die gleiche Routine schreiben.

Außerdem erfassen beide Apps redundante Daten. In beiden Fällen wird der Schrittsensor genutzt und beide Applikationen verbinden sich über einen Bluetooth Datenstrom mit dem

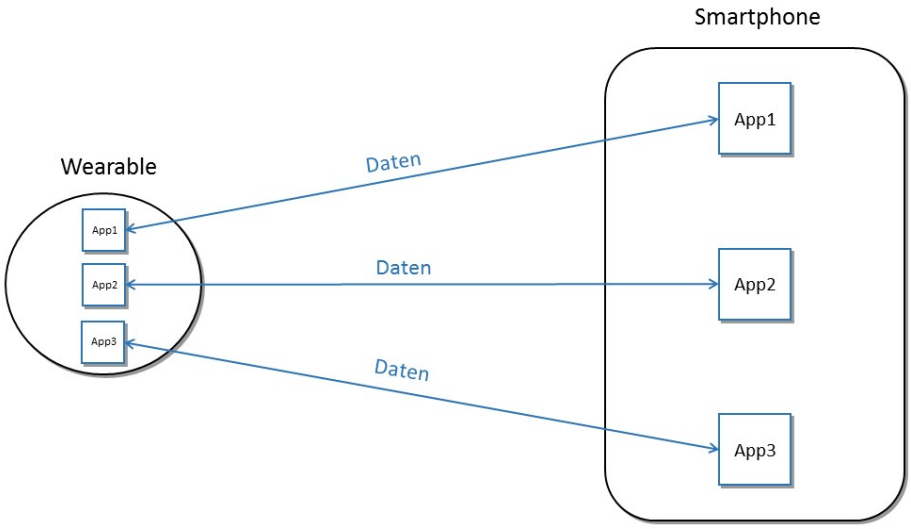

Abbildung 4.1.: Applikationen auf Wearables und Smartphones bisher

Smartphone. Beides kostet Rechen- und somit auch Batterieleistung des Wearables. Vorteilhafter wäre es, wenn die Applikationen auf dem Wearable keine redundanten Daten erfassen und diese parallel an das Smartphone schicken, sondern eine Applikation die Datenerfassung übernimmt und diese Daten an das Smartphone weiterleitet, wie es in Abbildung 4.2 dargestellt ist. In dieser Abbildung stellt die *App X* die App dar, die als einzige Daten auf dem Wearable sammelt und diese gebündelt an das Smartphone weiterleitet. Der Teil der *App X*, der sich auf dem Smartphone befindet, übernimmt dann die Verteilung der Daten an die jeweiligen Applikationen auf dem Smartphone, welche die Daten benötigen. Möchten Entwickler nun eine Applikation schreiben, die Daten der Sensoren braucht, müssen sie sich nur noch auf den auf dem Smartphone befindlichen Teil der Applikation konzentrieren. Für die Daten des Wearables, die sie benötigen, können sie auf eine Schnittstelle der *App X* zugreifen. Dabei soll es für den Entwickler keine Rolle spielen, welches Wearable an das Smartphone angeschlossen ist. Wenn die entwickelte Applikation auf Pulsdaten zugreifen möchte und das Smartphone ist in diesem Moment mit einer Smartwatch verbunden, so erhält sie über die Schnittstelle zur *App X* die Pulsdaten der Smartwatch. Ist aber zu einem anderen Zeitpunkt lediglich ein Pulssensor ohne Smartwatch mit dem Smartphone verbunden, so erhält die Applikation des Entwicklers die Pulsdaten des Pulssensor. Für die Applikation des Entwicklers ist es nur entscheidend, dass sie Pulsdaten bekommt, woher sie diese erhält spielt keine Rolle. Dies hat den Vorteil, dass die Applikation auf dem Smartphone unabhängig von den angebunden Geräten ihre Funktionen erfüllen kann und der Entwickler der Applikation sich nicht mit unterschiedlichen Technologien auseinander setzen muss, sondern über Schnittstellen die Daten erhält, die er benötigt.

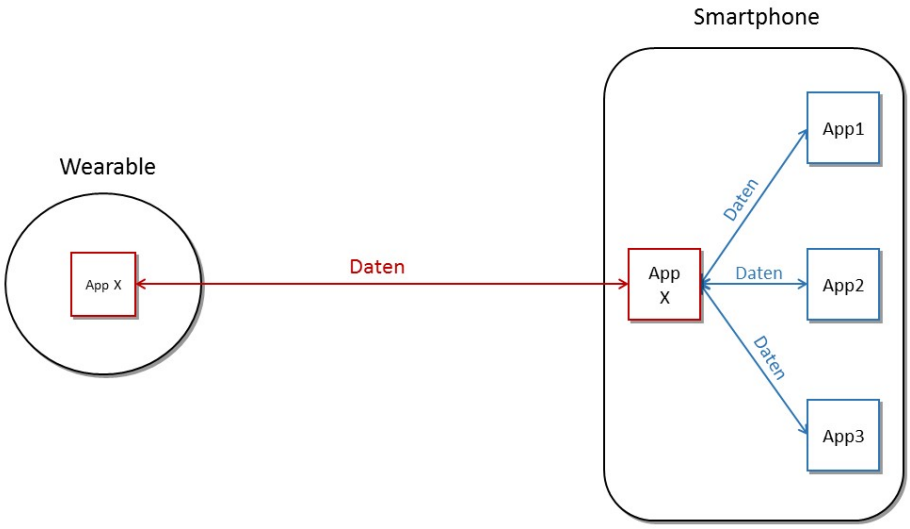

Abbildung 4.2.: Datenübertragung von und zur Smartwatch über eine Applikation

Der Nutzer hat keine Garantie, dass die Daten, die erfasst werden richtig sind oder überhaupt korrekt sein können. Sensoren könnten falsche Daten aufgrund von Messfehlern aufzeichnen und diese werden trotzdem an Applikationen weitergeleitet, die vielleicht Statistiken bereit stellen und deren Werte durch diese Messfehler nicht mehr korrekt sind. Der Nutzer soll sich darauf verlassen können, dass die Daten plausibel sind oder sich zumindest in einem Bereich befinden, der plausibel sein kann. Wird zum Beispiel eine Gehgeschwindigkeit von über 100 km/h aufgezeichnet, so soll dieser Wert verworfen und nicht an die Applikationen weitergeleitet werden. Dass so eine Kontrollinstanz vorhanden ist war bisher nur für die Entwickler von Applikationen ersichtlich, da dies zum Nutzer nicht kommuniziert wurde. Wurde eine solche Kontrollinstanz eingeführt, dann musste der Entwickler diese für jede seiner Applikationen jedes mal gesondert implementieren. Die *App X* des neuen Konzepts soll eine Plausibilitätskontrolle für mögliche Werte enthalten, so dass sich der Nutzer sicher sein kann, dass sich die übermittelten Werte in plausiblen Bereichen befinden. Dabei wird diese Kontrolle auf dem Wearable-Teil der *App X* durchgeführt, so werden falsche Daten erst gar nicht an das Smartphone übertragen.

Wie in der Einleitung in Kapitel 1 erwähnt, bestehen Sicherheitslücken bei Wearables, die vor allem im gesundheitlichen Bereichen zu massiven Gefahren führen können. Um Angreifern den Missbrauch von Daten zu erschweren und die Privatsphäre des Nutzers zu wahren sollen im neuen Konzept die Daten zwischen Wearable und Smartphone verschlüsselt gesendet werden. So sind selbst bei einem unrechtmäßigen Zugriff auf diese Daten die Daten für den Angreifer nutzlos . Ein weiterer wichtiger Bestandteil des Konzepts ist das Berechtigungssystem,

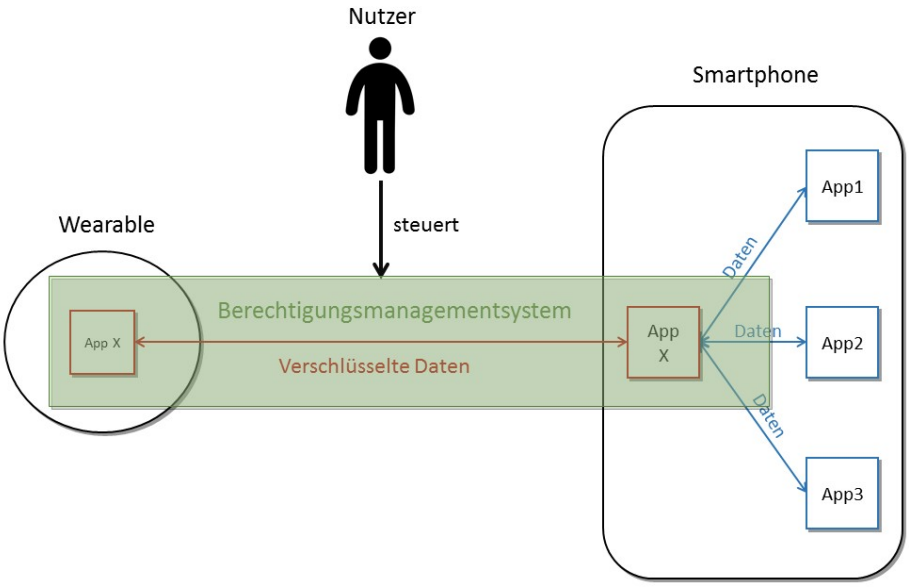

Abbildung 4.3.: Integration eines Berechtigungssystem

das in Abbildung 4.3 zu sehen ist. Über dieses System kann der Nutzer direkt steuern welche Applikationen auf welche Daten zugreifen können. Möchten Applikationen auf Daten zugreifen so benötigen sie die explizite Erlaubnis des Nutzers. Dies soll über eine für den Nutzer leicht verständliche Oberfläche steuerbar sein. Dieses Berechtigungssystem steuert dann, ob die *App X* die Daten überhaupt erheben darf und wenn ja, an welche Applikationen diese Daten dann von dem auf dem Smartphone befindlichen Teil der *App X* weitergeleitet werden dürfen. Das heißt, der Nutzer soll für jede Applikation, die sich auf seinem Smartphone befindet, einstellen können welche Daten diese erhält und welche nicht. Diese Regeln sollen so frei wie möglich sein, das heißt, dass der Nutzer Applikationen je nach Tageszeit oder je nach angemeldetem Nutzer unterschiedliche Daten zur Verfügung stellen kann. Auch soll der Nutzer einstellen können, ob Applikationen richtige, verfälschte oder frei erfundene, sogenannte Dummy-Daten, erhalten. Größtmögliche Individualität für jeden Nutzer ist ein zentraler Bestandteil des Konzepts.

Die Daten, die heutzutage auf Smartphones gespeichert sind, werden meist unverschlüsselt gespeichert. Bei Android Smartphones muss beispielsweise die Verschlüsselung des Geräts manuell durchgeführt werden und ist nicht von Beginn an aktiviert [Her16]. Durch das Berechtigungssystem kann der Nutzer zwar entscheiden welche Applikationen auf legalem Weg Zugriff auf Daten erhalten, jedoch wird durch dieses ein illegaler Zugriff auf die auf dem Smartphone gespeicherte Daten nicht vollständig unterbunden. Finden Datenübertragungen und -speicherungen sensibler Daten auf dem Smartphone nur verschlüsselt statt, kann dem Nutzer auf diese Weise zu mehr Privatheit und Sicherheit verholfen werden.

5. Umsetzung

In diesem Kapitel wird die Umsetzung des in Kapitel 4 vorgestellten Konzepts erläutert. Bei einem zentralen Bestandteil des Konzepts handelt es sich um das Berechtigungssystem. Es gibt unterschiedliche Berechtigungssysteme, auf die bei der Umsetzung des Konzepts zurück gegriffen werden kann. Stach vergleicht in seiner Arbeit verschiedene Aspekte diverser Berechtigungssysteme [Sta13]. Zwei dieser Aspekte, kontextsensitive Regeln und Dummy Daten, sind im Konzept in Kapitel 4 enthalten. Beim Aspekt der kontextsensitiven Regeln handelt es sich um die Möglichkeit, dass Regeln komplett individualisiert eingegeben werden können, wie zum Beispiel nach Tageszeit oder benutzerabhängig, während Dummy Daten es dem Nutzer ermöglichen eine Applikation zu nutzen, aber dieser lediglich Dummy Daten, also keine realen Daten, zur Verfügung zu stellen. Im Vergleich der unterschiedlichen Berechtigungssysteme in Stachs Arbeit wird deutlich, dass lediglich die PMP diese zwei Funktionalitäten unterstützt. Somit wird in dieser Arbeit für die Umsetzung des neuen Konzepts die PMP genutzt.

5.1. PMP - Privacy Management Platform

Bei der PMP handelt es sich um ein Berechtigungsmanagementsystem, das auf Smartphones genutzt werden kann. Ihre Funktionsweise befindet sich bildlich gesprochen zwischen den installierten Applikationen des Smartphones und den Funktionen des Betriebssystems. In Abbildung 5.1 ist die Funktionsweise der PMP dargestellt, die im Folgenden nochmals kurz mit den wichtigsten Komponenten erläutert wird [SM15]. Ein Nutzer eines Smartphones installiert auf seinem Smartphone eine PMP-kompatible Applikation (N & 1). Diese Applikation enthält sogenannte *Service Features*. Diese Service Features sind die Funktionen für die spezielle Berechtigungen erteilt werden sollen. Dabei kann es sich zum Beispiel um die Nutzung verschiedener Sensoren, der Kamera oder den Zugriff auf auf dem Smartphone gespeicherte Bilder handeln. Die installierte Applikation registriert sich mit ihren Service Features bei der PMP (2) und es wird in der *Privacy Policy* ein Eintrag für diese Applikation erstellt. Der Nutzer wird über die Service Features der Applikation informiert und legt für jedes Service Feature, das genutzt wird, initiale Berechtigungen fest (3). Die Funktionen, die die Applikationen benötigen, also die Schnittstellen zwischen den Applikationen und dem Betriebssystem, sind in sogenannten *Ressourcen* festgelegt. Diese Ressourcen sind kein direkter Bestandteil der PMP, sondern Services, die auch nachträglich hinzugefügt werden können. Benötigt eine Applikation eine Ressource, die noch nicht vorhanden ist, so wird diese automatisch aus einem geschützten Datenarchiv installiert und steht dem Nutzer ab diesem Moment zur Verfügung (4).

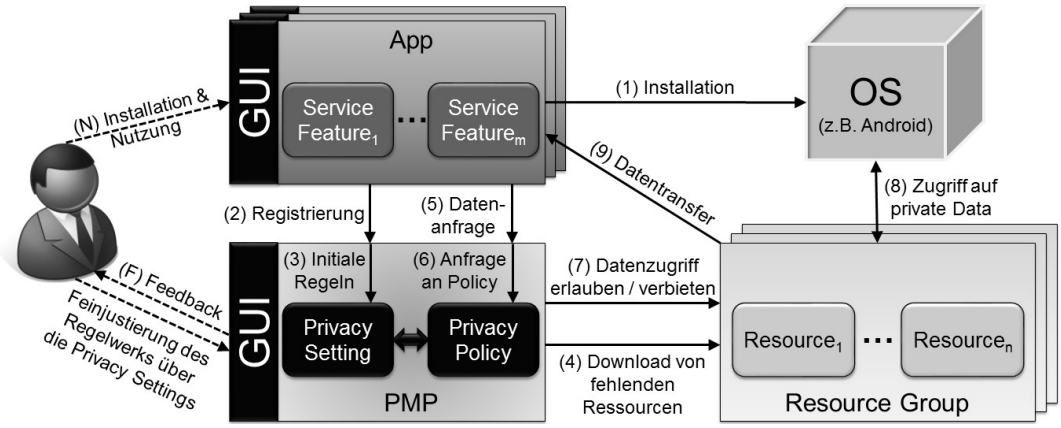

Abbildung 5.1.: Funktionsweise der PMP [SM15]

Ressourcen, die inhaltlich zusammen gehören, können in *Resource Groups* zusammengefasst werden. Wird nun von einer Applikation, die bei der PMP registriert ist, ein Service Feature aktiv benötigt, so schickt die Applikation eine Datenanfrage an die PMP (5) und es wird von der PMP in der *Privacy Policy* überprüft über welche Berechtigungen diese verfügt (6). Wird der Zugriff erlaubt, (7) so wird über die entsprechende Ressource auf die benötigte Funktion des Betriebssystems zurückgegriffen (8) und an die Applikation werden die gewünschten Daten weitergeleitet (9) [Sta13]. Zu jeder Zeit kann der Nutzer über die Oberfläche der PMP prüfen welche Berechtigungen für welche Applikation derzeit hinterlegt sind und diese auch ändern (F).

5.1.1. Secure Data Container

Durch den Einsatz der PMP wird das im neuen Konzept geforderte Berechtigungskonzept eingeführt, jedoch werden dabei die Daten noch nicht verschlüsselt. In ihrer Arbeit stellen Stach und Mitschang eine Erweiterung der PMP durch den SDC vor [Sta16a]. Dieser ermöglicht es Applikationen Daten auf sichere Weise zu speichern. Dazu werden die Daten in einem Container verschlüsselt gespeichert. In diesen verschlüsselten Containern sind die Daten in relationale Datenbanken als key-value-pairs gespeichert. Nach außen hin kann der Container über eine Schnittstelle angesprochen werden. Der Besitzer der Daten erhält immer Zugang zu diesen Daten, kann aber auch anderen Applikationen Zugriff auf diese erteilen. Dies geschieht über IDs, die vom SDC beim erstmaligen registrieren vergeben werden. Der genaue Ablauf dieser Anfrage ist in Abbildung 5.2 zu sehen. Gewährt der Besitzer von Daten anderen Applikationen den Zugriff werden die IDs dieser in einer *share*-Tabelle gespeichert. Beim Zugriff auf Dateien wird anhand dieser Tabelle dann den Applikationen der Zugriff und die Entschlüsselung der Daten gewährt oder nicht. Schreibrechte besitzt jedoch nur der Besitzer der Daten.

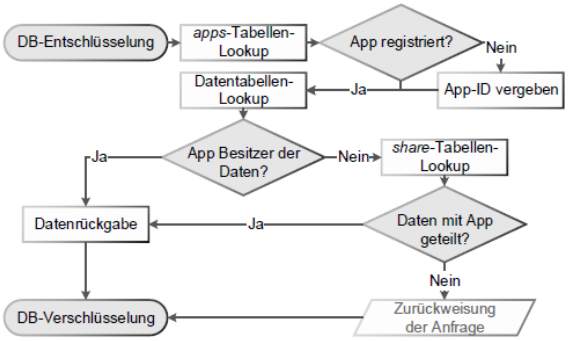

Abbildung 5.2.: SDC-Anfragealgorithmus als Aktivitätsdiagramm [Sta16a]

5.2. Eingliederung der PMP

Mit der PMP kann das geforderte Berechtigungssystem des Konzepts aus Kapitel 4 umgesetzt werden. An dieser Stelle wird erklärt, wie dies mit der PMP realisiert werden kann.

Beispielsweise soll eine Applikation den Wert eines Sensors einer Smartwatch auslesen und auf dem Smartphone anzeigen. Als Service Feature ist in dieser Applikation der Zugriff auf die Daten des benötigten Sensors hinterlegt. Mit diesem Service Feature registriert sich die Applikation bei der PMP und der Nutzer kann den Zugriff erlauben oder verbieten. Der Zugriff auf den Sensor und die Datenübermittlung an das Smartphone wird über eine Ressource durchgeführt. Diese Ressource ist in einem Datenarchiv zugänglich. Besitzt das Smartphone diese Ressource noch nicht, wird diese über die PMP heruntergeladen. Das Abrufen des Wertes des Sensors auf dm Wearable und die Datenübertragung vom Wearable zum Smartphone sind in dieser Ressource hinterlegt. Somit wird die komplette *App X* aus dem Konzept, die die Verbindung zwischen Wearable und Smartphone darstellt, als Ressource umgesetzt. Diese Ressource besteht aus zwei Teilen, dem Teil auf dem Wearable und dem Teil auf dem Smartphone. Der Wearable Teil ruft den Sensor ab, überprüft die Plausibilität des gelieferten Wertes und sendet den Wert an das Smartphone. Dort wird er über eine Schnittstelle zwischen Ressource und Applikation an die Applikation weitergeleitet. Natürlich nur, wenn der Nutzer davor den Zugriff für diese Daten über die PMP freigegeben hat. Möchte eine andere Applikation auf den gleichen Sensor des Wearables zugreifen, registriert diese sich ebenfalls mit diesem Service Feature bei der PMP und kann, bei erlaubtem Zugriff, über die gleiche Ressource den Wert erhalten. Über diese Struktur mit der Realisierung des Datenabrufs über eine Ressource, wird anstatt zwei Applikationen, die auf dem Wearable installiert sein müssen, nur noch der Wearable Teil der Ressource auf der Wearable benötigt.

Durch die Integration der SDC-Technologie in die Ressource kann der Aspekt der Sicherheit erfüllt werden. Für jede Applikation, die auf Daten des Wearables zugreifen möchte, wird ein Eintrag für die erlaubten Datensätze in der *share*-Tabelle angelegt. So können nur berechtigte Applikationen diese entschlüsseln und dadurch auf sie zugreifen.

Durch das Prinzip der Ressourcen wird für die Entwickler von Applikationen Flexibilität im Zugriff auf Sensordaten erreicht. Für den Entwickler einer Applikation spielt es keine Rolle mehr von welchem Gerät er seine Daten erhält. Ist das Smartphone in einem Moment mit einer Smartwatch verbunden, dann wird eine Ressource zur Datenübertragung zwischen Smartphone und Smartwatch angesprochen. Ist das Smartphone zu einem anderen Zeitpunkt mit einem Pulssensor verbunden, so wird eine Ressource zur Datenübertragung zwischen Smartphone und Pulssensor genutzt. In beiden Fällen bleibt seine Applikation lauffähig und kann ihre Funktionen weiter durchführen, sofern das Smartphone mit einem Gerät verbunden wird, das die erforderlichen Daten liefert.

6. Implementierung

In diesem Kapitel wird die technische Umsetzbarkeit des Konzepts anhand eines Prototyps aufgezeigt. In diesem Prototyp zeigt eine Applikation den Wert des Pulssensors eines Wearables an. Dazu wird eine Applikation umgesetzt, die als Service Feature, den Zugriff auf den Pulssensor beinhaltet. Für den Zugriff auf den Pulssensor des Wearables wird eine Ressource geschrieben, die den Sensor ausliest, abspeichert und für andere Applikationen zur Verfügung stellt. Der Nutzer kann über die PMP selbst entscheiden, ob die Applikation die Befugnis erhält auf diesen Wert zuzugreifen oder nicht oder ob dieser verfälscht weitergeleitet werden soll.

6.1. Android Wear

Für die Umsetzung musste ein Betriebssystem festgelegt werden. Es soll es ermöglichen die nötigen Anforderungen umzusetzen und es soll es sich am heutigen Markt orientieren. Dazu soll es die Möglichkeit bieten, das Konzept weiter zu entwickeln und für die Zukunft tauglich zu machen. Etwa 80 Prozent der Smartphones weltweit haben eine Android Version als Betriebssystem [Sch17]. Das und die Tatsache, dass Android ein sehr offenes Betriebssystem ist, sind die Gründe, warum die Wahl auf Android bei der Implementierung fiel. Als Betriebssystem für das Wearable stand somit gleichzeitig das Betriebssystem Android Wear fest, mit dessen Hilfe sich eine Verbindung zu einem Android Smartphone herstellen lässt (vgl. Kapitel 2.3). Hierbei gilt, dass Android Wear lediglich für die Anbindung des Smartphones an das Wearable genutzt wird. Das Konzept im Allgemeinen lässt sich auch auf andere Systeme übertragen und durch den auf dem Smartphone befindlichen Teil der Applikation ist das System nicht nur an Android Wear gebunden.

6.2. Hardware

Als Wearable wurde eine Moto 360 der ersten Generation verwendet. Im September 2014 wurde die Moto 360 vorgestellt. Sie verwendet das Betriebssystem Android Wear und nutzt die Funkverbindung über Bluetooth 4.0 Low Energy. Sie besitzt einen 1,5 Zoll LCD Bildschirm und enthält einen optischen Herzfrequenzmesser, einen Schrittsensor, sowie einen Umgebungslichtsensor. Sie kann durch Sprachbefehle und Gesten gesteuert werden. Im Prototyp wird die

Abbildung 6.1.: Die verwendeten Moto 360 und Samsung Galaxy S5

Moto 360 als Wearable verwendet, deren Pulssensor ausgelesen werden soll. Dazu wird auf ihr der Wearable Teil der Ressource installiert.

Als Smartphone wird ein Samsung Galaxy S5 verwendet, welches das Betriebssystem Android 6.0.1 Marshmallow verwendet und Bluetooth 4.0 unterstützt. Im Prototyp wird auf dem Samsung Galaxy S5 die Applikation installiert, die den Pulswert des Wearables anzeigt. Außerdem wird die PMP auf dem Samsung Galaxy S5 installiert, über die der Nutzer die Berechtigungen der Applikation steuern kann. Ebenso wird der auf dem Smartphone befindliche Teil der Ressource auf dem Samsung Galaxy S5 installiert. Dieser erhält die Daten der auf dem Wearable befindlichen Teil der Ressource, speichert diese und stellt sie anderen berechtigten Applikationen zur Verfügung. Beide verwendeten Geräte sind in Abbildung 6.1 abgebildet.

6.3. Puls-App

Die „Puls-App“ übernimmt im neuen Konzept exemplarisch die Rolle einer Applikation, die auf den Pulssensor zugreifen möchte. Im Konzept in Kapitel 4 einer der Apps 1,2 oder 3. Sie selbst wird als reine Android Applikation ohne Wearable-Teil implementiert, da diese Aufgabe der Ressource zuteil wird. Die Puls-App enthält das Service-Feature, dass sie auf die Daten des Pulssensors zugreifen möchte. Mit diesem registriert sie sich bei der PMP und der Nutzer kann über die Privacy Settings einstellen, welche Daten und in welcher Form (richtig oder verändert) zur Verfügung gestellt werden. Über die Oberfläche der Puls-App kann der Benutzer zum Beispiel entscheiden, ob er nur einen Wert des Pulssensors angezeigt bekommen möchte oder ob er den Verlauf des Pulssensors über einen längeren Zeitraum abrufen möchte. Wenn die Puls-App Daten über einen längeren Zeitraum erhält, kann sie beispielsweise den Verlauf

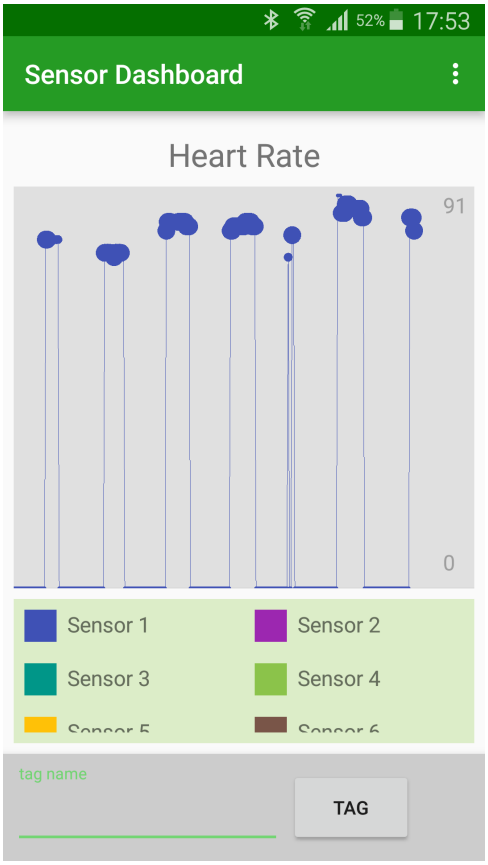

Abbildung 6.2.: Anzeige des Pulssensors der Dashboard App [Leh15]

dieser Daten als Schaubild aufzeigen, wie das auch bei der App Sensor Dashboard [Leh15] der Fall ist (siehe Abbildung 6.2). Im Rahmen dieser Arbeit wurde die Applikation einer Puls-App prototypisch umgesetzt.

6.4. Ressource

Die Ressource soll wie in Kapitel 5.2 erläutert als Schnittstelle zwischen Smartphone und Wearable dienen. Wie in Kapitel 5.1 erwähnt übernimmt eine Ressource in der Funktionsweise der PMP die Schnittstelle zwischen Betriebssystem und Applikation. In dieser Umsetzung übernimmt die Ressource ebenso die Schnittstelle zwischen Smartphone und Wearable. Die Funktionen des Wearables, besonders der Zugriff auf die Sensoren, kann als äquivalent zur Nutzung der Funktionen eines Smartphones angesehen werden. Die Ressource enthält die Funktionen und Methoden einer Android Wear App. Da sie die Verbindung zwischen Smartphone und Wearable darstellt, muss sie, wie alle Applikationen für Wearables auf Android Wear, wie in Kapitel 2.3 erläutert, aus zwei Komponenten bestehen. Dem Teil, der sich auf dem Smartphone befindet „mobile“, sowie dem Teil der sich auf dem Wearable „wear“ befindet.

Der mobile-Teil liest dabei den Sensor aus und prüft, ob der Wert plausibel ist. Das heißt ob er sich zwischen einem vorher definierten Minimum und Maximum befindet. Handelt es sich um einen plausiblen Wert wird dieser an den mobile-Teil der Ressource gesendet. Der Wert des Pulssensors kann in Form einer *Message* (vgl. Kapitel 2.3) vom wear-Teil an den mobile-Teil der Ressource übertragen werden. Der mobile-Teil lauscht dabei in Form eines *DataLayerListenerService* auf diese Nachricht und speichert die übermittelten Werte, um sie dann den berechtigten Applikationen zur Verfügung stellen zu können. Um eine Verschlüsselung zwischen dem Wearable und dem Smartphone bei der Datenübertragung zu erreichen können *Sealed Objects* verwendet werden . Diese ermöglichen es Entwicklern Objekte durch kryptografische Algorithmen zu verschlüsseln. Die Einstiegsklasse der Ressource enthält drei Implementierungen der zurückgegebenen Werte. Je nachdem, was der Nutzer einstellt, wird entweder der richtige Wert oder ein veränderter Wert zurückgegeben. Wird ein veränderter Wert zurück gegeben, kann die App darüber informiert werden (*getMockedAndroidInterface()*) oder die App wird nicht darüber informiert (*getCloakedAndroidInterface()*).

Die Schnittstelle zwischen Ressource und Applikation wird in einer Android Interface Definition Language (AIDL)-Datei beschrieben [Goo16a]. Darin sind die Methoden aufgelistet, über die eine Applikation auf die Ressource zugreifen kann. Im Beispiel der Puls-App enthält die AIDL-Datei die Methode, die den aktuellsten Wert des Puls-Sensors zurückgibt, als auch Methoden über die die Puls-Applikation einen Strom an Puls-Werten geliefert bekommt. Im Rahmen dieser Arbeit wurde eine Ressource prototypisch umgesetzt.

6.5. Folgerung

Es wurde gezeigt, wie eine Applikation mit einer Ressource umgesetzt werden kann. Für die Entwickler von Applikationen auf Wearables ändert sich einiges durch die Umsetzung des neuen Konzepts. Im hier gelieferten Beispiel müsste ein Entwickler, der eine Puls-App schreiben möchte, nur noch diese Puls-App ohne des auf dem Wearable befindlichen Teils schreiben. Die Ressourcen werden ihm zur Verfügung gestellt und bei Bedarf über die PMP heruntergeladen. Der Entwickler kann auf Sensoren egal welcher Wearables so zugreifen, als ob es sich um lokale Sensoren des Smartphones handelt. Ohne das neue Konzept muss jeder Entwickler, der auf den Puls-Sensor eines Wearables zugreifen möchte für jedes Wearable eine eigene Applikation schreiben, über die der Pulssensor ausgelesen wird. Auch für den Nutzer einer Applikation bietet sich der Vorteil, dass er jede Applikation mit unterschiedlichen Wearables nutzen kann, wenn diese lediglich die erforderlichen Daten liefern. Dies stellt eine große Vereinfachung für die Entwicklung und in der Nutzung von Applikationen dar.

7. Evaluation

In diesem Kapitel wird das in Kapitel 4 vorgestellte und in Kapitel 5 umgesetzte Konzept im Vergleich mit den in Kapitel 2 beschriebenen Technologien hinsichtlich der in Kapitel 3 vorgestellten Kriterien evaluiert. Bei den Kriterien handelt es sich um Flexibilität (Krit1), Interoperabilität (Krit2), Sicherheit (Krit3), Datenkontrolle (Krit4), Bündelung (Krit5) und Plausibilität (Krit6).

Die zu vergleichenden Technologien werden hier nochmals in Kürze vorgestellt:
GATT: Mithilfe dieses Profils werden Charakteristiken definiert, die in sogenannten Services gespeichert sind. Die Hersteller technischer Geräte sind dazu aufgefordert sich an diese Definitionen zu halten, so dass durch UUIDs auf die vordefinierten Charakteristiken zugegriffen werden kann.
HDP: Dieses Profil wurde entwickelt um medizinische Geräte miteinander zu verbinden. Es unterstützt mehrere simultane Datenkanäle über Bluetooth.
Terminal I/O Profile: Dieses Profil erweitert das GATT-Profil hinsichtlich Credits, die an verschiedene Geräte verteilt werden und so einen geregelteren Kommunikationsablauf ermöglichen.
Android Wear: Hierbei handelt es sich um ein Betriebssystem für Wearables mit Smartphones, die Android nutzen.
PMP-basierter Ansatz: Ein neues Konzept, das für mehr Sicherheit sorgt, die Datenkontrolle durch den Benutzer ermöglicht und durch weniger gleichzeitig auf dem Wearable installierte Apps, dieses schont und den Überblick für den Nutzer erleichtert.

Im Folgenden werden alle Kriterien des Kapitels 3 ausführlich für alle Technologien evaluiert.

7.1. Kriterium 1: Flexibilität

In diesem Kriterium geht es darum eine Verbindung zu verschiedenen Geräten, von Wearables, die lediglich aus Sensoren bestehen bis zu Wearables mit Betriebssystemen und auch eventuell weiterer smarter Geräte, herstellen zu können. Es sollen keine technischen Hindernisse überwunden werden müssen um unterschiedliche Arten von Wearables ansprechen zu können.

GATT: Mithilfe dieses Profils kann auf jedes Gerät, das Bluetooth LE nutzt zugegriffen und Sensordaten ausgelesen werden. Es ist keine weitere Umgebung nötig, denn dabei handelt es

sich um die direkten Signaldaten. Das bedeutet, dass jedes Wearable von Sensor zu Smartwatch, welches über Bluetooth oder Bluetooth LE eine Verbindung herstellt, über GATT angesprochen und Daten übermittelt werden können.

HDP: Das HDP wurde dafür entwickelt unterschiedliche Geräte anbinden zu können. Jedoch kann es bei kleineren Geräten, z.B. Sensoren, die keine große Batterieleistung besitzen, keine Anwendung finden, da es auf Bluetooth, statt auf Bluetooth LE ausgelegt ist und somit sind in diesem Bereich weitere Anpassungen beim HDP notwendig.

Terminal I/O: Das Terminal I/O Profil erfüllt dieses Kriterium nicht. Es muss ein spezielles Bluetooth Modul in den Geräten integriert sein, damit dieses Profil unterstützt wird. Dies ist heutzutage bei nur den wenigsten Geräten der Fall.

Android Wear: Android Wear unterstützt nur Geräte, die dieses Betriebssystem besitzen. Kleinere Geräte, wie z.B. Sensoren, besitzen gar kein Betriebssystem, somit werden diese nicht unterstützt.

PMP-basierter Ansatz: Durch die Nutzung der PMP und ihrer flexiblen Ressourcen wird durch das neue Konzept dieses Kriterium vollständig erfüllt. Eine Ressource kann als Applikation auf Android Wear laufen, wie es in der Implementierung der Fall war. Sie kann jedoch auch direkt auf das GATT Profil zugreifen und somit Sensordaten direkt unterstützen.

7.2. Kriterium 2: Interoperabilität

In diesem Kriterium geht es um die Interoperabilität verschiedener Wearables mit einer Applikation. Für den Entwickler einer Applikation soll es keine Rolle spielen auf den Sensor welches Wearables oder ob er auf den internen Sensor eines Smartphones zugreift.

GATT: Greift ein Entwickler einer Applikation über GATT auf die Daten eines Sensors zu, muss er die jeweiligen Spezifikationen kennen. Halten sich Gerätehersteller dabei an vorgegebene Spezifikationen, kann es möglich sein über die gleiche Applikation die Daten zu verarbeiten. Jedoch müsste der Entwickler erst kontrollieren, ob die übermittelten Daten den Vorgaben entsprechen, ob beispielsweise die richtigen Einheiten verwendet werden oder ob die gleiche UUID für den gleichen Sensor verschiedener Geräte vergeben wurde. Beim Zugriff über GATT auf gleiche Sensoren unterschiedlicher Geräte kann dem Entwickler keine Sicherheit für richtig übermittelte Daten gegeben werden. Dieses Kriterium wird nicht erfüllt.

HDP: Das HDP wurde zu diesem Zweck definiert und erfüllt dieses Kriterium vollständig. Das HDP baut auf dem ISO/IEEE 11073 Standard auf. Wenn Wearables über dieses Gerät verbunden werden, müssen sie sich an die vorgegebenen Standards halten, so dass jederzeit auf die Richtigkeit der angesprochenen Sensoren Verlass ist.

Terminal I/O: das Terminal I/O Profil baut auf GATT auf. Dadurch besitzt es die gleiche Problematik. Der Entwickler einer Anwendung muss eine genaue Spezifikation vorliegen

haben, welche Daten er über welche UUID abfragen kann. Hat er diese nicht vorliegen kann nicht zwischen unterschiedlichen Geräten gewechselt werden. Außerdem kann jeder Hersteller von Produkten über das Terminal I/O Profil eigene Spezifikationen definieren.

Android Wear: Das Android Wear Betriebssystem ermöglicht es über Methoden auf Sensoren zugreifen zu können. Gerätehersteller, die Android Wear auf ihren Geräten nutzen, müssen sich an die vorgegeben Spezifikationen halten. Dies ermöglicht es Entwicklern von Applikationen über Schnittstellen auf die richtigen Sensoren zuzugreifen, egal welches Wearable verbunden ist.

PMP-basierter Ansatz: Das Konzept ist nicht an eine Technologie gebunden. Es stellt ein Konzept dar, welches an unterschiedliche Technologien angepasst und auf diesen umgesetzt werden kann. Somit lässt es sich auf unterschiedliche Geräte verschiedenster Hersteller anwenden. Für jedes Gerät und jeden Sensor kann eine eigene Ressource zur Verfügung gestellt werden, über die auf diesen Sensor zugegriffen werden kann. Für den Entwickler einer Applikation spielt es keine Rolle welches Wearable seine Applikation nutzt. Dieses Kriterium wird durch das neue Konzept erfüllt.

7.3. Kriterium 3: Sicherheit

Dieses Kriterium fordert eine Verschlüsselung der Daten, bei der Übermittlung vom Wearable auf das Smartphone.

GATT: Da beim GATT Profil direkt die Sensordaten ausgelesen und übertragen werden findet keinerlei Verschlüsselung statt.

HDP: Beim HDP findet keinerlei Verschlüsselung statt, obwohl es sich um gesundheitsrelevante Daten handelt.

Terminal I/O: Beim Terminal I/O Profil werden die Daten auf keine Weise verschlüsselt.

Android Wear: Bei Android Wear können die Daten verschlüsselt übertragen werden, wenn Applikationsentwickler dies in ihre Applikationen integrieren. Jedoch findet im Allgemeinen bisher keine Verschlüsselung statt.

PMP-basierter Ansatz: Im neuen Konzept wird es empfohlen die Daten zu verschlüsseln, da es sich um persönliche und gesundheitliche Daten von Nutzern handelt und es wird eine Möglichkeit aufgezeigt, wie dies umgesetzt werden kann. Somit wird dieses Kriterium durch das neue Konzept erfüllt.

7.4. Kriterium 4: Datenkontrolle

Dieses Kriterium beschreibt, ob der Nutzer Kontrolle über seine eigenen Daten besitzt.

GATT: Wenn Applikationen über GATT Sensordaten erheben, gibt es keinen zentralen Kontrollmechanismus in welchem dies registriert und erlaubt werden kann. Das Betriebssystem der verwendeten Applikation wird nicht darüber informiert und kann dies somit auch nicht verhindern. Die Applikation müsste von sich aus melden, dass sie diese Daten erheben möchte und einen Kontrollmechanismus anbieten. Dieses Kriterium wird von GATT nicht erfüllt.

HDP: Auch beim Zugriff auf Sensordaten über das HDP in diesem Profil ist keine Kontrollstruktur implementiert, die regelt welche Daten erhoben werden und dass der Nutzer dies beeinflussen kann. Es besteht keine Schnittstelle zum Betriebssystem der verwendeten Applikation über die dies vom Nutzer geregelt werden könnte.

Terminal I/O: Auch in diesem Profil gibt es für den Nutzer keine Möglichkeit die Erfassung und Weitergabe der Daten aktiv mitzugestalten. Es wird keine Schnittstelle zum Betriebssystem zur Verfügung gestellt wird, über die das gesteuert werden könnte.

Android Wear: Seit Android 6.0 wird der Nutzer bei der Ausführung von Applikationen aktiv gebeten dieser verschiedene Berechtigungen zu erteilen [Goo16c]. Bis zu diesem Zeitpunkt konnte der Nutzer bei der Installation einer Applikation den Berechtigungen widersprechen, die Applikation dann aber nicht installieren. Die neue Funktionalität ist ebenso in Android Wear enthalten. Somit kann der Nutzer Applikationen den Zugriff auf bestimmte Berechtigungen, zum Beispiel auf den GPS Sensor oder Körpersensoren, sperren.

PMP-basierter Ansatz: Durch die Integration der PMP in das neue Konzept hat der Nutzer vollste Datenkontrolle und kann sogar, wenn er dies möchte, die Genauigkeit von Daten einstellen oder Dummy-Daten an bestimmte Applikationen weiterleiten.

7.5. Kriterium 5: Bündelung

Bei diesem Kriterium geht um die Bündelung von Daten, so dass sie zentral an einer Stelle, gesammelt und danach an die weiteren notwendigen Stellen weitergeleitet werden können.

GATT: Mithilfe des GATT Profils wird jeder Sensor einzeln über die UUID angesprochen, somit findet keine Bündelung von Daten statt.

HDP: Mithilfe des HDP können Daten teilweise gebündelt abgerufen und an weitere notwendige Geräte verteilt werden. Dies ist eine wichtige Aufgabe des HDP und somit wird dieses Kriterium durch das HDP erfüllt.

Terminal I/O: Hier findet keine Bündelung von Daten statt, da die Datenübertragung lediglich auf der GATT Ebene stattfindet. Dieses Kriterium wird durch das Terminal I/O Profil nicht erfüllt.

Android Wear: Bei Android Wear werden die Daten der Sensoren durch Applikationen auf den Wearables erfasst und dann lediglich an die dazugehörige Applikation auf dem Smartphone weitergeleitet. Da diese Applikationen nicht miteinander arbeiten findet keine Bündelung von Daten statt und dieses Kriterium wird von Android Wear nicht erfüllt.

PMP-basierter Ansatz: Das neue Konzept sieht eine allgemeine Applikation vor, die alle Sensordaten einer Wearable an das Smartphone weiterleitet und diese erst hinterher je nach Berechtigung der Applikationen weiter verteilt, vor. Somit wird dieses Kriterium durch das neue Konzept erfüllt.

7.6. Kriterium 6: Plausibilität

Dieses Kriterium enthält die Anforderung, dass es sich bei den übermittelten Daten um plausible Daten handelt, die möglichst keine Messfehler enthalten.

GATT: Bei GATT werden alle Daten übermittelt und es findet keine Kontrolle über diese statt. Es gibt keine Kontrollinstanz, die verhindern könnte, dass Messfehler übermittelt werden. Somit wird alles, was erfasst wird, auch übermittelt. Erst im Nachhinein wird in der Applikation unter Umständen eine Plausibilitätsprüfung durchgeführt. Diese Anforderung wird durch das GATT Profil nicht erfüllt.

HDP: Beim HDP werden keine Kontrollen auf Plausibilität durchgeführt, sondern alle Daten übermittelt. Auch hier kann erst im Nachhinein eine Plausibilitätsprüfung durch verwendete Applikationen erfolgen. Somit wird das Kriterium der Plausibilität nicht erfüllt.

Terminal I/O: Das Terminal I/O Profil besitzt ebenfalls wie das GATT Profil keinerlei Kontrollinstanzen der übermittelten Daten. Ebenfalls könnten hier erst nach dem Übermitteln der Daten Plausibilitätsprüfungen durchgeführt werden. Somit wird dieses Kriterium durch das Terminal I/O Profil nicht erfüllt.

Android Wear: Bei Android Wear hängt es von den genutzten Applikationen ab, ob falsche Daten übermittelt werden oder nicht. Besitzt der auf der Uhr befindliche Teil der Applikation eine Methode zur Kontrolle der Daten, so können diese herausgefiltert und nicht übermittelt werden. Diese Anforderung kann somit erfüllt werden, wenn es ein Anliegen des Applikationsentwicklers ist. Der Nutzer hat darauf keinen Einfluss.

PMP-basierter Ansatz: Im neuen Konzept soll es eine Kontrollinstanz geben, die überprüft ob Daten plausibel sind, bevor diese übermittelt werden. Dies wird durch die Ressource realisiert und somit wird diese Anforderung vollständig erfüllt.

	Krit1	Krit2	Krit3	Krit4	Krit5	Krit6
GATT	✓	X	X	X	X	X
HDP	X	✓	X	X	✓	X
Terminal I/O	X	X	X	X	X	X
Android Wear	X	✓	X	(✓)	X	(✓)
PMP-basierter Ansatz	✓	✓	✓	✓	✓	✓

Tabelle 7.1.: Evaluation des Kriterienkatalogs - Übersicht

7.7. Fazit

Es wurde eine Evaluation bezüglich der vorher definierten sechs Kriterien durchgeführt. Eine Übersicht über alle Kriterien befindet sich in 7.1. Ein Häkchen symbolisiert, dass das Kriterium von der links stehenden Technologie erfüllt ist, ein „X “ symbolisiert, dass dies nicht der Fall ist. Ein Häkchen in Klammern bedeutet, dass das Kriterium zu einem Großteil in der Technologie umgesetzt ist oder es die Möglichkeit für den Entwickler gibt dieses einfach umzusetzen.

Im direkten Vergleich des neuen Konzepts zu den anderen Technologien ist gut zu erkennen, dass das neue Konzept alle geforderten Kriterien umsetzt, während dies bei keiner anderen Technologie der Fall ist. Besonders im Kriterium der Sicherheit hebt es sich von den anderen Technologien ab, da dieser Aspekt bei keiner anderen Technologie umgesetzt ist. Das neue Konzept liefert Möglichkeiten wie alle Kriterien erfüllt werden können und ist zudem höchst anpassbar, da es durch den Mechanismus der Ressourcen alle anderen Technologien in sich vereinen kann.

8. Zusammenfassung und Ausblick

Es werden immer mehr smarte Geräte entwickelt, vor allem der Markt der Wearables befindet sich im Wachstum. Durch die Nutzung von einer stetig wachsenden Anzahl an technischen Geräten entstehen immer mehr Schnittstellen zwischen diesen. Man benötigt einen Mechanismus, durch den sich Wearables mit Smartphones verbinden lassen und der dabei verschiedene Anforderungen erfüllt. Vor allem wenn es sich um gesundheitsrelevante und persönliche Daten handelt, sollte der Nutzer sich sicher sein können, dass er plausible Daten auswertet, informiert wird welche Daten erhoben werden und selbst entscheiden können wer auf seine Daten Zugriff hat. Geräte unterschiedlichster Hersteller und Funktionen sollen miteinander verbunden werden und auch gegenseitig austauschbar sein, ohne dass es für den Nutzer von Applikationen zu Problemen führt. Auch für Entwickler soll die Entwicklung von Applikationen vereinfacht und standardisiert werden, so dass sie nicht mehr geräte- und technologieabhängig ist.

In dieser Arbeit wird ein Konzept für eine sichere Schnittstelle zwischen Smartphones und Wearables vorgestellt. Dazu wurden bestehende Schnittstellen und Technologien erläutert. In einem Kriterienkatalog wurden Anforderungen an das neue Konzept gestellt, anhand derer dieses dann erarbeitet wurde.

Für die Umsetzung des Konzepts wurde die PMP genutzt, da mit ihr ein Berechtigungssystem zur Verfügung gestellt wird, das individuelle Berechtigungen für den Nutzer und Nutzung von Dummy-Daten erlaubt und so Applikationen auch ohne Zugriff auf die realen Daten genutzt werden können. Sie erlaubt eine Kontrolle von sogenannten Service Features durch den Nutzer. Dieser kann einstellen, welche Applikationen auf welche Service Features zugreifen können, so dass der Nutzer diese individuell für verschiedene Applikationen einstellen kann. Durch die Integration der PMP konnte die Schnittstelle zwischen Wearable und Smartphone anhand einer Ressource entwickelt und prototypisch implementiert werden.

In einer Evaluation wurde das neue Konzept mit den anfangs erläuterten Technologien verglichen und anhand des Kriterienkatalogs ausgewertet. Hierbei konnte gezeigt werden, dass das neue Konzept alle Anforderungen erfüllt, während dies bei keiner anderen Technologie der Fall ist.

Durch die Nutzung der PMP kann das entwickelte Konzept erweitert und individuell angepasst werden. Die Ressourcen ermöglichen nicht nur eine Schnittstelle zu Wearables mit Betriebssystemen, wie es beim Prototyp der Fall ist, sondern auch eine direkte Kommunikation zu einzelnen Sensoren, wenn beispielsweise eine Ressource lediglich anhand des GATT Profils mit dem Sensor kommuniziert. Auch könnte das Kriterium der Bündelung weiter ausgearbeitet werden. So könnten unterschiedliche Applikationen die parallele Nutzung von einzelnen

Daten und Datenströmen weiter optimieren und so die Menge der zu übertragenden Daten deutlich verringern. Diese Möglichkeiten der Erweiterung führen dazu, dass das entwickelte Konzept noch weiter ausgereift und in vielen weiteren Bereichen integriert werden kann. Es ist nicht an bestehende Technologien gebunden und kann deshalb auch zukünftig für neue Technologien angepasst und weiter implementiert werden. So kann es beispielsweise weitere Anwendungen im Bereich der Smart Homes finden und die Schnittstellen zu diesen Geräten können anhand des entwickelten Konzepts ebenfalls sicherer gestaltet werden und weiterhin mit den schon vorhandenen Geräten des Nutzers gesteuert werden. So kann der Nutzer auch in seinem zukünftigen Smart Home für jedes angeschlossene Gerät einzelne Berechtigungen vergeben und so trotz Vernetzung seines Zuhauses weiterhin die Kontrolle über seine Daten behalten.

Literaturverzeichnis

[Ärz14] Ärzte Zeitung. *Am Körper tragbare Medizinprodukte vor dem Durchbruch?* 2014. URL: http://www.aerztezeitung.de/praxis_wirtschaft/medizinprodukte/article/868276/wearable-devices-koerper-tragbare-medizinprodukte-durchbruch.html (zitiert auf S. 13, 18).

[AVC15] B. Altenhoff, H. Vaigneur, K. Caine. „One Step Forward, Two Steps Back: The Key to Wearables in the Field is the App“. In: (Aug. 2015). DOI: 10.4108/icst.pervasivehealth.2015.259049 (zitiert auf S. 13).

[Bib16] Bibliographisches Institut GmbH. *Interoperabilität*. 2016. URL: http://www.duden.de/rechtschreibung/Interoperabilitaet (zitiert auf S. 31).

[Blu09] Bluetooth SIG, Inc. *HEALTH DEVICE PROFILE Implementation Guidance Whitepaper*. 2009 (zitiert auf S. 22, 23).

[Blu16a] Bluetooth SIG, Inc. *Bluetooth 5*. 2016. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth5 (zitiert auf S. 19).

[Blu16b] Bluetooth SIG, Inc. *Bluetooth Core Specification*. 2016. URL: https://www.bluetooth.com/specifications/bluetooth-core-specification (zitiert auf S. 19).

[Blu16c] Bluetooth SIG, Inc. *GATT Characteristics*. 2016. URL: https://www.bluetooth.com/specifications/gatt/characteristics (zitiert auf S. 60).

[Blu16d] Bluetooth SIG, Inc. *GATT Services*. 2016. URL: https://www.bluetooth.com/specifications/gatt/services (zitiert auf S. 59).

[Blu16e] Bluetooth SIG, Inc. *Generic Attributes (GATT) and the Generic Attribute Profile*. 2016. URL: https://www.bluetooth.com/specifications/generic-attributes-overview (zitiert auf S. 25).

[Bra15] M. Brand. *Die Zukunft von Wearables hängt am Handgelenk*. 2015. URL: https://de.statista.com/infografik/3364/wearable-absatzprognose/ (zitiert auf S. 14, 18, 19).

[Goo16a] Google Inc. *Android Interface Definition Language (AIDL)*. 2016. URL: https://developer.android.com/guide/components/aidl.html (zitiert auf S. 46).

[Goo16b] Google Inc. *Google Fit – Fitness-Tracking*. 2016. URL: https://play.google.com/store/apps/details?id=com.google.android.apps.fitness (zitiert auf S. 13).

[Goo16c] Google Inc. *Requesting Permissions at Run Time*. 2016. URL: https://developer.android.com/training/permissions/requesting.html (zitiert auf S. 50).

[Goo16d] Google Inc. *Sending and Syncing Data*. 2016. URL: https://developer.android.com/training/wearables/data-layer/index.html (zitiert auf S. 29, 30).

[Goo17a] Google Inc. *Android Wear - Smartwatch*. 2017. URL: https://play.google.com/store/apps/details?id=com.google.android.wearable.app&hl=en (zitiert auf S. 29).

[Goo17b] Google Inc. *App-Berechtigungen bei Android 6.0 und höher verwalten*. 2017. URL: https://support.google.com/googleplay/answer/6270602?hl=de (zitiert auf S. 29).

[Goo17c] Google Inc. *Häufig gestellte Fragen von neuen Android Wear-Nutzern*. 2017. URL: https://support.google.com/androidwear/answer/6056390?hl=de (zitiert auf S. 28).

[Her16] E. Hermann. *Android-Geräte: Die Verschlüsselung sollte aktiviert werden*. 2016. URL: https://www.androidpit.de/verschluesselung-sollte-niemandem-egal-sein (zitiert auf S. 38).

[Ins15] Institut für Qualität und Wirtschaftlichkeit im Gesundheitswesen (IQWiG). *Kann die kontinuierliche Glukosemessung die Insulinbehandlung verbessern?* 2015. URL: https://www.gesundheitsinformation.de/kann-die-kontinuierliche-glukosemessung-die.2196.de.html?part=behandlung-fg-xaqp-522m (zitiert auf S. 18).

[ITW17a] ITWissen. *ACL (asynchronous connectionless link)*. 2017. URL: http://www.itwissen.info/definition/lexikon/asynchronous-connectionless-ACL-Asynchron-verbindungslos.html (zitiert auf S. 20).

[ITW17b] ITWissen. *SCO (synchronous connection oriented link)*. 2017. URL: http://www.itwissen.info/definition/lexikon/SCO-synchronous-connection-oriented-SCO-Verfahren.html (zitiert auf S. 20).

[kio14] kioskea.net. *Funktionsweise von Bluetooth*. 2014. URL: http://de.ccm.net/contents/605-funktionsweise-von-bluetooth (zitiert auf S. 20).

[Leh15] J. Lehtimaeki. *Sensor Dashboard*. 2015. URL: https://play.google.com/store/apps/details?id=com.github.pocmo.sensordashboard (zitiert auf S. 45).

[Mar14] MarketWired. *MOTA® to Debut the MOTA SmartRing at IFA Berlin This Week*. 2014. URL: http://www.marketwired.com/press-release/motar-to-debut-the-mota-smartring-at-ifa-berlin-this-week-1942871.htm (zitiert auf S. 18).

[MBK+12] I. Muslukhov, Y. Boshmaf, C. Kuo, J. Lester, K. Beznosov. „Understanding Users' Requirements for Data Protection in Smartphones". In: *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering Workshops*. ICDEW '12. IEEE Computer Society, 2012, S. 228–235. ISBN: 978-0-7695-4748-0. DOI: 10.1109/ICDEW.2012.83 (zitiert auf S. 14).

[MVV+14] D. Martin, O. Vicente, S. Vicente, J. Ballesteros, M. Maynar. „I Will Prescribe You an App". In: *Proceedings of the 2014 Summer Simulation Multiconference*. SummerSim '14. Society for Computer Simulation International, 2014, 58:1–58:8. URL: http://dl.acm.org/citation.cfm?id=2685617.2685675 (zitiert auf S. 14).

[ODo15] B. O'Donnell. *The Slow Build: Smart WearablesForecast, 2014-2020*. 2015. URL: http://www.technalysisresearch.com/downloads/TECHnalysis%20Research%20Smart%20Wearables%20Forecast%20Summary,%20May%202015.pdf (zitiert auf S. 13).

[Par12] A. Parmar. *Hacker shows off vulnerabilities of wireless insulin pumps*. 2012. URL: http://medcitynews.com/2012/03/hacker-shows-off-vulnerabilities-of-wireless-insulin-pumps/#ixzz2aGxCwGxa (zitiert auf S. 13).

[RPP14] R. Rawassizadeh, B. A. Price, M. Petre. „Wearables: Has the Age of Smartwatches Finally Arrived?" In: *Commun. ACM* 1 (Dez. 2014), S. 45–47. DOI: 10.1145/2629633 (zitiert auf S. 35).

[Sam16] Samsung. *Samsung Gear S3*. 2016. URL: http://www.samsung.com/de/galaxy/gear-s3/ (zitiert auf S. 18).

[Sch03] P. Schnabel. *Kommunikationstechnik-Fibel: Grundlagen; Festnetz; Mobilfunktechnik; Breitbandtechnik; Netzwerktechnik*. Books on Demand, 2003. ISBN: 3-8330-0567-X (zitiert auf S. 20, 22, 24).

[Sch17] R. Schanze. *Smartphone-Betriebssysteme: Vergleich und Marktanteile*. 2017. URL: https://www.smartwatch.de/smartwatch/ (zitiert auf S. 43).

[SM13] C. Stach, B. Mitschang. „Privacy Management for Mobile Platforms – A Review of Concepts and Approaches". In: *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management*. MDM'13. IEEE, Juni 2013, S. 305–313. DOI: 10.1109/MDM.2013.45 (zitiert auf S. 15).

[SM15] C. Stach, B. Mitschang. „Der Secure Data Container (SDC)". In: *Datenbank-Spektrum* 15.2 (2015), S. 109–118. ISSN: 1610-1995. DOI: 10.1007/s13222-015-0189-y (zitiert auf S. 15, 39, 40).

[Sma16a] Smartwatch.de. *Android Wear*. 2016. URL: https://www.smartwatch.de/android-wear/ (zitiert auf S. 28).

[Sma16b] Smartwatch.de. *Smartwatches – kleine Wunderwerke*. 2016. URL: http://www.giga.de/downloads/android-6.0-marshmallow/specials/smartphone-betriebssysteme-vergleich-und-marktanteile/ (zitiert auf S. 19).

[Sop13] Sophos Ltd. *Security Threat Report 2014 Smarter, Shadier, Stealthier Malware*. 2013. URL: https://tavaana.org/sites/default/files/sophos-security-threat-report-2014.pdf (zitiert auf S. 14).

[Sta13] C. Stach. „Wie funktioniert Datenschutz auf Mobilplattformen?" In: *Tagungsband der 43. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*. Bd. 220. LNI. GI, Sep. 2013, S. 2072–2086. ISBN: 978-3-88579-614-5 (zitiert auf S. 39, 40).

[Sta16a] C. Stach. „Secure Candy Castle—A Prototype for Privacy-Aware mHealth Apps". In: *Proceedings of the 2016 IEEE 17th International Conference on Mobile Data Management*. MDM'16. IEEE, Juni 2016, S. 361–364. DOI: 10.1109/MDM.2016.64 (zitiert auf S. 40, 41).

[Sta16b] Statista GmbH. *Fakten zum Thema: Smartphones*. 2016. URL: https://de.statista.com/themen/581/smartphones/ (zitiert auf S. 35).

[Sto15] Stollmann. *Terminal I/O Profile. Client Implementation Guide, Stollmann Entwicklungs- und Vertriebs-GmbH*. 2015 (zitiert auf S. 26–28).

[TCR14] K. Townsend, C. Cufí, A. Robert Davidson. *Getting Started with Bluetooth Low Energy*. O'Reilly Media, 2014. ISBN: 978-1-4919-4951-1 (zitiert auf S. 24).

[Tec16] Techbook. *Lohnt sich die Apple Watch auch ohne iPhone?* 2016. URL: http://www.techbook.de/techstyle/lohnt-sich-die-apple-watch-auch-ohne-iphone (zitiert auf S. 35).

[TM14] K. Tehrani, A. Michael. *Wearable Technology and Wearable Devices: Everything You Need to Know*. 2014. URL: http://www.wearabledevices.com/what-is-a-wearable-device/ (zitiert auf S. 17).

[Tow15a] K. Townsend. *GAP*. 2015. URL: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap (zitiert auf S. 24).

[Tow15b] K. Townsend. *GATT*. 2015. URL: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt (zitiert auf S. 26).

[Uda16] Udacity. *How is an android app built*. 2016. URL: https://classroom.udacity.com/courses/ud875A/lessons/4582940110/concepts/45808002700923# (zitiert auf S. 29).

[US 17] U.S. Food and Drug Administration. *Cybersecurity Vulnerabilities Identified in St. Jude Medical's Implantable Cardiac Devices and MerlinAThome Transmitter: FDA Safety Communication*. 2017. URL: http://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm535843.htm (zitiert auf S. 13).

[Wea16] Wearables Tech. *Was sind Wearables?* 2016. URL: http://www.wearables-tech.de/was-sind-wearables-tech-trend (zitiert auf S. 13, 17).

[Wei91] M. Weiser. „The Computer for the 21st Century". In: *Scientific American* (Jan. 1991), S. 66–75. URL: http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html (zitiert auf S. 13).

[Wic16] S. van Wickern. *Bluetooth Low Energy: "Wie funktioniert Bluetooth Low Energy?"* 2016. URL: http://netzhaft.de/blog-post-bluetooth-low-energy-Was-ist-eine-Charakteristik.html (zitiert auf S. 24).

Alle URLs wurden zuletzt am 30. 01. 2017 geprüft.

A. Anhang

SpecificationName	SpecificationType	AssignedNumber	SpecificationLevel
Alert Notification Service	org.bluetooth.service.alert_notification	0x1811	Adopted
Automation IO	org.bluetooth.service.automation_io	0x1815	Adopted
Battery Service	org.bluetooth.service.battery_service	0x180F	Adopted
Blood Pressure	org.bluetooth.service.blood_pressure	0x1810	Adopted
Body Composition	org.bluetooth.service.body_composition	0x181B	Adopted
Bond Management	org.bluetooth.service.bond_management	0x181E	Adopted
Continuous Glucose Monitoring	org.bluetooth.service.continuous_glucose_monitoring	0x181F	Adopted
Current Time Service	org.bluetooth.service.current_time	0x1805	Adopted
Cycling Power	org.bluetooth.service.cycling_power	0x1818	Adopted
Cycling Speed and Cadence	org.bluetooth.service.cycling_speed_and_cadence	0x1816	Adopted
Device Information	org.bluetooth.service.device_information	0x180A	Adopted
Environmental Sensing	org.bluetooth.service.environmental_sensing	0x181A	Adopted
Generic Access	org.bluetooth.service.generic_access	0x1800	Adopted
Generic Attribute	org.bluetooth.service.generic_attribute	0x1801	Adopted
Glucose	org.bluetooth.service.glucose	0x1808	Adopted
Health Thermometer	org.bluetooth.service.health_thermometer	0x1809	Adopted
Heart Rate	org.bluetooth.service.heart_rate	0x180D	Adopted
HTTP Proxy	org.bluetooth.service.http_proxy	0x1823	Adopted
Human Interface Device	org.bluetooth.service.human_interface_device	0x1812	Adopted
Immediate Alert	org.bluetooth.service.immediate_alert	0x1802	Adopted
Indoor Positioning	org.bluetooth.service.indoor_positioning	0x1821	Adopted
Internet Protocol Support	org.bluetooth.service.internet_protocol_support	0x1820	Adopted
Link Loss	org.bluetooth.service.link_loss	0x1803	Adopted

Abbildung A.1.: Auszug der GATT Services [Blu16d]

Fat Burn Heart Rate Upper Limit	org.bluetooth.characteristic.fat_burn_heart_rate_upper_limit	0x2A89	Adopted
Firmware Revision String	org.bluetooth.characteristic.firmware_revision_string	0x2A26	Adopted
First Name	org.bluetooth.characteristic.first_name	0x2A8A	Adopted
Five Zone Heart Rate Limits	org.bluetooth.characteristic.five_zone_heart_rate_limits	0x2A8B	Adopted
Floor Number	org.bluetooth.characteristic.floor_number	0x2AB2	Adopted
Gender	org.bluetooth.characteristic.gender	0x2A8C	Adopted
Glucose Feature	org.bluetooth.characteristic.glucose_feature	0x2A51	Adopted
Glucose Measurement	org.bluetooth.characteristic.glucose_measurement	0x2A18	Adopted
Glucose Measurement Context	org.bluetooth.characteristic.glucose_measurement_context	0x2A34	Adopted
Gust Factor	org.bluetooth.characteristic.gust_factor	0x2A74	Adopted
Hardware Revision String	org.bluetooth.characteristic.hardware_revision_string	0x2A27	Adopted
Heart Rate Control Point	org.bluetooth.characteristic.heart_rate_control_point	0x2A39	Adopted
Heart Rate Max	org.bluetooth.characteristic.heart_rate_max	0x2A8D	Adopted
Heart Rate Measurement	org.bluetooth.characteristic.heart_rate_measurement	0x2A37	Adopted
Heat Index	org.bluetooth.characteristic.heat_index	0x2A7A	Adopted
Height	org.bluetooth.characteristic.height	0x2A8E	Adopted
HID Control Point	org.bluetooth.characteristic.hid_control_point	0x2A4C	Adopted
HID Information	org.bluetooth.characteristic.hid_information	0x2A4A	Adopted
Hip Circumference	org.bluetooth.characteristic.hip_circumference	0x2A8F	Adopted

Abbildung A.2.: Auszug der GATT Characteristics [Blu16c]

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift