

Konzeption und prototypische Umsetzung  
einer Steuerzentrale eines smarten Büros mit  
dem Fokus einer einfachen Handhabung der  
formalisierten Interaktionen für  
Softwareentwickler

MASTER-THESIS

für die Prüfung zum

Master of Science

des Studienganges Professional Software Engineering

an der

Knowledge Foundation @ Reutlingen University

von

**Mikka Jenne**

Abgabedatum 31. August 2022

Bearbeitungszeitraum

24 Wochen

Teilnehmernummer

800864

Kurs

PSEJG20

ErstprüferIn

Prof. Dr. Natividad Martinez Madrid

ZweitprüferIn

Dr. Robin Braun

## **Zusammenfassung**

Augmented Reality ist eine Technologie, die dem Nutzer ein visuelles Erlebnis mit einer angereicherten Welt voller virtueller Objekte ermöglicht. Das Resultat, eine Kombination aus Realität und Virtualität, bietet dem Benutzer eine neue Art der Wahrnehmung der Gegenwart.

Diese Bachelorarbeit befasst sich mit der Konzeption und Umsetzung eines industriellen Assistenzsystems unter Verwendung der Augmented Reality Technologie. Dabei soll die Umgebung mit Hilfe des SLAM Verfahrens analysiert werden, um auf dieser Basis dreidimensionale Objekte als Referenz zu realen Objekten im Raum virtuell platzieren zu können. Durch die entstehende Visualisierung können Informationen zu den jeweiligen Objekten in eine Datenbank eingetragen und angezeigt werden, dadurch kann das Überwachen von Industriemaschinen vereinfacht werden.

Zu dem Konzept gehört sowohl die Ausarbeitung der grundlegenden Softwarearchitektur, als auch ein allgemein-gültiges Datenmodell zur Persistierung der generierten Daten. Für die bestmögliche Umsetzung der Augmented Reality Experience werden hierzu bereits schon bestehende Frameworks und Software Development Kits, beispielsweise Google ARCore, verwendet.

Der entstandene Prototyp ist ein eigenständiges System. Die Architektur ist modular aufgebaut, um eine stetige Weiterentwicklung zu gewährleisten.

## **Abstract**

Augmented Reality is a technology that enables the user to have a visual experience with an enriched world full of virtual objects. The result offers the user a new way of perceiving surrounding as an Combination of reality and virtuality.

This bachelor thesis deals with the conception and implementation of an industrial assistance system using augmented reality technology. The environment can be analyzed with the help of the SLAM method in order to be able to place three-dimensional objects virtually as a reference to real objects in space. The resulting visualization enables information on the respective objects to be entered in a database and displayed, which enables the simplified monitoring of Industrial machines.

The concept includes the development of the basic software architecture as well as a generally applicable data model for saving the generated data. Already existing frameworks and software development kits where used for the best possible implementation of the augmented reality experience as Google ARCore.

The created prototype is an standalone system. The architecture is modular in order to ensure continuous further development.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Forschungsfragen . . . . .	3
1.3	Zielsetzung der Arbeit . . . . .	3
1.4	Forschungsstrategie und Forschungsmethoden . . . . .	3
1.4.1	Experteninterview . . . . .	4
1.4.2	Systematisches Literatur Review . . . . .	4
1.4.3	Usability-Tests . . . . .	5
1.5	Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Internet der Dinge . . . . .	7
2.1.1	Paradigmen und Kommunikationsmodelle . . . . .	12
2.1.2	Historische Entwicklung . . . . .	13
2.1.3	Ziele von IoT . . . . .	14
2.2	Smart Home . . . . .	15
2.2.1	Smart Office - Intelligentes Büro . . . . .	19
2.2.2	Historische Entwicklung . . . . .	20
2.2.3	Ziele von Smart Home . . . . .	21
2.3	Technologien . . . . .	22
2.3.1	Übertragungsmethoden . . . . .	22
2.3.2	MQTT . . . . .	23
2.3.3	AMQP . . . . .	26
2.4	Home Assistant . . . . .	27
2.4.1	Konzept und Architektur . . . . .	28
2.4.2	Ziele und Schwerpunkte . . . . .	30
2.4.3	Stärken und Schwächen . . . . .	31
2.4.4	Optionen der Regel- und Automatisierungserstellung . . . . .	32
2.5	openHAB . . . . .	33
2.5.1	Konzept und Architektur . . . . .	34
2.5.2	Ziele und Schwerpunkte . . . . .	38
2.5.3	Stärken und Schwächen . . . . .	38

2.5.4	Optionen der Regel- und Automatisierungserstellung . . . . .	39
2.6	Vergleich von Home Assistant und openHAB . . . . .	40
<b>3</b>	<b>Stand der Technik</b>	<b>41</b>
3.1	Systematisches Literaturreview . . . . .	41
3.1.1	Ziele des Systematischen Literaturreviews . . . . .	41
3.1.2	Suchstrategie- und anfragen . . . . .	41
3.1.3	Datenextraktion und Synthese . . . . .	44
3.2	Zusammenfassung . . . . .	44
3.2.1	Publikationen . . . . .	45
3.2.2	Stand der Technik aus Nutzer- und Produktsicht . . . . .	48
<b>4</b>	<b>Anforderungsanalyse</b>	<b>49</b>
4.1	Marktanalyse . . . . .	49
4.1.1	Allgemeine Marktsituation und Marktprognose . . . . .	50
4.2	Zielgruppenanalyse . . . . .	54
4.2.1	Ziel der Zielgruppenanalyse . . . . .	54
4.2.2	Zielgruppe Benutzer . . . . .	54
4.2.3	Zielgruppe Anwender . . . . .	56
4.3	Anwendungsfälle - Use Cases . . . . .	57
4.3.1	Check In mit einem Service-Roboter . . . . .	58
4.3.2	Notfall-Evakuierung mit einem Service-Roboter . . . . .	59
4.4	Experteninterview . . . . .	59
4.4.1	Ziel des Experteninterviews . . . . .	60
4.4.2	Aufbau des Experteninterviews . . . . .	60
4.4.3	Zusammenfassung der Experteninterviews . . . . .	60
4.5	Anforderungen . . . . .	61
<b>5</b>	<b>Konzeption</b>	<b>63</b>
5.1	Ziel der Konzeption . . . . .	63
5.2	Anwendungsumfeld . . . . .	64
5.3	Architekturkonzept . . . . .	64
5.3.1	Überlegungen, Anstöße und Herausforderungen . . . . .	64
5.3.2	Schnittstellen . . . . .	64
5.3.3	Datenbanken . . . . .	64
5.4	Softwarekonzept . . . . .	64
5.5	Auswahl des Frameworks . . . . .	64
5.5.1	OSGi . . . . .	65
5.5.2	Spring Boot . . . . .	65

<b>6</b>	<b>Umsetzung</b>	<b>66</b>
6.1	Implementierung . . . . .	66
6.1.1	Auswahl der genutzten Technologien . . . . .	66
6.1.2	Aufbau der Architektur . . . . .	66
6.1.3	Einbindung der Funktionen abgeleitet von der Konzeption . . . . .	66
6.2	Ergebnis . . . . .	66
<b>7</b>	<b>Diskussion und Evaluation</b>	<b>67</b>
7.1	Analyse des Konzepts der Eigenentwicklung . . . . .	67
7.2	Usability Test . . . . .	67
7.2.1	Ziel des Usability Tests . . . . .	67
7.3	Experten Interview . . . . .	67
7.3.1	Ziele des Experten Interviews . . . . .	67
7.4	Vergleich zwischen Eigenentwicklung und bestehenden Softwarelösungen . . . . .	67
<b>8</b>	<b>Fazit</b>	<b>68</b>
<b>9</b>	<b>Ausblick</b>	<b>69</b>
	<b>Index</b>	<b>I</b>
	<b>Literaturverzeichnis</b>	<b>VI</b>
	<b>Anhang</b>	<b>VI</b>
<b>A</b>		<b>VI</b>
<b>B</b>		<b>VII</b>
<b>C</b>		<b>IX</b>

# Kapitel 1

## Einleitung

Die folgende Master-Thesis befasst sich mit der Konzepterstellung einer zentralen Steuerzentrale, die dem Entwickler die formalen Interaktionen, weitere Funktionen hinzuzufügen, erleichtern soll. Hierfür werden bereits bestehende Plattformen für Smart Home analysiert und daraus ein Konzept erstellt, die den Anforderungen entsprechend einen größeren Mehrwert in der Weiterentwicklung der Plattform bietet. Die Umsetzung des ausgearbeiteten Konzeptes wird nur in sehr geringem Maß behandelt.

In diesem Teil der Arbeit wird auf die Motivation des Themas eingegangen. Darüber hinaus werden sowohl die Forschungsfragen als auch die Zielsetzung der Arbeit genauestens dargelegt. Darauf folgend findet eine Übersicht über die Arbeit im Gesamten statt, mit der die Inhalte angerissen werden. Eine nähere Betrachtung des Standes der Technik untermauert die Beweggründe dieser Themenwahl und Ausarbeitung dessen.

### 1.1 Motivation

Jede neu entwickelte Technologie durchlebt im Laufe der Entstehung und Publikation ein enormes Aufsehen. So lange bis diese Technik eine standardisierte Verwendung in der Gesellschaft findet oder sich als unpraktikabel erweist und nicht weiter vorangetrieben oder eingestellt wird. Es wird in der Zeit des Aufkommens und der Forschung viel darüber fantasiert, debattiert und geplant, ohne jedoch die Ausmaße und Resultate der Forschungen und Praktiken abwägen zu können. Durch fehlende Erfahrung und nicht ausgereifte Konzepte werden Höhepunkte und Illusionen erwartet, die zu diesem Zeitpunkt technisch nicht umsetzbar sind. Um solche kühnen Versprechungen und Übertreibungen, sogenannte Hypes, die jede neue technologische Idee mit sich bringt, von dem zu differenzieren was wirtschaftlich umsetzbar ist, werden bestimmte Phasen der Entwicklung durchlaufen. [GARTNER 2022]

Die oben erwähnten Phasen der Entwicklung sind in einem sogenannten Hype-Zyklus, engl. Hype-Cycle, dargestellt. Dieser Zyklus ist ein visualisiertes Modell, das die Entwicklung einer neuen Technologie von der Innovation und Entstehung über die Forschung und Umsetzung bis hin zur ausgereiften Marktfähigkeit repräsentiert und so diese Phasen der Entwicklung versinnbildlicht.

Entwickelt wurde der Hype Cycle von der Gartner Inc. Forschungsgruppe. Durch die Mitarbeiterin Jackie Finn wurden die Definitionen der Entwicklungsphasen<sup>1</sup> geprägt. Diese sind wie folgt in fünf Phasen dargestellt:

1. *Innovationsauslöser, engl. Innovation Trigger*: Ein potentieller technologischer Durchbruch löst die Dinge aus. Frühe Proof-of-Concept (PoC) Ansätze und ein großes Medieninteresse lösen eine erhebliche Publizität aus. Oft gibt es keine brauchbaren Produkte und die Marktreife ist nicht bewiesen. [GARTNER 2022]
2. *Höhepunkt überhöhter Erwartungen, engl. Peak of Inflated Expectations*: Frühe Publizität bringt eine Reihe von Erfolgsgeschichten hervor – oft begleitet von zahlreichen Misserfolgen. Einige Unternehmen ergreifen Maßnahmen; viele nicht. [GARTNER 2022]
3. *Trog der Ernüchterung, engl. Trough of Disillusionment*: Das Interesse schwindet, da Experimente und Implementierungen nicht liefern. Hersteller der Technologie reißen es heraus oder scheitern. Investitionen werden nur fortgesetzt, wenn die überlebenden Anbieter ihre Produkte zur Zufriedenheit der frühzeitigen Anwender verbessern. [GARTNER 2022]
4. *Steigung der Erleuchtung, engl. Slope of Enlightenment*: Mehr Beispiele dafür, wie die Technologie dem Unternehmen zugute kommen kann, beginnen sich zu herauszukristallisieren und werden allgemeiner verstanden. Produkte der zweiten und dritten Generation erscheinen von den Technologieanbietern. Mehr Unternehmen finanzieren Pilotprojekte; Konservative Unternehmen bleiben vorsichtig. [GARTNER 2022]
5. *Plateau der Produktivität, engl. Plateau of Productivity*: Mainstream-Akzeptanz beginnt sich abzuheben. Kriterien zur Bewertung der Lebensfähigkeit des Anbieters sind klarer definiert. Die breite Markteinsatzbarkeit und Relevanz der Technologie zahlen sich eindeutig aus. [GARTNER 2022]

Nachdem ein innovativer Gedanke den *Höhepunkt überhöhter Erwartungen* passiert hat, z.B. die Revolutionierung der Softwareentwicklung oder Szenarien, wie z.B. die Vollautomatisierung eines Gebäudes oder Service-Roboter die uneingeschränkt interagieren können, die man in der Form nur aus Science-Fiction Filmen kennt, folgt der *Trog der Ernüchterung*. In Folge dessen wird festgestellt, dass die Erwartungen nicht in Gänze übertragbar sind, bzw. nur zu einem geminderten Teil in die Realität umgesetzt werden können und der verfolgte Gedanke an Interesse verliert. Nach erneutem Aufgriff der Technologie findet eine realistischere Beurteilung der Innovation statt, die dazu beiträgt, dass die Technologie wieder an Interesse gewinnt. Die objektive und realitätsnahe Betrachtungsweise formt ein neues und realistisches Bild der Potentiale, als auch der Grenzen. Mit dem neu gewonnenen Maßstab geht die ehemals neue innovative Idee in eine routinierte Technologie über, die an Anerkennung gewinnt und in der breiten Masse akzeptiert wird. Die Technologie erfährt mit steigender Zuwendung eine stetigere Weiterentwicklung, die dann zu einer Community geformt wird. Mit der Erreichung dieses Status befindet sich die Innovation, bezogen auf den Hype Cycle, in der letzten Phase, dem *Plateau der Produktivität*, und bestätigt so die Marktreife. Dieser Zeitpunkt löst

---

<sup>1</sup>Die Entwicklungsphasen der Gartner Inc. ist unter folgender URL zu finden: "<https://www.gartner.com/en/research/methodologies/gartner-hype-cycle>"



die Zukunftsvision auf und es handelt sich um eine am Markt etablierte Technologie.

Zum aktuellen Zeitpunkt befindet sich die Technologie rund um Plattformen für intelligente Geräte im privaten Bereich, engl. *Smart Home (SH)* oder *Connected Home*, im Anfangsstadium der letzten Phase, dem sogenannten *Plateau der Produktivität*. Mit zunehmender Akzeptanz werden im Umfeld des Internets der Dinge, engl. *Internet of Things*, stetig Szenarien entwickelt, die das Wachstum und die Verwendung von solchen Plattformen vorantreibt. Mit einer immer tiefer gehenden Forschung und Umsetzung von Anwendungsbeispielen werden Bereiche offenbart, die eine solche Plattform im privaten als auch im geschäftlichen Umfeld immer attraktiver gestaltet. Mit steigender Konnektivität und Kompatibilität mehreren Geräten und Gegenständen können Bereiche und Szenarien, wie die Steuerung von Service-Robotern, umgesetzt werden. Der jetzige technologische Fortschritt und die über die Forschungsjahre gesammelten Erfahrungen bringt das Segment der intelligenten Geräte der IoT den ursprünglich angedachten Visionen und Ideen näher, sodass ein weiterer Ausbau dieser Technologie und dessen Anwendung stattfindet und sich vollständig in den Markt etabliert. Der finale Schritt der endgültigen Marktreife ist ein faszinierender und wichtiger Grund für meine Motivation, mich dieser Technologie und der dahinterstehenden Theorie zu widmen.

Die Einsatzgebiete von intelligenten Geräten, beziehungsweise die Verwendung einer Kompaktlösung beschränkt sich räumlich auf Gebäude, Häuser und Wohnungen, bieten trotz dessen viele Verwendungs- und Einsatzmöglichkeiten. Diese sehen wie folgt aus:

- Komfort
- Entertainment
- Überwachung und Sicherheit
- Steuerung von Prozessen
- Management von Automationen

Neben der Affinität von Smart Home zum Internet der Dinge und der damit einhergehenden Technologie bringt diese Vorteile mit sich, wie z.B. die Modernisierung von Wohn- und Bürogebäuden und die angestrebte Verbesserung der Lebensqualität.

## 1.2 Forschungsfragen

## 1.3 Zielsetzung der Arbeit

## 1.4 Forschungsstrategie und Forschungsmethoden

Dieser Abschnitt der Arbeit widmet sich der Forschungsstrategien und der darauf angewendeten Forschungsmethoden. Hierbei werden die Strategien kurz erläutert und die Methoden skizziert. Die Anwendung der Strategien findet zu späterem Zeitpunkt statt.

Diese Arbeit ist in die drei Phasen aufgeteilt, Forschung, Konzeption und Evaluation. Zuerst erfolgt durch ein systematisches Literaturreview eine Analyse des aktuellen Stands der Technik, welche die Forschungsfragen aufgreift und anhand dessen den aktuellen Stand identifiziert. Schwerpunkt dabei liegt auf der einfachen Handhabung der formalisierten Interaktionen eines Softwareentwicklers bei der Weiterentwicklung, bzw. bei der Ergänzung einer bestehenden Softwarelösung, um weitere Anwendungsfälle abzudecken. In der Phase der Anforderungserhebung werden Experteninterviews durchgeführt, mit denen die Anforderungen des Produktes, für welches ein Konzept erstellt wird, identifiziert werden. Ergänzend dazu werden im Rahmen der Arbeit weitere Anforderungen durch das Requirements Engineering (RE) und der Anwendung des *user-centered design*-Prinzips<sup>2</sup> sowie des *target group analysis*-Ansatzes (4.2) ermittelt. Dabei wird die Nutzerorientierung auf den Softwareentwickler ausgelegt. Zur anschließenden Evaluation wird ein Usability Test durchgeführt und darauf erneut ein Experten Interview, damit Eindrücke über das Konzept entstehen und die Erfahrungen der Experten während des Usability Tests in die Beantwortung der Forschungsfrage mit einfließen.

Die soeben genannten Forschungsmethoden werden im Nachfolgenden näher erklärt.

#### 1.4.1 Experteninterview

Zu Anfang wird bei einem Experten Interview der Hintergrund des Interviews erläutert. Nachdem die interviewende Person den Grund dargelegt hat, stellt ein Forscher einem Experten ausgewählte Interview-Fragen, die für die Forschung relevant sind und dazu beitragen die Forschungsfrage zu beantworten.

Die Interview-Fragen können als offene oder geschlossene Fragen formuliert werden, je nachdem ist eine beliebige Reihe an Antworten möglich oder nur eine begrenzte Anzahl an Antworten [ROBSON 2002]. Der Verlauf des Interviews kann von dem Forscher selbst bestimmt werden. Sind nur bestimmte Antworten gewünscht, so können die Fragen strukturiert und geschlossen formuliert werden. Ebenso gibt es den semi-strukturierten und den unstrukturierten Ansatz, bei dem das Interview offen gestaltet werden kann. Der Semi-Strukturierte Ansatz eignet sich, wenn spezielle Fragen adressiert werden, bzw. eine Reihenfolge festgelegt ist, sodass der Interviewer eine klare Reihenfolge hat. Mit dem unstrukturierten Vorgehen wird ein völlig offenes Interview angestrebt, bei dem der Verlauf abhängig von dem Inhalt der Konversation ist.

#### 1.4.2 Systematisches Literatur Review

Mit einem systematischen Literaturreview wird zur evidenzbasierten Identifizierung, Bewertung und Interpretation von bestehender Literatur eine wissenschaftliche Methode angewendet, mit der Fragestellungen zu einer bestimmten Thematik herausgearbeitet werden können. Mithilfe diesen Ansatzes soll die Erzielung einer Schlussfolgerung zu einem untersuchten Objekt ermöglicht werden. Die Methodik des systematischen Literaturreviews orientiert sich an den Richtlinien von Kitchenham et al.. Diese ist in drei aufeinander aufbauende Schritte gegliedert. Zu Beginn erfolgt die Planung,

---

<sup>2</sup>Iterativer Prozess zur Ermittlung von Anforderungen, die nutzerorientiert aufgestellt werden. <https://www.interaction-design.org/literature/topics/user-centered-design> Abgerufen am 08.05.2022

anschließend die Durchführung und abschließend die Dokumentation und Offenlegung der Ergebnisse [KITCHENHAM 2007].

### 1.4.3 Usability-Tests

## 1.5 Aufbau der Arbeit

Die vorliegende Master-Thesis gliedert sich nach den soeben genannten einleitenden Information im Aufbau in insgesamt zehn Kapitel. Das erste Kapitel (1) beschreibt die Motivation (1.1), welche die Intension kundtut, diese Thematik rund um IoT und Smart Home zu bearbeiten. Darauf folgend werden die Forschungsfragen (1.2), die im Rahmen der Thesis behandelt werden, erläutert. Nach der Beschreibung der Forschungsfragen wird im anknüpfenden Abschnitt (1.3) die Zielsetzung der Arbeit erläutert. Hierbei werden zusätzliche Schwerpunkte und Ziele aufgegriffen. Abschließend wird das Unternehmen, in der die Thesis geschrieben wird, hervorgehoben und deren Absichten in Verbindung mit Innovationen beleuchtet.

Das Kapitel (2) widmet sich den essentiellen und wichtigen Grundlagen dieser Arbeit. Zu Anfang wird dem Leser der Terminus des Internet of Things (2.1) offenbart, um zum Teil den Kontext im Bezug zu dieser Arbeit zu begreifen, gefolgt von einer Einführung in die Thematik des Smart Home (2.2), der Problematik der Begriffsdefinition, der historischen und kontinuierlichen Entwicklung und mit den Zielen, die mit der Verwendung einer Smart Home Lösung bewältigt werden sollen. Mit dem Verständnis der übergeordneten Begriffe, IoT und Smart Home, werden Technologien (2.3) aufgegriffen, die im Rahmen dieser Arbeit erwähnenswert sind und verwendet werden. Abschließend werden in Kapitel (2) die Softwarelösungen, Home Assistant und openHAB (2.4 & 2.5), dargestellt. Diese dienen zur Grundlage für die Evaluation als auch zur Gegenüberstellung der Lösungen in Kapitel (7) Diskussion und Evaluation.

Die theoretischen und methodischen Hintergründe sowie den Stand der Technik wird in Kapitel (3) angesprochen. Dieser Teil enthält Beschreibungen, Forschungen und aktuelle Erkenntnisse über Technologien, die im Umfeld der Smart Home Anwendungen innerhalb des IoT verwendet werden. Zudem werden in Zusammenhang der Erkenntnisse und Möglichkeiten der Technologie die Szenarien dargestellt.

Kapitel (4) befasst sich mit den Anforderungen, engl. Requirements, die für die eigentliche Konzeption relevant sind. Innerhalb dieses Kapitels wird anhand von Informationen und den umzusetzenden Szenarien die Anforderungen für die Konzeption erarbeitet. Hierbei werden aus der Praxis bekannte Verfahren verwenden, um die Anforderungen zu definieren. Mittels den zugrundeliegenden Anforderungen wird im nachfolgenden Schritt die eigentliche Konzeption dargelegt.

Nach Aufbereitung der Anforderungen durch das sogenannte Anforderungsmanagement, engl. *Requirements Engineering*, wird in Kapitel (5) das Konzept erarbeitet, welches als Grundlage für die prototypische Implementierung und Umsetzung des Konzepts dient. Das Konzept befasst sich mit den Anforderungen und setzt diese ein, um die Organisation des Systems in Komponenten, deren

Beziehungen zueinander und zur Umgebung sowie deren Prinzipien zu definieren. Zum Ende des Konzepts steht eine Architektur, die sich aus den Anforderungen und auch aus den Analysen der eigentlichen Forschungsfrage abzeichnet.

In Kapitel (6) wird die Umsetzung des Konzepts skizziert. Darunter welche Problem während der Implementierung auftraten als auch deren Lösungsfindung. Ebenso wird hier aus praktischer Sicht auf die Architektur geschaut, welche Komponenten, Bibliotheken und zusätzliche Systeme, engl. *Frameworks*, verwendet wurden. Das erzielte Ergebnis und Resultat wird abschließend zusammengefasst und neutral betrachtet.

Nach Abschluss der Umsetzung und dessen Ergebnisdokumentation befasst sich das Kapitel (7) mit der Diskussion und Evaluation. Hier findet eine Analyse des Konzepts sowie deren Umsetzung und objektive Betrachtung statt. Anschließend werden Vergleiche zwischen der Eigenentwicklung und bereits bestehender Softwarelösungen, die im Grundlagenkapitel aufgefasst werden, aufgestellt und bewertet.

Im vorletzten Teil, Kapitel (8), wird ein Fazit aus den Erkenntnissen und Ergebnissen gezogen. Dieses Schlussresümee führt nochmals die Höhepunkte sowie eine eigene Einschätzung auf.

Zum Abschluss der Thesis wird in Kapitel (9) ein Ausblick gegeben. Dieser gibt Aufschluss darüber, welche Erweiterungsmöglichkeiten es für die in dieser Thesis erfolgten Arbeit gibt und wie innovativ sich dieser Grundbaustein in Zukunft erweisen könnte.

# Kapitel 2

## Grundlagen

In diesem Kapitel werden die für diese Thesis relevanten Grundlagen geschaffen, um ein Grundverständnis und fundiertes Wissen über verwendete Technologien zu erlangen und die nachfolgende Recherche, Konzeption und Umsetzung besser verstehen zu können.

### 2.1 Internet der Dinge

Das Internet der Dinge (IdD), im Englischen Internet of Things (IoT), zählt als eines der Schlagworte in der Informationstechnologie (IT). In der Domäne des IoT bekommen Gegenstände und Objekte eine eindeutige Identität, die eine Kommunikation miteinander als auch das Entgegennehmen von Befehlen erlaubt. Mit dem Internet der Dinge lassen sich Anwendungen sowie Prozesse automatisieren und Aufgaben erledigen ohne dass von außen Eingegriffen werden muss [LUBER und LITZEL 2016]. Die Prozessautomatisierung findet sich auch im Kontext des Smart Home wieder, welches in nachfolgendem Kapitel genauer aufgegriffen wird.

In der einschlägigen Literatur gibt es für das Internet of Things keine allgemeingültige Definition, die alle Anwendungsbereiche abdeckt. Die Definitionen und Auslegungen der Interpretation unterscheiden sich je nach Anwendungsgebiet. Demnach gibt es viele verschiedene Forschungsgruppen, darunter Forscher, Akademiker, Innovatoren, Entwickler und Geschäftsleute, die den Begriff oder die zugrundeliegende Problemstellung definiert haben. Die Ursprünge jedoch sind dem Experten für digitale Innovationen, Kevin Ashton<sup>1</sup>, zuzuschreiben.

Die in der Literatur auffindbaren Definitionen verfolgen zwei Sichtweisen. Zum einen die aktive Sicht, d. h. die Daten sind von Menschen erstellt, zum anderen die passive, bei der die Daten von Dingen, darunter die Sensoren und Aktoren, erstellt werden [MADAKAM, RAMASWAMY und TRIPATHI 2015]. Eine aus dem Zusammenhang hervorgehende, aus dem wissenschaftlichen Artikel entnommene Definition ist folgende:

---

<sup>1</sup>Britischer Technologie-Pionier, Mitgründer des Auto-ID Centers am Massachusetts Institute of Technology (MIT). [https://de.wikipedia.org/wiki/Kevin\\_Ashton](https://de.wikipedia.org/wiki/Kevin_Ashton) (Abgerufen am 22.03.2022)

“An open and comprehensive network of intelligent objects that have the capacity to auto-organize, share information, data and resources, reacting and acting in face of situations and changes in the environment” [MADAKAM, RAMASWAMY und TRIPATHI 2015]

Daraus kann die Ableitung erfolgen, dass der Begriff des Internet der Dinge für die Vernetzung von Gegenständen im privaten Gebrauch, sowie von industriellen Geräten und Maschinen über das Internet, steht. Damit Geräte individuell angesprochen werden können, werden diese mit einer eindeutigen Identität, genauer einer Internetprotokoll (IP)-Adresse, im Netzwerk belegt und mit elektronischer Intelligenz ausgestattet [LUBER und LITZEL 2016]. Darüber können die Netzwerkteilnehmer über das Internet kommunizieren und Prozesse automatisiert erledigen. Die sogenannten *intelligenten Geräte* werden auch oft mit dem englischen Begriff, *Smart Devices*, betitelt.

Neben der Kommunikation der Geräte untereinander kann ebenso entweder über das Gerät selbst oder eine zentrale Schnittstelle via Internet interagiert werden. Dadurch sind Objekte und Gegenstände durch einen Benutzer von beliebigen Orten aus auch außerhalb des Netzwerks erreichbar und bedienbar. Diese Art und Weise wird auch in dem zentralen Thema des Smart Home verwendet. Die Funktion als auch die Umsetzung wird im Kapitel (2.2) näher beleuchtet.

Das Internet der Dinge ist ein elementarer Baustein der IT-Welt. Mit dem IoT wird die Vision verfolgt, eine globale Infrastruktur zu erstellen, mit der physische Objekte miteinander vernetzt werden und jeder Zeit zur Verfügung stehen. Das Internet of Things kann auch als globales Netzwerk angesehen werden, indem die Kommunikation zwischen Mensch zu Mensch, Gerät zu Mensch und Gerät zu Gerät ermöglicht wird. Viele Forschungsartikel sprechen von der Verschmelzung der digitalen und physischen Welt.<sup>2</sup> Die Vereinigung beider Welten ist die Verknüpfung physischer Objekte, die eindeutig identifizierbar sind, mit einer virtuellen Repräsentation in einer vergleichbaren Internet-Struktur.

## Gesamtbild des Internet of Things

Der folgenden Abbildung (2.1) ist zu entnehmen, welche Technologien rund um das Internet der Dinge liegen und in Verbindung damit stehen.

---

<sup>2</sup>Das Internet der Dinge – der digitale Zwilling der Welt. Kompetenzzentrum Öffentliche IT in Kooperation mit dem Fraunhofer Institut. <https://www.oeffentliche-it.de/trendsonar-iot> Abgerufen am 23.03.2022.



Abbildung 2.1: Technologische Einordnung von IoT [SIEPERMANN und LACKES 2018]

Beispielsweise ist das IoT eine wesentliche Grundlage für das Themengebiet *Big Data*. Die durch Sensoren und Aktoren erzeugten Daten können eine Grundlage für die Verwendung im Bereich *Big Data* sein. Dabei werden die Datenmengen gespeichert und mithilfe von Mustern und Herangehensweisen des Big Data<sup>3</sup> analysiert. Big Data ist kein Bestandteil dieser Arbeit und wird demnach nicht weiter ausgeführt. Das Beispiel dient lediglich zu Veranschaulichung und Interpretation der oben aufgeführten Abbildung (2.1).

Eine allgemeine exemplarische Skizzierung eines Systems, welches nach dem IoT-Konzept aufgebaut ist, kann der folgenden Abbildung (2.2) entnommen werden:

<sup>3</sup>Definition und Funktionsweise von Big Data. <https://www.oracle.com/big-data/what-is-big-data/> Abgerufen am 25.03.2022



Abbildung 2.2: Exemplarische Darstellung eines IoT-Systems [GILLIS 2022]

Hierbei werden die jeweiligen Komponenten verdeutlicht, die in einem System zu finden sind. Mit den *IoT-Geräten*, darunter beispielsweise Sensoren und Aktoren, findet die Datenerzeugung statt. Mit dem dahinterstehenden *Gateway* werden die aus den Geräten erzeugten Daten gesammelt und an zentraler Stelle an die Cloud gesendet. Nach der Datensammlung können diese an verschiedene Komponenten zur weiteren Verarbeitung gesendet werden, die durch die Visualisierung über das Smartphone stattfinden kann oder zur Durchführung von Prozessanalysen dient. Ebenso können die Daten an weitere Backend-Systeme zur Weiterverarbeitung übermittelt werden.

## Anwendungsbereiche des IoT

Grundlegend kann im Bereich des Internet of Things zwischen zwei Anwendungsbereichen unterschieden werden. Dies ist zum einen der private Bereich und zum anderen der industriellen. Der private Bereich deckt hauptsächlich die Thematik rund um den Gebrauch von Alltagsgegenständen ab und deren Vernetzung untereinander zur komfortableren und intelligenteren Nutzung der Geräte. Darin inbegriffen sind Gebäudeautomatisierungen und Ereignissteuerungen über das Internet. Diese Funktionen sind Hauptbestandteil des Smart Home-Konzeptes, welches in Abschnitt (1.5) genauer aufgegriffen wird.

Der industrielle Bereich beschäftigt sich mit der Vernetzung von Maschinen und Anlagen miteinander, so dass sich ganze industrielle Prozesse automatisieren lassen und so die Effizienz der Prozess- und Produktionsabläufe gesteigert werden. Die Nutzung des IoT im industriellen Bezug ist



ein elementarer Bestandteil der heutigen *Industrie 4.0*<sup>4</sup>, auch bekannt als *Industrial Internet of Things*, dt. *Industrielles Internet der Dinge (IIoT)*.

Im Rahmen dieser Arbeit wird zwischen den beiden Anwendungsbereichen differenziert und den Fokus auf den privaten Bereich gelegt.

Mit dem Internet der Dinge-Ansatz gibt es zwei Paradigmen, die in Kombination als auch alleinstehend Anwendung finden. Diese werden im folgenden Abschnitt kurz erläutert.

## Edge und Cloud Computing

Bei den beiden Paradigmen handelt es sich zum einen um Edge Computing und zum anderen um Cloud Computing. Ziel beider Ansätze ist das Verwalten von Daten, bzw. das Arbeiten mit erzeugten Daten.

Das Edge Computing verfolgt einen *dezentralen Ansatz*, bei dem die Berechnung und Erhebung der Daten direkt auf dem Gerät durchgeführt wird. Dies bedeutet, dass jedes Gerät über eine eigene Intelligenz verfügt, bei der Daten direkt nach der Erzeugung verarbeitet und gespeichert werden können. Zu einem späteren Zeitpunkt können in Form einer Datenbündelung oder einer Vorauswahl die Informationen an ein Rechenzentrum weitergegeben werden.

“Edge computing is different from traditional cloud computing. It is a new computing paradigm that performs computing at the edge of the network. Its core idea is to make computing closer to the source of the data [...]” [CAO u. a. 2020]

Bei Cloud Computing handelt es sich grundlegend um Bereitstellungen von Computerdiensten, Kapazitäten, Ressourcen und der Rechenleistung über das Internet, die nicht in einem eigenen Rechenzentrum verwaltet und betrieben werden müssen. Die Verwaltung wird dem Anbieter des Cloud Computing überlassen. Hauptsächlich steht dabei die flexible Ressourcennutzung, Skalierung und Verteilung von Rechenleistungen im Vordergrund. In Zusammenhang mit den verfügbaren Ressourcen werden auch verschiedene Modelle, Cloud Typen und diverse Dienste angeboten<sup>5</sup>. Diese werden im Rahmen der Arbeit nicht weiter ausgeführt.

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” [MELL und GRANCE 2011]

Mit dem Cloud Computing wird im Vergleich zum Edge Computing ein zentraler Ansatz verfolgt, bei dem die erzeugten Daten von den Aktoren und Sensoren direkt an eine zentrale Stelle gelangen. Von dort aus findet die Verarbeitung und Analyse der Daten statt.

Eine Kombination beider Ansätze vereint deren Vorteile und ist als *Hybrid Cloud* bekannt. Diese Konstellation ist in den meisten Architekturen wiederzufinden.

<sup>4</sup>Definition und Beschreibung der Industrie 4.0 <https://www.plattform-i40.de/IP/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html> Abgerufen am 25.03.2022

<sup>5</sup>Einblick in die Cloud-Umgebung von dem Anbieter Microsoft. <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/> Abgerufen am 28.03.2022

### 2.1.1 Paradigmen und Kommunikationsmodelle

Damit eine umfassende Grundlage im Bereich IoT geschaffen wird, behandelt folgender Abschnitt Paradigmen und Kommunikationsmodelle, die in dieser Umgebung verwendet werden. Für die Darstellung dieser Modelle wird eine Literaturquelle genutzt, die ein paar der gängigsten und prägnantesten Architekturen und Modelle abgedeckt. Die Repräsentation der Interaktionsparadigmen sind den Schaubildern (2.3) zu entnehmen.

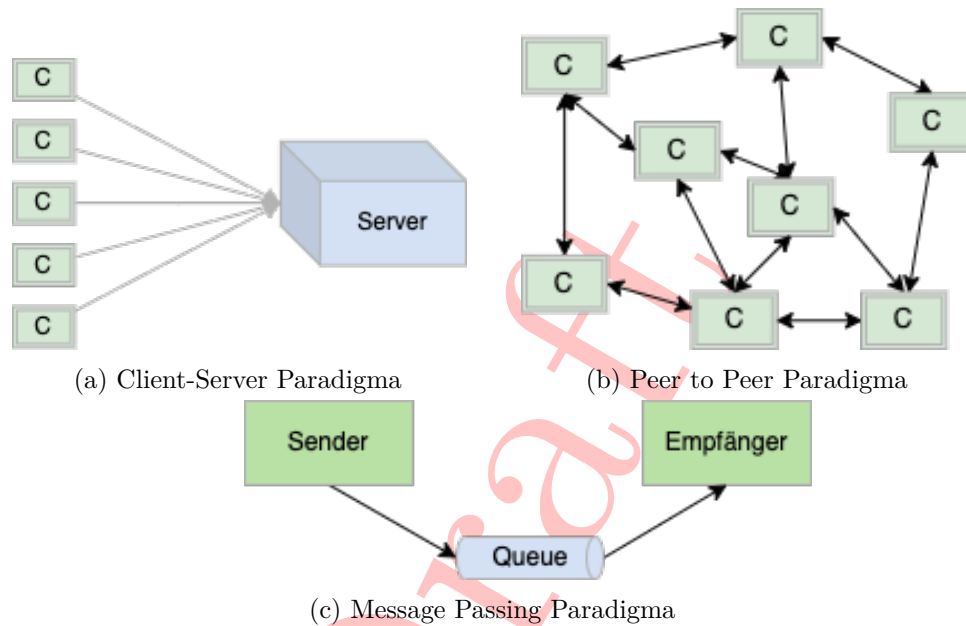


Abbildung 2.3: Client-Server, Peer-to-Peer und Message Passing Interaktionsparadigma [MINERVA, BIRU und ROTONDI 2015]

Name	Beschreibung
Client-Server (2.3a)	Basierend auf einer sehr einfachen Interaktion zwischen Clients und dem Server. Ein Client sendet eine Anfrage an den Server und erwartet dementsprechend eine Antwort.
Peer to Peer (P2P) (2.3b)	Gegensatz zu Client-Server Konzept. Hierbei können Geräte sowohl als Client Dienste abfragen, als auch als Server Dienste anbieten.
Message Passing (2.3c)	Basierend auf einer einfachen Organisation in der ein Absender eine Nachricht mittels einer Warteschlange (Queue) an einen Empfänger weiterleitet.

Tabelle 2.1: Interaktionsparadigmen nach [MINERVA, BIRU und ROTONDI 2015]

Eine Analyse und detailliertere Darstellung der jeweiligen Ansätze kann der Ausarbeitung, aus der die Paradigmen verwendet wurden, entnommen werden.

Im Zusammenhang mit Architekturen, die in dem Bereich IoT Anwendung finden, und Interaktionsparadigmen wird häufig zur Unterstützung verschiedener Interaktionsmodelle auf Protokolle gesetzt, die Daten und Primitiven transportieren. Diese Kommunikationsmodelle sind in zwei Bereiche, die ihre Anwendung in Machine to Machine (M2M) finden, kategorisiert:

- Simple Object Access Protocol (SOAP)<sup>6</sup>: Dabei handelt es sich um ein Standardprotokoll, das dafür entwickelt wurde, um verschiedene Programmiersprachen auf verschiedenen Plattformen untereinander kommunizieren zu lassen.
- Representational State Transfer (REST)<sup>7</sup>: Dabei handelt es sich um eine Reihe von Architekturprinzipien, die auf die Anforderungen leichtgewichtiger Webdienste und mobiler Anwendungen abgestimmt sind. Anfragen, die über diese Form gesendet werden, können in mehreren Formaten beantwortet werden, darunter beispielsweise Hypertext Markup Language (HTML), Extensible Markup Language (XML) und JavaScript Object Notation (JSON).

Beide verfolgen das Ziel der Datenübermittlung zwischen Web-Anwendungen über eine Schnittstelle, das sogenannte Application Programming Interface (API). REST Architekturen werden bei einem IoT-Szenario aufgrund ihrer Simplität und ihres Komforts bevorzugt. [MINERVA, BIRU und ROTONDI 2015]

### 2.1.2 Historische Entwicklung

Die ersten Konzepte zu dem heute bekannten IoT liegen schon einige Jahrzehnte zurück. Die historische Definition des Internet of Things wurde im Jahr 1999 von Kevin Ashton geprägt. Von der Definition bis hin zum erstmaligen Einsatz, bzw. zur Umsetzung eines Konzepts vergingen noch ein paar Jahre. Ein kurzer Einblick in die chronologische Entstehung des IoT ist der folgende Tabelle zu entnehmen:

---

<sup>6</sup>Detailliertere Definition des Begriffs SOAP. <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest> Abgerufen am 29.03.2022

<sup>7</sup>Detailliertere Definition des Begriffs REST. <https://www.redhat.com/en/topics/integration/whats-the-difference-between-soap-rest> Abgerufen am 29.03.2022

Jahr	Industrielle Beteiligung und Zuordnung
1970	Der erste Vorschlag zu miteinander verbundenen Geräten und Maschinen
1990	John Romkey und Simon Hackett entwickelten den ersten Toaster, der über das Internet an- und ausgeschaltet werden konnte. <sup>8</sup>
1995	Das Unternehmen Siemens leitete das erste Mobilfunkmodul ein, welches für die Kommunikation zwischen Maschinen zuständig ist (M2M-Technologie).
1999	Die Definition des Begriffs IoT von Kevin Ashton wurde veröffentlicht und von der Gesamtheit akzeptiert, als er bei Procter & Gamble (P&G) an den sogenannten Radio-Frequency Identification (RFID) Chips arbeitete. RFID-Chips verwenden elektromagnetische Felder, die an Objekte angebracht sind, um diese identifizieren und verfolgen zu können. Ein solches System besteht aus einem Funktransponder, einem Funkempfänger und -sender.
2004 - 2005	Der Begriff wurde in angesehenen Publikationen verwendet, darunter dessen von Boston Globe und The Guardian. International Telecommunication Union (ITU) veröffentlichte deren ersten Bericht über das Thema.
2008 - 2011	Die Definitionen und Publikationen fanden erste Anwendungen in praktischen Umsetzungen. Gartner Inc. nahm den Begriff in ihre Recherche-Arbeiten mit auf.

Tabelle 2.2: Historische Entwicklung vom Internet der Dinge [DURGA, PROF und KUMAR 2020]

Mittlerweile ist das Internet der Dinge ein fester Bestandteil der Industrie, ebenso im privaten Umfeld. Viele Szenarien und Konzepte werden erarbeitet und umgesetzt. Die Kommunikation von Maschinen untereinander ist in der heutigen Zeit fester Bestandteil der Industrie und wird immer weiter ausgebaut. Forschergruppen sind daran interessiert, mehrere Gebiete und Anwendungsbereiche innerhalb des IoT zu erschließen. Stark vertreten ist dabei das Fraunhofer Institut für Integrierte Schaltungen (IIS)

### 2.1.3 Ziele von IoT

Die Definition von dem Internet der Dinge gibt die Richtung an, in die sich die Zielsetzung von IoT bewegt. Davon ist die Intention abzuleiten, dass die Lücke zwischen der realen und der virtuellen Informationswelt so gering wie möglich gehalten, bzw. zunehmend minimiert wird. Dieser Informationsbedarf oder auch die Informationslücke besteht, da in der realen Welt Objekte, Dinge und Gegenstände einen bestimmten Zustand vorzeigen. Ein Beispiel dazu ist „die Temperatur liegt bei 35 Grad Celsius“ oder „das Licht ist an und hat 75 % Helligkeit“. Diese Informationen sind in der virtuellen Welt nicht ohne weiteres verfügbar. Das Bestreben demnach, welches auch zum Großteil im Bereich Smart Home abgedeckt wird, ist, dass viele reale Gegenstände ihre Zustandsinformationen für die Weiterverarbeitung in der virtuellen Welt, beziehungsweise im Netzwerk zur Verfügung stellen. Diese Informationen können beispielsweise Aussagen zu der aktuellen Nutzung, zur bestimmten Positionierung im Raum und über die Alterung eines Gerätes geben. Diese Zustandsinformationen können dazu beitragen, dass die Nutzung des Gegenstandes vom Anwender ausgewertet und so

optimiert werden kann, z.B. durch die Erkennung eines Defekts oder einer Automatisierung, damit die Ressource effizient genutzt wird.

## 2.2 Smart Home

Smart Home, im Deutschen *“intelligentes Zuhause“*, ist eine Domäne des IoT. Diese Rubrik der Anwendung widmet sich überwiegend dem Gebrauch von sämtlichen Haushaltsgeräten und -einrichtungen. Ein kleiner Ausschnitt solcher Nutzgegenstände sind unter anderem Lampen, Kontaktsensoren, Thermostate, Service-Roboter, Staubsauger-Roboter, Kühlschränke Geräte rundum die Haussicherheit.

Unter dem Oberbegriff Smart Home ist eine Weise zu verstehen, mit der die Erhöhung der Wohn- und Lebensqualität erzielt, Energienutzung unter Verwendung vernetzter und fern steuerbarer Geräte effizienter gestaltet, Sicherheit gesteigert und Abläufe verschiedener Prozessschritte automatisiert werden kann.

Der Begriff *intelligentes Zuhause* wird verwendet, wenn die Haustechnik und Haushaltsgeräte untereinander vernetzt sind. Die Definition im Deutschen Gebrauch, welche nach (Strese et al. 2010) in der Untersuchung im Rahmen der wissenschaftlichen Begleitung zum Programm *„Next Generation Media (NGM) des Bundesministeriums für Wirtschaft und Technologie“* aufgegriffen wird, lautet wie folgt:

„Das Smart Home ist ein privat genutztes Heim (z.B. Eigenheim, Mietwohnung), in dem die zahlreichen Geräte der Hausautomation (wie Heizung, Beleuchtung, Belüftung), Haushaltstechnik (wie z.B. Kühlschrank, Waschmaschine), Konsumelektronik und Kommunikationseinrichtungen zu intelligenten Gegenständen werden, die sich an den Bedürfnissen der Bewohner orientieren. Durch Vernetzung dieser Gegenstände untereinander können neue Assistenzfunktionen und Dienste zum Nutzen des Bewohners bereitgestellt werden und einen Mehrwert generieren, der über den einzelnen Nutzen der im Haus vorhandenen Anwendungen hinausgeht.“ [STRESE u. a. 2010]

Eine vergleichbare Definition wurde zu späterem Zeitpunkt durch eine Literaturrecherche publiziert. Diese beschreibt die zugrundeliegende Thematik weniger aus Anwendersicht sondern widmet sich vielmehr dem System und der Konnektivität.

„A smart home is a place with heterogeneous systems to many front devices with the support of embedded information and communication architectures[...]“ [BALAKRISHNAN, VASUDAVAN und MURUGESAN 2018]

Den beiden Definitionen ist zu entnehmen, dass die Kernaussage eine ähnliche ist, es jedoch in Büchern, Fachartikeln, Publikationen an Universitäten und in den verbreiteten Medien bis heute keine durchgängige gibt. Aus der einschlägigen Literatur wird ersichtlich, dass viele Synonyme für die Benennung der Thematik verwendet werden, darunter beispielsweise [STRESE u. a. 2010]:

- Connected Home
- Elektronisches Haus
- Intelligentes Haus (engl. Smart House)
- Smart Living
- Home of the Future

Eine elementare Information im Zusammenhang dieser Arbeit ist die Verwendung des Begriffs *intelligentes Büro*, die ebenso in den Kontext des Smart Home gehört. Hierbei wird lediglich die Räumlichkeit im unternehmerischen Jargon verwendet, die ebenso eine Grundlage für die Verwendung von Komponenten des Smart Home bietet. Im Abschnitt (2.2.1) wird nochmals konkret darauf eingegangen.

Es wird deutlich, dass die Verwendung des Begriffs als auch die zugrundeliegenden technischen Verfahren weiträumig einsetzbar sind und deshalb die Begriffsdefinition nicht eindeutig festgehalten werden kann.

### Teilsysteme des Smart Home

Der primäre Anwendungsbereich des Smart Home ist die Automatisierung häuslicher Prozesse. Dadurch sollen dem Nutzer in vielerlei Hinsicht Aufwände erspart und Informationen zentralisiert angezeigt werden. Die Hausautomatisierung umfasst eine Menge von Teilsystemen. Ein Ausschnitt dieser Teilsysteme ist der folgenden tabellarischen Auflistung zu entnehmen:

Segment	Beschreibung
Licht	Beleuchtung, Lichtmanagement/Szenarien, Storen/Rollos
Zutritt	Zutrittskontrolle, Klingelanlage, Schlösser, Anwesenheits- und Bewegungserfassung
Überwachung	Technische Alarmer: Feuer, Rauch, Gas; Intrusion: Glasbruchmelder, Video; Babyphon, Urlaubswachschutz
Notfall	Sprinkleranlage, unabhängige Stromversorgung, Fluchtwegsystem
Metering	Verbrauchszähler für Strom, Gas, Wasser, Wärme, uvm.
Konsumelektronik	TV, Internet, Smartphones, Tablets, Spielekonsolen etc.
Hausgeräte	Kühlschrank, Waschmaschine, Staubsauger, Service-Roboter; Hausgeräte-monitoring, -diagnostik, und -fernbedienung
Heimlogistik	Einkaufs- und Speiseplanung, häusliche Dienste
Hobby	Haustiersversorgung, Aquarienmanagement, etc.
Mobilität	PKW mit Diagnostik, Navigationssystem mit local based services, Info-/Entertainmentangebote etc.

Tabelle 2.3: Teilsysteme des Smart Home [STRESE u. a. 2010]

Eine mögliche Vernetzung und auch Verwendung solcher Komponenten wird in folgender Abbildung (2.4) skizziert. Diese Grafik dient als grobe Übersicht potentieller Anwendungsszenarien, repräsentiert jedoch nicht alle Möglichkeiten der Anwendung. Darüber hinaus gibt es weitaus mehr



17

## Einordnung von Smart Home in das Internet der Dinge

Im Konsumentenmarkt wird die Technologie des IoT in Produkten eingesetzt, die das Konzept des Smart Home verfolgen. Diese beinhalten Haushaltsgeräte und -ausstattung, wie beispielsweise Thermostate, Sensoren, Sicherheitssysteme und Lampen, die mehrere Systeme und Übertragungstechnologien unterstützen. Dazu zählen unter anderem Plattformen, darunter Google Nest, Apple HomeKit und Amazon Alexa uvm., ebenso die Übertragungstechnologien, die im Abschnitt (2.3) aufgegriffen werden. Weitere Beispiele sind der Tabelle (2.3) zu entnehmen.

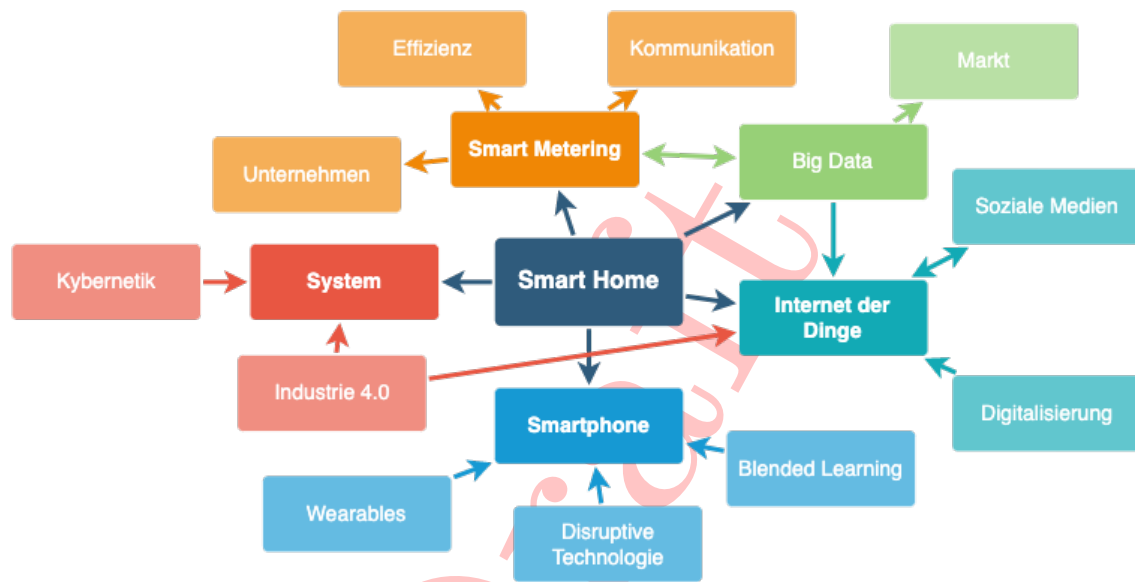


Abbildung 2.5: Technologische Einordnung von Smart Home in Verbindung zu IoT [BENDEL 2021]

Die Abbildung (2.5) zeigt die Verbindungen, als auch die Beziehungen von Smart Home zu anderen Technologien. Hier wird deutlich, dass im Bereich des Smart Home viele Fragestellungen thematisiert werden können, die andere Themenbereiche tangieren. Mit inbegriffen ist zum Beispiel die Komponente Smartphone, da dieses genutzt wird, um als Fernbedienung zu fungieren und Prozesse und Automationen anzeigen und überwachen zu können. Der Bereich des Smart Metering<sup>9</sup> deckt die Messung von Verbrauchsdaten ab. Dabei handelt es sich um intelligente Messsysteme, die Daten zum Verbrauch, darunter Strom, Gas und Wasser, erheben und diese von den jeweiligen Anbietern zur Rechnungsstellung genutzt werden. Ein Beispiel dazu ist ein digitaler intelligenter Stromzähler mit direkter Kommunikationsmöglichkeit zum Anbieter selbst.

Der Aufbau eines Smart Home ist architektonisch ähnlich zu dem Grundprinzip einer IoT-Lösung. Die Veranschaulichung des zugrundeliegenden Aufbaus ist der Abbildung (2.2) zu entnehmen. Um die Bezugspunkte zu IoT und die Einordnung zu untermauern, wird in folgendem Abschnitt auf die Funktionsweise von Smart Home eingegangen.

<sup>9</sup>Smart Metering ist das computergestützte Messen, Ermitteln und Steuern von Energieverbrauch und -zufuhr. <https://wirtschaftslexikon.gabler.de/definition/smart-metering-53998> Abgerufen am 06.04.2022



## Funktionsweise eines Smart Home

Ein Smart Home System besteht aus mehreren Teilsystemen, die in der Regel aus verschiedenen Komponenten bestehen. Wichtige Elemente eines grundlegenden Aufbaus sind die Endgeräte, die sogenannten Aktoren, Eingabegeräte, Sensoren, Gateway und die Vernetzung über Funk, Kabel oder Stromnetz. Die Endgeräte sind die Ausgabegeräte, die über die intelligente Steuerung angesprochen werden können. Darunter zählen zum Beispiel LED-Lampen, Rolläden, Lüftungsanlagen, Lautsprecher, Fernseher, Waschmaschinen und jegliche Arten von Service-Robotern. Eingabegeräte sind die Schnittstelle zwischen der Interaktion des Nutzers und des Smart Home Systems. Das können Wandschalter, Touchdisplays, Fernbedienungen, Smartphones und Regler sein. Mithilfe dieser Schnittstelle können Zustände und Aktionen an den Endgeräten ausgelöst werden. Bei einer fehlenden Verbindung zwischen den Steuerelementen sind diese trotzdem noch über direkte Schaltbefehle steuerbar. Damit die Zustände ebenso digitalisiert werden können und dem System zur Verfügung stehen, werden Sensoren benötigt. Diese greifen die physikalischen oder elektronischen Eigenschaften des Endgerätes ab, um die Zustände zu ermitteln. Das Gateway repräsentiert die zentrale Steuereinheit, auf dem die Sensordaten eingehen und die Sendung von Befehlen an die Aktoren stattfindet. Ebenso ermöglicht das Gateway die Kommunikation der Endgeräte und Sensoren untereinander. Eine mögliche Internetverbindung zwischen dem Gateway und einer zentralen Plattform, die über die Cloud erreichbar ist, kann ebenso hergestellt werden. Je nach Gerät kann über das Gateway auch eine direkte Steuerung einzelner Elemente stattfinden. Das letzte Element, die Vernetzung, ist für die Verbindung aller Elemente zuständig. Hierfür kommen verschiedene Protokolle, die unter anderem in Abschnitt (2.3) beschrieben werden, per Funk, Kabel oder Stromnetz zum Einsatz. Ein exemplarischer Aufbau der Komponenten ist der folgenden Abbildung zu entnehmen:



Abbildung 2.6: Aufbau und Funktionsweise einer Smart Home Infrastruktur

### 2.2.1 Smart Office - Intelligentes Büro

Ein Smart Office, im Deutschen intelligentes Büro, ist ein konkreter Anwendungsbereich des Smart Home. Während das Smart Home vorwiegend von der Verbesserung der Lebensqualität, der Einsparung von Energiekosten, der Optimierung des Wohlbefinden im privaten Wohnraum und der Erleichterung des Alltags des Nutzers handelt, zielt das Smart Office darauf ab, das Wohlbefinden und die Produktivität der Mitarbeiter zu steigern. Weitere wichtige Punkte sind neben den beiden aufgeführten auch die Erhöhung der Sicherheit der Büroräume und der Nachhaltigkeit durch z.B. Reduzierung des Stromverbrauchs. In diesem Zusammenhang spielt die Automation von Gebäudemanagement- und die damit verbundenen Sicherheitssysteme eine große Rolle.

Viele Funktionalitäten und Vorteile sind deckungsgleich zu denen des Smart Home. Durch intelligente Geräte, Aktoren und Sensoren ist die Steuerung von Licht, Raumtemperatur etc. möglich. Hervorzuheben ist unter anderem die mögliche Zutrittskontrolle per Gesichtserkennungssysteme, die an zentraler Stelle mit weiteren Komponenten kommunizieren und beliebige zusätzliche Schritte einleiten können. Ebenso zu nennen ist der Aspekt in Richtung Gebäudesicherheit und deren Ausbau, denkbar in Form von intelligenten Feuer- und Wassermeldern oder weiteren Maßnahmen, die durch Automationen abgebildet werden können.

In unserer heutigen Zeit ist die Nachhaltigkeit und Ressourceneffizienz von immer größerer Bedeutung. Automatische Abschaltungen von Beleuchtungen, Kaffeemaschinen, Monitoren oder anderen Geräten durch Präsenz- und Bewegungsmelder oder Remotesteuerungsoptionen können Ressourcen- und Energieeinsparungen umsetzen.

### 2.2.2 Historische Entwicklung

Im Bereich des Smart Home wurde im Jahr 1975 die erste Netzwerktechnologie für Hausautomationen präsentiert und vermarktet. Bekannt wurde diese unter dem Namen *xX0*<sup>10</sup>. Dabei handelt es sich um ein stromleistungs-basiertes Netzwerkprotokoll zur Gebäudeautomation. Die Schaltsignale werden über die Hausinstallation, das Stromnetz des Hauses, transportiert. Eingeführt wurde die Technologie von dem Unternehmen Busch-Jaeger unter dem Namen *Timac X10*. Es zeichnete sich durch die einfache Konfiguration und dessen interessanten Funktionen zu diesem Zeitpunkt aus [ASCHENDORF 2014]. Die Weiterentwicklung des Systems, *Timac X10*, fand im Jahre 1998 statt, indem von Busch-Jaeger ein neues Produkt Namens *Powernet EIB* in Deutschland eingeführt wurde. Dieses baute ebenso auf dem grundlegenden Netzwerkprotokoll X10 auf und fügte sich nahtlos in den europäischen Installationsbus (EIB/KNX) ein [BUSCH-JAEGER 2021]. KNX ist ein Bussystem zur Gebäudeautomation, welches basierend auf dem EIB weiterentwickelt wurde. Es zählt heute noch zu den kabelgebundenen Standards. Im Jahre 2001 eröffnete die Fraunhofer IMS in Kooperation mit der Universität Duisburg-Essen die Fraunhofer-inHaus-Forschungsanlage [FRAUNHOFER 2021]. Innerhalb dieser Institution erforschten, entwickelten, testeten und demonstrierten Dienstleister, Hersteller und Nutzer mit dem Fraunhofer-Institut und der Universität neue Systemlösungen und weitere Produktkomponenten sämtlicher Arten im Bereich des Wohnens. Anfang des Jahres 2005 wurde die deutsche Telekom in dem Smart Home Bereich aktiv und präsentierte der Öffentlichkeit

---

<sup>10</sup>[https://de.wikipedia.org/wiki/X10\\_\(Protokoll\)](https://de.wikipedia.org/wiki/X10_(Protokoll)) - X10 Protokoll Erklärung. Abgerufen am 06.04.2022

ein vollständig vernetztes, „intelligentes“ Musterhaus in Berlin, das sogenannte *T-Com-Haus*. Anfang des Jahres 2012 wurde das Bundesministerium für Wirtschaft und Klimaschutz (BMWK) im Sektor des Smart Home aktiv. Seitdem fördert die Behörde das „*Zertifizierungsprogramm Smart Home + Building*“, bei dem Forschungen im Bereich des Smart Home von Vertretern akademischer Einrichtungen und Industrieunternehmen durchgeführt werden. Ziele dabei sind unter anderem die Erstellung von Standards und Prüfsiegeln für systemübergreifende Interoperabilität der Geräte eines Smart Home [V. 2013]. 2013 präsentierte die deutsche Telekom neue Lösungen in einem weiteren Musterhaus in Darmstadt. Darin finden sich mehrere Komponenten und Vernetzungen als im Berliner Modell [BRAJKOVIC 2014], so die Nutzung verschiedener Funkstandards, die es ermöglichen, intelligente Geräte unterschiedlichster Hersteller zu konfigurieren und über ein Smartphone, Tablet oder Computer zu kontrollieren und zu steuern [BRAJKOVIC 2014]. Die Auswirkungen der rasanten Entwicklung dieser Technologie trieben das Interesse in die Höhe, sodass 2014 und 2015 die Technologie das Hauptthema der Internationale Funkausstellung (IFA) bei vielen Ausstellern war. Bis heute wird in dem Segment des Smart Home geforscht und immer weitere Funktionalitäten entwickelt. Der Trend der Nutzung von intelligenten Geräten ist weiter zunehmend. Die Marktsituation und der aktuelle Ist-Stand wird in dem Abschnitt Marktanalyse (4.1) nochmals aufgegriffen und mit Statistiken und Umfragen belegt.

### 2.2.3 Ziele von Smart Home

Die Ziele der intelligenten Vernetzung sind in erster Linie diejenigen, die die großen Domänen abdecken [LUBER und LITZEL 2019]:

- Komfort (Steuerung, Fernbedienung, Delegation und Automationen)
- Energie- und Kosteneffizienz
- Sicherheit (Überwachung)

Allgemein lässt sich der Komfort durch die entfernungsunabhängige, bequeme Steuerung von Licht, Heizung, Unterhaltungselektronik, Service-Robotern und vielen weiteren Geräten steigern. Hierfür spielen beispielsweise Smartphones, zentrale Steuerungsgeräte und Tablets eine wichtige Rolle. Diese fungieren ähnlich wie eine herkömmliche Fernbedienung. Es können ebenso Prozesse automatisiert oder durch ein bestimmtes vorgegebenes Ereignis, welches eintreten kann, angestoßen werden. Mit diesen Automationen können gewünschte Wohnbedingungen zu bestimmten Anlässen und Zeiten erzeugt werden. Beispielsweise kann die Heizung vor Eintreffen reguliert werden, um die optimale Temperatur zu erreichen. Die Steuerung von Geräten spielt gleichermaßen eine Rolle in den Bereichen der Energie- und Kosteneffizienz und der Sicherheit.

Durch das Verbinden mehrerer Komponenten mithilfe einer Automation können Energiekosten verringert werden, wenn bspw. ein Temperaturabfall durch ein offenes Fenster von Sensoren gemessen wird und diese ein Signal übermitteln. Dementsprechend kann die Heizung für den Zeitraum ausgeschaltet oder reguliert werden, um überflüssige Wärmeerzeugung zu vermeiden. Weitere Einsparungen können durch das Ausschalten von aktuell nicht benötigten Ressourcentypen, wie z.B. Elektrogeräten oder LED-Lampen, vorgenommen werden.

Zuletzt ist die Erhöhung der Sicherheit und das Überwachen von Eingängen und Fenstern zu nennen. Die Bewegungsmelder und Sensoren können Aktivitäten registrieren und sind Auslöser für die Weitergabe der Information an den Eigentümer. Zusätzlich können mit Kameras Objekte überwacht werden, beispielsweise ein Haustier, das kurze Zeit alleine zuhause ist oder es können Einbruchversuche registriert und gemeldet werden. Auch eine Alarmierung bei Flutung des Kellers oder bei Hausbrand sind denkbar.

Die Geräte im Smart Home erreichen weitestgehend diese Ziele und werden stetig weiterentwickelt, verbessert und um neue Funktionalitäten ergänzt.

## 2.3 Technologien

Die bereits angesprochene Kommunikation und Vernetzung zwischen Geräten basiert im Allgemeinen auf diversen Protokollen. Um diese Datenbewegung und Kommunikation besser verstehen zu können, werden im Folgenden bekannte Protokolle erwähnt und aufgeführt und eines der meist verwendeten näher betrachtet. Um einen Vergleich herzustellen, wird ein vergleichbares Protokoll betrachtet. Diese werden dann zum Abschluss gegenübergestellt.

### 2.3.1 Übertragungsmethoden

Allgemein gibt es im Bereich des Smart Home mehrere Methoden und Möglichkeiten, die Objekte miteinander zu vernetzen. Dazu gehören Protokolle über Bluetooth, Ethernet, WLAN, Bussysteme, Funk und Stromleitung. Jeder Hersteller setzt diese abhängig von seinem Produkt passend ein. Proprietäre Systeme funktionieren nur über eine Übertragungsmethode. So erzwingen die Hersteller die Nutzung einer Produktlinie, bzw. den Kauf von Elementen eines einheitlichen Systems. Geräte, die die Möglichkeiten besitzen, über mehrere Protokolle zu kommunizieren, sind flexibler einsetzbar und mit mehreren Plattformen und Geräten kompatibel. Grundlegend werden mit diesen Übertragungsmethoden Netzwerke erstellt, über das die Geräte in einem Smart Home kommunizieren können.

Die Auflistung der zum aktuellen Zeitpunkt am meist verwendeten Übertragungsmethoden dient lediglich als Einblick in die verschiedensten Ausprägungen der Kommunikationstechnologie. Demnach wird im Rahmen dieser Arbeit das Thema nicht ausführlich vertieft. Die populärsten Übertragungsmethoden werden in folgender Tabelle aufgelistet<sup>11</sup>:

---

<sup>11</sup>Auswahl von derzeit verwendeten Übertragungsmethoden. [https://de.wikipedia.org/wiki/Smart\\_Home#Übertragungsmethoden](https://de.wikipedia.org/wiki/Smart_Home#Übertragungsmethoden) Abgerufen am 02.04.2022

Technologie	Übertragung	Frequenzbereich (Funk)	Proprietär
ZigBee	Funk	2,4 GHz, 868 MHz	Nein
Z-Wave	Funk	868 MHz	Nein
HomeMatic	Funk / Datenleitung	868,3 MHz	Ja
KNX	Funk / Strom- und Datenleitung	868 MHz / -	Nein / Ja (Datenleitung als Gesamtsystem)
Wi-Fi / WLAN	Funk	2,4 - 5 GHz	Nein
Bluetooth	Funk	2,4 GHz	Nein
io-homecontrol	Funk	868-870 MHz	Ja

Tabelle 2.4: Übertragungsmethoden des Smart Home

### 2.3.2 MQTT

Das Message Queue Telemetry Transport (MQTT)-Protokoll ist eines der ältesten offenen Netzwerk- und Nachrichtenprotokolle der M2M-Kommunikation. Dies wurde 1999 von IBM Mitarbeiter Andy Stanford-Clark<sup>12</sup> und von Cirrus Link Solutions Mitarbeiter Arlen Nipper<sup>13</sup> entwickelt. Die Technologie ermöglicht die Übertragung von Messdaten, sogenannten Telemetriedaten, in Form von Nachrichten zwischen Maschinen und Geräten. Die erzeugten Messdaten durch beispielsweise Sensoren und Aktoren können durch ihre minimale Größe und die kompakte Form des Protokolls in einem kleinen Datenpaket auch bei hoher Verzögerung oder bei beschränktem Netzwerk übertragen werden [NAIK 2017]. MQTT ist ein klassisches Client-Server-Protokoll, welches nach dem *Publish/-Subscribe* Kommunikationsmodell entwickelt wurde. Ein MQTT-Client veröffentlicht Nachrichten an einen MQTT-Server, den sogenannten MQTT-Broker. Diese können von anderen Clients abonniert oder auf dem Broker für zukünftige Abonnements aufbewahrt werden. Jede erzeugte Nachricht wird an eine Adresse veröffentlicht, die als Thema, im Englischen *Topic*, bezeichnet wird [NAIK 2017]. MQTT-Clients können mehrere Topics abonnieren und erhalten jede Nachricht, die an das jeweilig abonnierte Topic gesendet wird.

Die Leichtigkeit des Protokolls ermöglicht es, die Nachrichten bei eingeschränkter Netzwerkverfügbarkeit zu übermitteln. Ausschlaggebend dafür ist das binär-basierte Protokoll, welches normalerweise einen festen Header von zwei Bytes mit kleinen Nachrichtennutzlasten von maximal bis zu einer Größe von 256 MB [NAIK 2017] enthält. Grundsätzlich ist MQTT auf der Basis des Transportprotokoll Transmission Control Protocol (TCP) aufgebaut und nutzt zur Verstärkung der Sicherheit die Transport Layer Security (TLS)/Secure Sockets Layer (SSL)-Verschlüsselung. Dadurch sind Client und Broker mit ihrer Kommunikation verbindungsorientiert. Die Anwendung

<sup>12</sup>Informationen zu Herrn Stanford-Clark <https://stanford-clark.com> Abgerufen am 12.04.2022

<sup>13</sup>Informationen zu Herrn Nipper <https://www.inductiveautomation.com/resources/podcast/the-coinventor-of-mqtt-arlen-nipper-from-cirrus-link-solutions> Abgerufen am 12.04.2022

von MQTT im Bereich des Smart Home zeichnet sich durch die Möglichkeit aus, große Netzwerke mit vielen kleineren Geräten, die von einem Backend-Server, bzw. dem Backend-System überwacht und gesteuert werden müssen, zu betreiben. Dennoch ist es nicht für Multicast-Daten oder Übertragungen von Gerät zu Gerät ausgelegt. Die Nutzbarkeit von MQTT ist aufgrund der wenigen Steueroptionen und der Einfachheit des Messaging-Protokolls sehr simpel und leichtgewichtig [NAIK 2017].

Ein interessanter Aspekt des MQTT-Brokers ist, dass er die gesamte Datenlage seiner Kommunikationspartner aufbewahrt und so die Option bietet, zusätzlich als Zustandsdatenbank betrieben werden zu können. Weniger leistungsfähige Geräte können dadurch mit einem MQTT-Broker verbunden werden und Befehle und Daten entgegennehmen, zugleich entsteht aber ein komplexeres Lagebild auf dem Broker. So können Daten an einen leistungsfähigeren Kommunikationspartner weitergeleitet und dort ausgewertet werden.<sup>14</sup>

Auf diese Weise können durch das Message Queue Telemetry Transport-Protokoll Automatisierungslösungen geschaffen werden, wodurch im Segment IoT und Smart Home eine einfache Verwendung ermöglicht wird. Dementsprechend fand bis heute eine schnelle Verbreitung statt.

Zusammengefasst ist der MQTT-Broker die Kommunikationsschnittstelle der Smart Home-Geräte und der Smart Home-Plattform. Alle Kommunikationspartner können so Informationen (Nachrichten / Messages) auf bestimmte Topics (Endpunkten) senden und diese abonnieren (publish / subscribe).

## **Publish/Subscribe Kommunikationsmodell**

Das Prinzip des Publish/Subscribe Kommunikationsmodells besteht darin, dass Komponenten, die an vorgegebenen Topics interessiert sind, bestimmte Informationen konsumieren und ihr Interesse anmelden [HUNKELER, TRUONG und STANFORD-CLARK 2008]. Dieser Vorgang wird als Abonnement (subscription) bezeichnet. Geräte, die an dem Vorgang oder an bestimmten Informationen interessiert sind, werden als Abonnenten (subscriber) definiert. Im Gegenzug können Geräte und Komponenten bestimmte Informationen produzieren und diese veröffentlichen (publish) und über den Markler (broker) an bestimmte Abonnenten weitergeben. Diese Vermittlung der Informationen zwischen Herausgeber (publisher) und Abonnent (subscriber) erfolgt über den Markler, welcher sämtliche Abonnements koordiniert. Alle Abonnenten müssen sich explizit bei dem Broker anmelden, um die Informationen zu erhalten [HUNKELER, TRUONG und STANFORD-CLARK 2008].

---

<sup>14</sup><https://mqtt.org> Abgerufen am 13.04.2022



Abbildung 2.7: Themenbasiertes Pub/Sub Kommunikationsmodell [HUNKELER, TRUONG und STANFORD-CLARK 2008]

Dieses Prinzip ist die Grundlage des MQTT-Protokolls. Im Folgenden wird ein Beispiel aufgezeigt, welches die Kommunikation über das Publish/Subscribe Modell darstellt.

### Beispiel

Damit am Beispiel der Steuerzentrale, die im Rahmen dieser Arbeit konzipiert und prototypisch implementiert wird, ein Prozess gestartet werden kann, müssen bestimmte Ereignisse durch MQTT-Nachrichten eintreten. Das hier verwendete Beispiel ist das Öffnen einer Büroeingangstür. Hierbei wird von einem Relais an der Tür, welches das Türschloss steuert, eine Nachricht an den *Broker* herausgegeben:

```

1      publish -topic: Buero/Tuer/Zustand -message: "offen"

```

Code-Beispiel 2.1: Erzeugung und Veröffentlichung einer Nachricht

Diese Nachricht wird von der Steuerzentrale konsumiert und weiterverarbeitet. Damit der Informationskanal (topic) über die Steuerzentrale verfügbar ist, muss dieser Kanal konsumiert werden:

```

1      subscribe -topic: Buero/Tuer/Zustand

```

Code-Beispiel 2.2: Empfang und Konsum einer Nachricht

Mit der empfangenen Nachricht kann dann über die Steuerzentrale ein Prozess, welcher vorab als Regel implementiert wurde, ausgelöst werden, beispielsweise eine Durchsage im Büro.

### 2.3.3 AMQP

Das Advanced Message Queue Protocol (AMQP)-Protokoll ist ebenso ein leichtgewichtiges M2M-Protokoll, welches im Jahre 2003 von John O'Hara JPMorgan Chase in London, Großbritannien, entwickelt wurde [NAIK 2017]. Der Fokus dieses Protokolls liegt auf der Unternehmens-Messaging-Ebene. Es wird hohen Wert auf die Zuverlässigkeit, Sicherheit, Bereitstellung und Interoperabilität der Kommunikation gelegt. AMQP unterstützt neben der Publish/Subscribe- auch die Request/Response Architektur. Es bietet eine Bandbreite an Funktionen im Zusammenhang mit Messaging, wie z. B. Queuing, themenbasiertes Publish-and-Subscribe-Messaging, flexibles Routing und Transaktionen [NAIK 2017]. Das Kommunikationsmodell nach dem AMQP Standard erfordert, dass der Herausgeber (publisher) oder der Empfänger (subscriber) einen *Austausch (exchange)* mit einem bestimmten Namen generiert und diesen dann sendet [NAIK 2017]. Die beiden Komponenten, Empfänger und Herausgeber, nutzen den Namen zum Austausch und zum Verbindungsaufbau. Der Empfänger erstellt darauf eine Warteschlange (queue) und hängt diese an den Austausch an. Nachrichten, die über diese Verbindung ausgetauscht werden, müssen über einen gesonderten Prozess (binding) mit der Warteschlange abgeglichen werden [NAIK 2017].

Das binäre Protokoll AMQP erfordert einen Header von acht Byte mit Nachrichtennutzlasten. Die Größe der Nachricht ist abhängig von dem Broker, bzw. dem Server. Die verbindungsorientierte Kommunikation von AMQP baut auf dem Standard-Transportprotokoll TCP auf. Durch das TLS/SSL-Protokoll wird die Kommunikationssicherheit gewährleistet. Ein Wesensmerkmal des AMQP Kommunikationsmodells ist die Zuverlässigkeit [NAIK 2017].

Der direkte Vergleich der beiden Protokolle, MQTT und AMQP, ist dem Anhang (A) beigelegt.

Neben den beiden aufgeführten Kommunikationsmodellen gibt es noch weitere, darunter das klassische Hypertext Transfer Protocol (HTTP) und das Constrained Application Protocol (CoAP), die allerdings im Rahmen dieser Arbeit nicht weiter ausgeführt werden.



## 2.4 Home Assistant

Eine der populärsten Smart Home Plattformen ist das sogenannte Home Assistant System. Die Open-Source-Software ist ein zentrales Steuerungssystem von Heimautomationen, ebenso ist eine Verwaltung von intelligenten Geräten mit dem Fokus der lokalen Steuerung und Sicherung der Privatsphäre möglich. Der Zugriff kann über die Smartphone-App, jeweils verfügbar für iOS und Android, oder auch über die webbasierte Benutzeroberfläche (Web-App) erfolgen. In dem lokalen System können auch Geräte per Sprachbefehl gesteuert werden. Kompatible Plattformen sind unter anderem Google Assistant, Amazon Alexa und Apple HomeKit. Home Assistant bietet eine weitaus vielfältigere Verknüpfung von Geräten, Services und Plattformen. Die zentrale Steuerung unterstützt durch modulare Integrationskomponenten die einzelnen Geräte, Anwendungen und Services. Für die drahtlose Kommunikation werden native Integrationskomponenten verwendet, darunter Bluetooth, ZigBee und Z-Wave. Diese werden verwendet, um lokale Personal Area Network (PAN) mit Geräten mit geringem Stromverbrauch aufzubauen. Die Steuerung kann auch mit proprietären Ökosystemen stattfinden, sofern diese eine offene API oder Anbindungen über MQTT anbieten.<sup>15</sup> Die Plattform ist in Python geschrieben, wird aktiv instand gehalten und durch eine große Community unterstützt. Die Software ist allgemein unter der Apache 2.0, veröffentlicht. Der folgende Abschnitt befasst sich in Kürze mit der Historie des Systems.

### Historie

Anfang des vierten Quartals im Jahr 2013 startete das Python-Projekt von Paulus Schoutsen und wurde im November 2013 erstmals auf GitHub veröffentlicht.<sup>16</sup>

Vier Jahre nach den ersten Entwicklungen der Smart Home Plattform wurde im Juli 2017 ein verwaltetes Betriebssystem mit dem Namen *Hass.io* implementiert.<sup>17</sup> Dadurch gelang der Durchbruch der vereinfachten Verwendung der Home Assistant Plattform auf kleineren Computern, sogenannten Einplatinencomputern, wie beispielsweise ein Gerät der Raspberry Pi Serie. In Zusammenhang mit dem Betriebssystem kam ein *Supervisor*-Verwaltungssystem hinzu, das den Benutzern die Verwaltung, Sicherung und Aktualisierung der lokalen Installation erleichtert. Ein weiteres Feature des Supervisors ist die Möglichkeit, der Plattform über Add Ons weitere Funktionalitäten zur Verfügung zu stellen.<sup>18</sup>

Die Software wird stetig weiterentwickelt und verbessert. Mittlerweile gehört sie zu den am meisten genutzten Open-Source-Plattformen im Bereich Smart Home.

---

<sup>15</sup>Grundlegende Ableitung der Definition von Home Assistant siehe [https://en.wikipedia.org/wiki/Home\\_Assistant](https://en.wikipedia.org/wiki/Home_Assistant) Abgerufen am 16.04.2022

<sup>16</sup>Anfänge von Home Assistant. <https://www.linux.com/topic/embedded-iot/home-assistant-python-approach-home-automation/> Abgerufen am 18.04.2022

<sup>17</sup>Verkündungen von Home Assistant. <https://www.home-assistant.io/blog/categories/announcements/> Abgerufen am 18.04.2022

<sup>18</sup>Einstieg in das Hass.io Betriebssystem. <https://www.home-assistant.io/blog/2017/07/25/introducing-hassio/> Abgerufen am 18.04.2022

### 2.4.1 Konzept und Architektur

Home Assistant bietet eine Plattform für die zentrale Haussteuerung und die damit einhergehende Steuerung von Heimautomationen. Die Software ist nicht nur eine einfache Steuerungs- und Konfigurations-Software, sondern ein eingebettetes Betriebssystem, das verbraucher- und nutzerorientiert das Verwenden und Konfigurieren von Haussteuerungen erleichtert.<sup>19</sup>

Damit der offene Ansatz von Home Assistant gegenüber Anbietern von intelligenten Geräten nicht eingeschränkt ist, bietet die Software Möglichkeiten, um viele Geräte zu integrieren und über die Plattform zu nutzen. Somit begegnet Home Assistant der Heterogenität des offenen Marktes, in sofern, dass diese Geräte auf gemeinsame Konzepte gebracht werden. Diese sind in vier Konzepte<sup>20</sup> aufgeteilt, mit der die Vereinheitlichung vorangetrieben werden kann:

- Integration (Integration): Integrationen repräsentieren die Geräte und Dienste innerhalb der Home Assistant Anwendung. Ebenso können mittels den Integrationen auch Daten von Datenpunkten abgerufen werden.
- Gerät (Device): Nach der Konfiguration der Integration werden die Geräte in Home Assistant angelegt. Diese werden dann als erkannte Geräte der Integration dargestellt, z.B. als Temperatur-, Licht- oder Feuchtigkeitssensor.
- Entität (Entity): Die Datenpunkte sind die Geräte, die sogenannten Entitäten, die durch die Integrationen standardisiert werden. Dies sind Objekte, die Funktionalitäten oder Daten des Geräts darstellen, z.B. die Temperatur, Helligkeit oder die Feuchtigkeit.
- Automatisierung (Automation): Automatisierungen sind Prozesse, die bei einem bestimmten ausgelösten Event ausgeführt werden sollen. Dieses Auslöser (trigger) können Zeitpunkte, Ereignisse oder manuell gesteuerte Aktionen des Nutzers sein, z.B. das Ausschalten des Bürolichts, wenn durch einen Bewegungssensor fünf Minuten keine Bewegung festgestellt wurde oder wenn der Helligkeitswert, der über den Sensor festgestellt wurde einen bestimmten Wert erreicht hat, soll das Licht ebenso ausgeschaltet werden.

Die Architektur der Home Assistant Anwendung ist grundlegend als eingebettetes System eines Betriebssystems aufgestellt, welches in drei Schichten aufgeteilt ist. In unterster Ebene befindet sich das Betriebssystem, welches als minimales Linux System aufgestellt ist, um die darauf liegenden Schichten, den Aufseher (supervisor) und den Kern (core), zu betreiben. Mit dem Supervisor wird das Betriebssystem verwaltet und konfiguriert. Der eigentliche Kern interagiert mit dem Supervisor, den Geräten und den Services.

Der Supervisor ist die Schicht über dem Betriebssystem. Die Kommunikation der beiden Komponenten findet über einen D-Bus statt. Diese Zwischenschicht ermöglicht dem Benutzer die

<sup>19</sup>Konzept und Architektur von Home Assistant. [https://developers.home-assistant.io/docs/architecture\\_index](https://developers.home-assistant.io/docs/architecture_index) Abgerufen am 19.04.2022

<sup>20</sup>Erläuterung der Konzepte von Home Assistant. <https://apiumhub.com/tech-blog-barcelona/domotics-with-home-assistant-concepts/> Abgerufen am 21.04.2022

Verwaltung der Home Assistant Installation.

Die Aufgaben des Supervisors sind wie folgt definiert<sup>21</sup>:

- Dieser führt den Home Assistant Kern (Core) aus.
- Dieser führt die Updates des Home Assistant Core aus.
- Dieser führt einen *Rollback* bei Fehlgeschlagenem Update durch.
- Dieser führt Sicherungen und Wiederherstellungen durch.
- Dieser verwaltet die Add Ons der Core Instanz



Abbildung 2.8: Architektur des Home Assistant Supervisors [SCHOUTSEN 2020]

Die auf dem Supervisor aufbauende Architektur ist die Architektur der Anwendung, der sogenannte Core. Dieser besteht aus vier Komponenten, welche den Hauptteil abbilden: Event Bus, State Machine, Service Registry und Timer<sup>22</sup>.

Mit dem Event Bus wird das Abhören und Auslösen von Events und Ereignisse erleichtert. Die Komponente stellt eine zentrale Eigenschaft der Home Assistant Anwendung dar. Die Zustandsmaschine (State Machine) ist eine weitere Komponente, mit der die Zustände von Dingen, darunter intelligente Geräte, Sensoren uvm., überwacht und Zustandsänderungsereignisse an den Event Bus ausgelöst werden. Dies erfolgt nach der Änderung des Zustandes eines Objektes. Mit der Dienstregistrierung (Service Registry) wird der Event Bus auf eingehende Aufrufe von Diensten abgehört. Über die Service Registry kann der Benutzer Dienste hinzufügen und verwalten. Mittels den Entwicklertools, die über das User Interface (UI) aufgerufen werden können, kann der Benutzer die Dienste und Automatisierungen aufrufen und konfigurieren. Das letzte Element, der Timer, ist ebenso eine Komponente der Architektur, die zeitveränderte Ereignisse gemäß einer gegebenen Frequenz an den Event Bus sendet. Somit können zeitbasierte Automatisierungen vereinfacht und ausgelöst werden. Als Datenbank wird eine nicht Cloud-basierte SQLite<sup>23</sup> Datenbank verwendet. Diese ist nur auf dem

<sup>21</sup>Architektur des Home Assistant Supervisors. <https://developers.home-assistant.io/docs/supervisor> Abgerufen am 22.04.2022

<sup>22</sup>Entwickler Dokumentation der Home Assistant Plattform. <https://developers.home-assistant.io/docs/> Abgerufen am 24.04.2022

<sup>23</sup>Structured Query Language, eine Datenbanksprache zur Definition, Abfrage und Bearbeitung von Datenstrukturen in relationalen Datenbanken

Gerät enthalten und wird nicht über das Internet übertragen. Im lokalen Netzwerk hat der Nutzer die Möglichkeit, um auf die Datenbank zuzugreifen und eine Historie von Aktionen einzusehen. Eine Verlaufskomponente, die ebenso im Core enthalten ist, speichert die Ereignisse innerhalb der Plattform. Somit können Nutzer auf alle gespeicherten Informationen zu Hause zugreifen und diese einsehen [LI u. a. 2018].



Abbildung 2.9: Architektur des Home Assistant Cores [LI u. a. 2018]

### 2.4.2 Ziele und Schwerpunkte

Jede Softwarelösung verfolgt bestimmte Ziele und Schwerpunkte, damit die Zielerreichung (Definition of Done (DoD)) messbar ist. Die mit der Home Assistant Plattform verfolgten Ziele sind Privatsphäre und Sicherheit, die Wahlmöglichkeiten von Geräten und die Haltbarkeit der Plattform<sup>24</sup>.

Das Thema der Privatsphäre wird bei Home Assistant dadurch forciert, dass die Geräte nur optional über das Internet erreichbar sind, bzw. nur über das lokale Netzwerk. So kann die Kategorisierung des Verhaltens durch Algorithmen vermieden werden.

Die großflächige Auswahlmöglichkeit von Geräten und Verknüpfungen ist ein weiteres großes Ziel von Home Assistant, da es möglich ist herstellerunabhängig Geräte zu verwenden und zu integrieren. Dadurch können über eine Zentrale mehrere Geräte von diversen Herstellern kombiniert werden. Haltbarkeit wird an der Stelle adressiert, an der die Plattform uneingeschränkt verfügbar ist und

<sup>24</sup>Schwerpunkte der Home Assistant Plattform. <https://www.home-assistant.io/blog/2021/12/23/the-open-home/> Abgerufen am 24.04.2022

Geräte zur Anbindung verwendbar sind. Damit ist auch zu erwarten, dass Geräte, die eingebunden werden können, so konzipiert und gebaut sind, damit sie eine lange Lebensdauer besitzen.

Schwerpunkte, bzw. Lösungen zur Problembehebung sind die Kontrollierbarkeit über eine einzige zentrale Stelle und die Konfiguration von Automationen, die Prozesse selbstständig auslöst. Durch die Möglichkeit der Automatisierung und Steuerung von Komponenten im Eigenheim, bzw. im Büro kann die Lebensqualität erhöht als auch die Energy- und Stromkosten gesenkt werden. Die Kompatibilität der Plattform ermöglicht die Verwendung von mehreren Geräten und Herstellern über eine zentrale Stelle, der Home Assistant Software. Dies ermöglicht dem Nutzer die uneingeschränkte Verwendung von Geräten und die Unabhängigkeit zu Herstellern, die ggf. den Preis der Geräte über dem Marktdurchschnitt verkaufen. Ein weiterer Schwerpunkt ist die Sichtbarkeit der Daten, die potentiell gesendet werden, um den Datenschutz und die Privatsphäre auf dem höchsten Standard zu halten. Ein Auswirkung dafür ist die zugrundeliegende lokale Datenhaltung über die Datenbank, die direkt auf dem Gerät der Plattform läuft.

### 2.4.3 Stärken und Schwächen

Die Stärken und Schwächen der Plattform belaufen sich auf die folgenden Aspekte:

Stärken	Schwächen
Open-Source Software	Schwerer Einstieg für neue Nutzer im Hinblick auf die Konfiguration der Automationen
Privatsphäre & Datenschutz	Automationen im YAML-Datenformat und benutzerdefinierte Komponenten können für neue Benutzer schwierig sein
Große Community	Langwierige Installationen von Integrationen
Automationen sind sehr mächtig	Die Erweiterung, Einrichtung und Konfiguration erfordert die Bearbeitung von YAML-Dateien und einen Neustart bei jeder Änderung
Stetige Aktualisierung, Weiterentwicklung und Fehlerbehebung	Automationen können kompliziert aufgebaut sein
Unterstützung vieler Geräte (Herstellerunabhängigkeit) und großflächige Kompatibilität	Nutzung von ZigBee und Z-Wave erfordert jeweils eigene Konfigurationen, welche die All-in-One Lösung erschwert.
Individuelle Modifikation der UI	
Kann auf beliebiger Hardware ausgeführt werden	

Tabelle 2.5: Stärken und Schwächen der Home Assistant Plattform

Die Konfiguration von Automationen, Funktionen und Integrationen über YAML-Dateien kann auf der einen Seite als Stärke gewertet werden, da die Syntax flexibel einsetzbar ist und Freiheiten

bei der Implementierung bietet und auf der anderen Seite als Schwäche, da die Konfigurationen schnell unübersichtlich werden und anfangs schwer zu verstehen sind. Der Aufwand steigt je nach zu implementierende Funktion und deren Abhängigkeiten zu Geräten und gegebenenfalls zu anderen Prozessen und Automationen.

#### 2.4.4 Optionen der Regel- und Automatisierungserstellung

Home Assistant bietet zum Umsetzen von Regeln und Automationen<sup>25</sup> eine Option an. Dabei werden die Informationen über YAML-Dateien dargestellt. Hierbei gilt ein klares Konstrukt, welches eingehalten werden muss, um lauffähige Regeln und Automationen ausführen zu können. Für den Nutzer wird lediglich die Option geboten, die Automationen über eine Oberfläche zu erstellen. In dieser Form gibt ein Template die einzustellenden Möglichkeiten vor. Hierbei kann der Name und der Modus der Automation festgelegt werden. Anschließend wird ein Auslöser, beispielsweise ein Gerät oder eine Zeitangabe, definiert und entsprechend eine Bedingung zur Ausführung der Automation hinzugefügt. Nach den Schritten wird die eigentliche Aktion hinzugefügt, die ausgeführt werden soll, wenn die Bedingung erfüllt ist. Die generierte Automation wird als YAML-Datei gespeichert und kann ebenso über einen Editor in Rohformat editiert werden. Eine Automation über eine YAML-Datei in der Home Assistant Software gibt ein Konstrukt vor, welches mit den oben genannten Eigenschaften gefüllt werden muss. Die dafür vorgesehene Schablone sieht wie folgt aus:

---

```
1      automation:
2        - alias: "<NAME>"
3          trigger:
4            <TRIGGER>
5          condition:
6            - <TRIGGER_CONDITION> OR
7              <TRIGGER_CONDITION2> ...
8          action:
9            - <ACTION>
```

---

Code-Beispiel 2.3: Konstrukt zur Regeldefinition über Home Assistant

---

<sup>25</sup>Erstellen von Regeln und Automationen über Home Assistant. <https://www.home-assistant.io/docs/automation/> Abgerufen am 28.05.2022

## 2.5 openHAB

Neben der so eben erläuterten Home Assistant Plattform zählt ebenso die openHAB Plattform als bekannt und in der Anwendung populär. Der open Home Automation Bus (openHAB) ist eine Plattform, bei der es sich um eine Softwarelösung handelt, die auf Basis der Programmiersprache Java aufgebaut ist. Die Software steht unter der Eclipse Public License und fällt daher unter die Rubrik der Open-Source Software. Durch die Verwendung von Java ist die Anwendung betriebssystemunabhängig und kann auf beliebigen Betriebssystemen laufen. Ähnlich zu der vorgestellten Home Assistant Software, bietet openHAB ebenso User Interfaces die durch den Webbrowser, Android- und iOS-Geräte unterstützt werden.

In Kombination mit Java wird bei openHAB das Open Service Gateway initiative (OSGi)-Framework für die Modularität der Software verwendet. Mit Apache Karaf wird ein Container bereitgestellt, der mit Eclipse Equinox als OSGi Laufzeitumgebung agiert. Als HTTP-Server ist Jetty in Gebrauch. Die einzelnen Frameworks werden nicht im Detail erläutert, lediglich die für das Verständnis des Konzeptes notwendigen.

Mit openHAB wird eine hochmodulare Software zur Verfügung gestellt, die durch sogenannte *Add-ons* erweitert werden kann. Durch diese wird der Plattform eine breite Palette an Funktionen geboten. Physische Geräte können in großer Anzahl mit der Plattform interagieren und verknüpft werden.<sup>26</sup>

### Historie

Das Smart Home Projekt begann im November 2009 und wurde von Kai Kreuzer entwickelt<sup>27</sup>. Im Februar 2010 wurde das Projekt unter dem Name openHAB bekannt. Nach zweieinhalb jähriger Entwicklung, im Dezember 2012, gab Herrn Kreutzer ein Statement zu der Verwendung von OSGi ab, die seine Überzeugung der Verwendung des Frameworks nach wie vor kundtut.

„Looking back at this evolution of the project, I am perfectly sure that if I had designed openHAB as a normal Java application instead of an OSGi application, it would not have prospered as it did. It is really the choice of the software architecture that made it happen - and as a nice side effect, the community is not a pure user community as it is the case for many other Open Source projects, but it is full of engaged people who actively contribute to the project.[KREUZER 2012]“

Mit der stetig wachsenden Anwendung und deren Community nimmt die Entwicklung des Projekts immer mehr Fahrt auf. 2014 betitelt Kreutzer das Jahr als Jahr des Smart Home, da die Beteiligung, das Interesse und die Entwicklung enorm zunahm und die Community immer größer wurde [KREUZER 2014]. Seitdem kamen weitere Unterstützungen zu anderen Geräten und Kommunikationsprotokollen dazu. Im Laufe der Entwicklung wurde eine mobile Anwendung entwickelt, mit

---

<sup>26</sup>Einleitung zu openHAB. <https://www.openhab.org/docs/> Abgerufen am 25.04.2022

<sup>27</sup>Chronologie und Blogbeiträge von Kai Kreutzer. <http://kaikreutzer.blogspot.com> Abgerufen am 28.04.2022.



der die Remotesteuerung implementiert ist. Die Software wird regelmäßig gepatched und von der Community unterstützt.

### 2.5.1 Konzept und Architektur

Die Steuerungsplattform openHAB bietet vergleichbar zu Home Assistant die Möglichkeit der multifunktionalen Verknüpfung von Geräten und Protokollen. An dieser Stelle werden ebenso mehrere Konzepte verwendet, die die Vereinheitlichung der Plattform verstärkt. Die Konzepte der openHab Software sind in drei größere Rubriken aufgeteilt, die sich wie folgt zusammensetzen:

Die erste Rubrik sind die Dinge (Things), diese sind die Entitäten, die als physische Komponente zu einem System hinzugefügt und viele Funktionalitäten als eines bereitstellen kann. Hierbei ist zu berücksichtigen, dass die sogenannten Dinge nicht immer Geräte sein müssen, diese können auch andere überschaubare Informationsquellen, andere Webdienste und Funktionalitäten darstellen. Aus Sicht des Benutzers sind sie für den Einrichtungs- und Konfigurationsprozess relevant, für den Betrieb jedoch potentiell zu vernachlässigen. Dinge können Konfigurationseigenschaften haben, die optional oder obligatorisch sein können. Solche Eigenschaften können grundlegende Informationen wie eine IP-Adresse, ein Zugriffstoken für einen Webdienst oder eine gerätespezifische Konfiguration sein, die sein Verhalten ändert [PRANZ und SCHILLER 2018]. Mit dem Konzept der Dinge kommen zwei Unterkategorien einher:

- Kanäle (Channels): Jedes Gerät, bzw. Ding stellt Kanäle bereit, mit denen die jeweiligen Funktionen abgebildet werden. An der Stelle an der das physische Gerät angebunden ist, ist der Kanal eine konkrete Funktion dieser Entität. Beispielsweise kann eine Glühbirne einen Kanal für die Farbtemperatur und einen für den Farbwert besitzen. Diese stellen beide die Funktionalität der einen physischen Glühbirne für das System bereit. Grundlegend sind Kanäle mit Elementen verknüpft, mit denen die virtuelle und physische Ebene verbunden wird. Ab dem Zeitpunkt, sobald eine Verbindung hergestellt wird, reagiert ein Ding auf Ereignisse, die für ein Element transferiert werden. Bedingung dafür ist die Verknüpfung zu einem Kanal. Auf der anderen Seite werden aktiv Ereignisse für Objekte gesendet, die mit Kanälen des Dings verknüpft sind [PRANZ und SCHILLER 2018].
- Brücken (Bridges): Die Brücke ist eine besondere Art von Ding. Diese müssen dem System hinzugefügt werden, damit der Zugriff auf andere Dinge ermöglicht wird, bzw. erhalten bleibt. Ein IP-Gateway für Hausautomationssysteme, welches nicht IP-basiert funktioniert, ist ein typisches Beispiel für so eine Brücke.

An zweiter Stelle stehen die Artikel (Items). Diese Elemente stellen Funktionen dar, die direkt von der Anwendung verwendet werden. Darunter zählen hauptsächlich die Automatisierungslogik oder auch Benutzeroberflächen. Durch Ereignisse werden die Elemente verwendet, da diese einen Zustand besitzen. Elemente können auch in eine Gruppe zusammengefasst werden. Ein Gruppenelement kann auch Mitglied einer weiteren Gruppe sein. Diese zyklischen Mitgliedschaften sind war nicht verboten, davon wird jedoch abgeraten. Über Benutzeroberflächen können einzelne Gruppenelemente als alleinstehender Eintrag angezeigt werden und eine Navigation zu den jeweiligen Mitgliedern



bereitstellen.

Die dritte große Rubrik sind die Bindungen und Links (Bindings and Links). Diese können als Softwareadapter betrachtet werden, die Dinge für ihr Hausautomationssystem zur Verfügung stellen [PRANZ und SCHILLER 2018]. Aufgabe der Komponente ist die Verknüpfung von Elementen mit physischen Geräten. Um dies zu ermöglichen werden die spezifischen Kommunikationsanforderungen eines Geräts abstrahiert. Dadurch ist eine allgemeinere Behandlung der Komponente durch das Framework möglich. Links stellen das Bindeglied zwischen Dingen und Gegenständen dar. Es ist die Verknüpfung von genau einem Element und einem Kanal. Elemente und Kanäle haben keine eins zu eins Beziehung zueinander. Eine Verknüpfung mit mehreren Komponenten ist auch hier möglich. Neben den übergreifenden Konzepten gibt es weitere Konzepte<sup>28</sup>. Diese werden in Folgendem aufgelistet:

- Seitenverzeichnis (Sitemaps): Dies beinhaltet die Konfiguration eines Seitenverzeichnisses, mit dem Schalter, Regler, Texte und vieles mehr hinzugefügt werden kann. Diese Ansichten repräsentieren den Status eines Elements und können darüber gesteuert und überprüft werden.
- Regeln (Rules): Eine Regel ist die Verknüpfung eines Ereignisses und einer Aktion. Somit wird nach einem bestimmten Ereignis eine Aktion ausgeführt. Ein Beispiel: Wenn ein Fenster geöffnet wird, soll die Heizung abgeschaltet werden. So lange, bis das Fenster wieder geschlossen wird.
- Persistenz (Persistence): Mit der Persistenzschicht kann festgelegt werden, welche historischen Zustandsereignisse und Informationen gespeichert werden sollen. Dadurch kann nach Systemausfall der Status des Elementes wiederhergestellt werden.
- Modelle (Models): Modelle stellen baumbasierte Gruppierungen von Elementen dar. In der Baumstruktur können Elemente Orte, Punkte und Ausrüstungen sein. Ein Beispiel: Das Wohnzimmer als Standort besteht aus zwei Geräten Heizung und Licht. Die Heizung als Ausstattung hat zwei Punkte: *Ist-Temperatur* und *Soll-Temperatur*.
- Seiten (Pages): Alle Seiten und Seitenverzeichnisse kontrollieren die Zustände von Elementen und greifen auf diese zu. Mit openHAB 3 würden weitere Möglichkeiten hinzugefügt, um benutzerdefinierte UI, Diagramme, Grundrisse und oder Layouts zu definieren.
- Skripte (Scripts): Skripte sind ähnlich zu den Regeln. Diese spezifizieren eine Aktion und kein Ereignis. Die Skripte können manuell, aber auch über andere Skripte und Regeln ausgeführt werden.

Die anfängliche Architektur der openHAB Anwendung teilt sich in drei Sektionen auf: Die Kernkomponenten (blau), das darunter liegende OSGi-Framework (grün) und die Erweiterungen (Add-Ons) (gelb).

---

<sup>28</sup>Historie und Konzepte von openHAB. <https://medium.com/smartsmarthome/openhab-41c30d50fc42> Abgerufen am 28.04.2022



Abbildung 2.10: Architektur der openHAB Plattform [KREUZER 2013]

Mit der wesentlichen Aktualisierung von openHAB zur aktuellen Version openHAB 2.0 gab es auch einige Änderungen an der grundlegenden Architektur. Es basiert hauptsächlich auf dem Eclipse SmartHome Framework und nutzt Apache Karaf zusammen mit Eclipse Equinox, um eine OSGi Laufzeitumgebung zu erstellen. Jetty wird als HTTP Server verwendet.

*Eclipse SmartHome*<sup>29</sup> ist ein IoT Projekt, welches sich aus der ersten Version der openHAB Plattform entwickelte. Diese wurde von der Eclipse Foundation übernommen und weiterentwickelt. Es stellt das Kern-Framework für Smart Home Systeme dar und ist essentieller Bestandteil der openHAB Plattform. Das Framework ist mittlerweile archiviert und erhält keine weiteren Aktualisierungen.

*Apache Karaf*<sup>30</sup> ist eine OSGi basierte Laufzeitumgebung über die verschiedene Komponenten mittels Container bereitgestellt werden können. Dieses Framework zählt zu denen der Apache Software Foundation und unterliegt der Apache V2 Lizenz.

*Eclipse Equinox*<sup>31</sup> ist eine modulare Java basierte Laufzeitumgebung und eine Implementierung

<sup>29</sup>Eclipse SmartHome. <https://projects.eclipse.org/projects/iot.smarthome> Abgerufen am 29.04.2022

<sup>30</sup>Apache Karaf. <https://karaf.apache.org/> Abgerufen am 29.04.2022

<sup>31</sup>Eclipse Equinox. <https://www.eclipse.org/equinox/> Abgerufen am 29.04.2022

des OSGi Framework. Der Begriff der *Bundles* ist das Herzstück der OSGi-Spezifikation. Diese werden verwendet, damit die Modularität für Java erfasst wird. Ein Bundle ist ein Standard-Java-JAR-Datei, dessen Manifest zusätzliches Markup enthält, welches das Bundle identifiziert und seine Abhängigkeiten angibt. Als solches ist jedes Bundle vollständig selbstbeschreibend [PRANZ und SCHILLER 2018]. *OSGi* fördert den modularen Aufbau von Software, die mit der Programmiersprache



Abbildung 2.11: Architektur der openHAB 2.0 Plattform [KREUZER 2016]

Java geschrieben ist. Der Open-Source Standard bietet einen Ansatz, der eine modulare Plattform beschreibt und eine Java Virtual Machine (JVM) voraussetzt. Die Java Virtual Machine<sup>32</sup> ist der Teil der Java-Laufzeitumgebung, die für die Ausführung des Java-Bytecodes verantwortlich ist. Die sogenannten *Bundles* sind Module und bilden die kleinste Einheit der Modularisierung. Aus technischer Sicht ist dieses Konstrukt eine Java Archive (JAR)-Datei mit Metainformationen. Diese sind in einer Manifestdatei gespeichert. Bundles sind eindeutig zu identifizieren. Dadurch ist es in OSGi möglich, *Bundles* mit demselben Namen, jedoch unterschiedlicher Versionen gleichzeitig zu nutzen und auszuführen [PRANZ und SCHILLER 2018].

<sup>32</sup>Detaillierte Beschreibung der JVM. <https://www.javatpoint.com/jvm-java-virtual-machine> Abgerufen am 01.05.2022

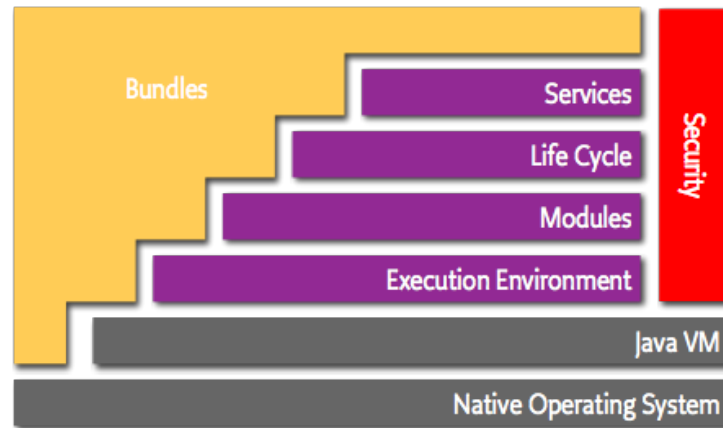


Abbildung 2.12: OSGi Schichtenarchitektur [KREUZER 2022]

Weitere Details der OSGi-Architektur<sup>33</sup> sind der openHAB Dokumentation zu entnehmen.

### 2.5.2 Ziele und Schwerpunkte

„openHAB empowering the smart home“<sup>34</sup>

Das große Ziel, dass mit der openHAB Anwendung verfolgt wird, ist allgemein die zentrale Steuerung von Geräten innerhalb eines Wohnraumes.

„... is an open source, technology agnostic home automation platform which runs as the center of your smart home!“<sup>35</sup>

### 2.5.3 Stärken und Schwächen

Die Stärken, die openHAB mit sich bringt, werden schon in den Dokumentationen aufgegriffen und bei der Nutzung der Anwendung schnell deutlich. Mit der Plattform können eine Vielzahl von Geräten und Systemen integriert werden, die miteinander kommunizieren. In der einzigen von openHab gegebenen Lösung können Heimautomatisierungssysteme, intelligente Geräte und andere Technologien, darunter beispielsweise Kommunikationsprotokolle, integriert werden [KREUZER 2020]. Durch die in dem letzten großen Update der Plattform hinzukommende einheitliche Benutzeroberfläche, die die Bedienung von Geräten und Systemen vereinfacht, ist die Integration von neuen Geräten einfacher. Mit dem Ansatz von Automatisierungsregeln können alle Geräte bei Bedarf miteinander kommunizieren. Das Aufstellen von solchen Regeln ist herstellerunabhängig

<sup>33</sup> Architektur und Erläuterung OSGi. <https://www.openhab.org/docs/developer/osgi/osgi.html> Abgerufen am 01.05.2022

<sup>34</sup> Log und Label von openHAB <https://www.openhab.org/> Abgerufen am 01.05.2022

<sup>35</sup> Einleitung und Übersicht der openHAB Dokumentation. <https://www.openhab.org/docs/> Abgerufen am 01.05.2022

und macht die Automatisierung flexibler. Der dritte und letzte Punkt der Stärken ist die allgemeine Vielfältigkeit und Flexibilität der Plattform. Hiermit ist im Allgemeinen die Umsetzung von Use Cases und Automatisierungen gemeint. Dem Nutzer sind keine Grenzen der Kompatibilität gesetzt.

Im Gegenzug sind die Schwächen hauptsächlich Auswirkungen der komplexen und weitgestrickten Architektur. Das System an sich und der Umgang damit stellt sich als komplex dar. In das Aufsetzen und Verstehen der Plattform, deren Konzepte und Möglichkeiten muss ein hoher Aufwand gesteckt werden. Die Dokumentation der Software geht auf diese Aspekte ein und gibt dem Leser mit, dass die Software nicht innerhalb kürzester Zeit verstanden werden kann [KREUZER 2020].

Eine weitere größere Schwäche ist die Weiterentwicklung des Systems. Diese findet langsam und nicht konstant statt. Auslöser dafür ist die Architektur und deren Abhängigkeiten zu weiteren Frameworks und Bibliotheken. Auch nimmt der Prozess bis zur Genehmigung von Weiterentwicklungen viel Zeit in Anspruch. Dies ist möglicherweise mit ein Grund, weshalb neue Versionen und Updates nur unregelmäßig zu beobachten sind.

#### 2.5.4 Optionen der Regel- und Automatisierungserstellung

Im Vergleich zu Home Assistant bietet openHAB zur Definition und Erstellung von Regeln mehrere Möglichkeiten an. Zur nutzerfreundlichen Bedienung unterbreitet die openHAB Lösung ebenso eine Oberfläche, über die Regeln definiert werden können. Die Konfiguration solcher Regeln kann jedoch mit unterschiedlichen Methoden umgesetzt werden. Zum einen mit einem visualisierten Baukasten Werkzeug Namens *Blockly*<sup>36</sup> und zum anderen über eine Domain Specific Language (DSL), im Deutschen eine domänenspezifische Sprache<sup>37</sup>. openHAB verwendet für die Syntax der Regeldefinition Xbase<sup>38</sup> und darauf aufbauend Xtend<sup>39</sup>. Diese DSL gibt ein Konstrukt vor, welches eine Regel definiert. Das vorgegebene Konstrukt sieht wie folgt aus:

---

```
1      rule "<RULE_NAME>"
2      when
3          <TRIGGER_CONDITION> [or <TRIGGER_CONDITION2> [or ...]]
4      then
5          <SCRIPT_BLOCK>
6      end
```

---

Code-Beispiel 2.4: Konstrukt zur Regeldefinition über Xbase

Jeder Regel muss ein eindeutiger Namen vergeben werden, der im besten Fall beim Aussprechen einen Sinn ergibt und den Inhalt der Regel wiedergibt. Danach muss eine Bedingung implementiert werden, die ebenso ein Event bei bestimmten eintreffenden Bedingungen auslöst. Die Logik der

<sup>36</sup>Ein visueller Code Editor von Google. <https://developers.google.com/blockly> Abgerufen am 29.05.2022

<sup>37</sup>Erklärung der domänenspezifischen Sprache. <https://martinfowler.com/dsl.html> Abgerufen am 29.05.2022

<sup>38</sup>Erklärung des Xtext, bzw. Xbase Frameworks. <https://www.eclipse.org/Xtext/index.html> Abgerufen am 29.05.2022

<sup>39</sup>Javadialekt, welcher durch das Framework in lesbaren Java 8 kompatiblen Quellcode kompiliert wird. <https://www.eclipse.org/xtend/index.html> Abgerufen am 29.05.2022

Regel wird dann über den Skript-Block implementiert. Für die Syntax der Regellogik wird *Xtend* verwendet.

## 2.6 Vergleich von Home Assistant und openHAB

Nach der Erläuterung der beiden Softwarelösungen im Bereich der Smart Home Plattformen, werden diese abschließend gegenübergestellt, um weitere Aspekte miteinander zu vergleichen. Grundlage dafür sind persönliche Erfahrungen als auch Eindrücke von Nutzern und Experten. Die Tabelle (B.1) zum Vergleich der Plattformen ist dem Anhang (B) zu entnehmen.

Zusammenfassend zu dieser tabellarischen Gegenüberstellung ist zu sagen, dass beide Plattformen ihre markanten Stärken vorweisen. Beide Ansätze bieten eine gute Grundlage, die Automatisierungen in dem eigenen Gebäude voranzutreiben. Die Entscheidung, welches Tool verwendet werden soll, liegt in der Verantwortung des Nutzers. Übergreifend sind die beiden Open-Source Lösungen eine gute Alternative zu kommerziellen Produkten und Lösungen, sofern der Nutzer herstellerunabhängig interagieren möchte. Dennoch bedeutet es für Nutzer und Entwickler einen hohen Aufwand zu leisten, um Automationen und Regeln zu definieren, da Werkzeuge eingesetzt werden, die nicht dem Standard der verwendeten Programmiersprache, beispielsweise von Java, entsprechen. Dadurch muss zu erst die Funktionsweise des Frameworks verstanden werden, um anschließend Automationen und Regeln definieren und anwenden zu können. Beide Lösungen versuchen diese Herausforderung zu lösen, indem über eine Weboberfläche die Regeln konfiguriert und erstellt werden können. Home Assistant vereinfacht dies durch die Nutzung des YAML-Dateiformates, im Gegenzug nutzt openHAB ein weiteres Framework, dass das Konstrukt der Regeldefinition vereinfacht, dennoch in seiner Interpretation und Anwendung eingeschränkt ist.

Beide Softwarelösungen bieten einen unterschiedlichen Ansatz zur Regeldefinition, die Flexibilität jedoch ist auf integrierte Komponenten beschränkt. Das bedeutet, dass nur vom System abbildbare Komponenten, Integrationen und Verknüpfungen verwendet werden können. Gibt es beispielsweise für eine Komponenten oder ein Gerät keine Integration oder kein Add-on, so muss dieses erst im Quellcode als Packet implementiert und angeboten werden, bevor der Anwender dieses verwenden kann.

Die aufgeführten Grundlagen bilden ein grundlegendes Verständnis des Kontextes dieser Arbeit. Die Erläuterung von IoT und Smart Home geben einen Einblick in die Domäne der Bereiche. Mit den Kommunikationsprotokolle werden Standards und derzeit oft verwendete Technologien beschrieben. Das Einführen in zwei Softwareplattformen gibt einen Überblick darüber, welche Lösungen bereits vorhanden sind und wie diese aufgebaut sind. Diese Information ist notwendig, um in weiterem Verlauf die Konzeption als auch die anschließende Analyse nachvollziehen zu können.

## Kapitel 3

# Stand der Technik

Zur Analyse des aktuellen Standes der Technik und Forschung in Bezug zur Konzeption von Software-Lösungen, mit denen die formalisierten Interaktionen der Softwareentwickler vereinfacht werden können, erfolgt in diesem Kapitel ein systematisches Literaturreview. Die Literaturprüfung wird gemäß den Richtlinien, die in der Publikation von [KITCHENHAM 2007] vorgeschlagen werden, durchgeführt.

### 3.1 Systematisches Literaturreview

Die Thematik des systematischen Literaturreviews wurde bereits in den einleitenden Kapiteln erwähnt. Dieser Abschnitt widmet sich ausschließlich der Anwendung der Richtlinien und dem daraus abgeleiteten Stand der Technik abhängig zu dem in der Arbeit behandelten Thema.

#### 3.1.1 Ziele des Systematischen Literaturreviews

Das Ziel dieses systematischen Literaturreviews ist es, die aktuellen Fortschritte von Smart Home Plattformen und Gateways in Richtung der entwicklerseitigen Benutzerfreundlichkeit zu recherchieren. Dabei liegt der Schwerpunkt auf der Usability und der einfachen Handhabung der formalisierten Interaktionen der Softwareentwickler. Es gilt zu analysieren, ob es in diesem Themenbereich bereits Publikationen und Forschungen gibt und welche Entscheidungen getroffen werden müssen, um die Weiterentwicklung eines Systems nicht je nach hinzukommender Funktionalität oder auch Bedingung komplexer werden zu lassen. Die Ergebnisse dieses systematischen Literaturreviews sollen daraufhin als Grundlage der Konzeption einer solchen Plattform dienen und mit einfließen.

#### 3.1.2 Suchstrategie- und anfragen

Dieser Abschnitt beschreibt die Suchstrategie und die Anfragen zu dem systematischen Literaturreview. Hierbei wird erläutert, anhand welcher Kriterien die Literatur ausgewählt wird. In den ersten Schritten werden anhand der Forschungsfragen in Kapitel (1.2) die Stichpunkte aufgegriffen und als Suchterm formuliert. Die daraus resultierenden Suchterme, die der tabellari-schen Darstellung (3.1) zu entnehmen sind, werden in diversen wissenschaftlichen Fachdatenbanken

recherchiert und analysiert. Die Ergebnisse werden nach ihrem Titel und der Zusammenfassung sortiert. Gibt es Publikationen mit vergleichbaren Inhalten, so sind diese in weiteren Schritten näher zu betrachten. Damit die Literaturrecherche weitere Ergebnisse erzielt, wird beim Studieren der Publikationen die Schneeballsuche angewendet. Dabei wird in den Quellen der jeweiligen Literatur auf weitere Verweise geschaut, die ebenso potentielle Inhalte bearbeiten. Die Kombination beider Literaturergebnisse bilden die Grundlage der zu analysierenden Quellen und geben so den Stand der Technik wieder.



Abbildung 3.1: Strategie der Literatursuche

Die generierten Suchterme, die daraus resultierenden Literaturergebnisse und der damit einhergehende Suchverlauf ist zur Nachvollziehbarkeit tabellarisch aufgeführt:



Suchanfrage	Datum	Filter	Plattform	Ergebnisse	Gesehene	Relevant
formalized interactions software development architecture	11.04.2022, 03.05.2022	Nein	IEEEExplorer, WebOfScience	13, 26	6, 4	0, 0
usability AND formalized interactions AND architecture	11.04.2022, 03.05.2022	Nein	IEEEExplorer, WebOfScience	3, 7	1, 3	0, 1
usability AND formalized interactions AND architecture AND smart home	01.05.2022	Nein	IEEEExplorer, WebOfScience	0, 0	0, 0	0, 0
usability AND formalized interactions AND software development AND architecture	02.05.2022	Nein	IEEEExplorer, WebOfScience	1, 1	1, 1	0, 1
usability AND architecture AND gateway AND smart home	06.05.2022	Nein	IEEEExplorer, WebOfScience	2, 4	1, 1	1, 1
usability AND formalized interaction AND (architecture OR smart home OR software developer OR gateway)	09.05.2022	Nein	IEEEExplorer, WebOfScience	3, 10	1, 2	0, 0
usability AND architecture AND smart home AND (formalized interaction OR software developer OR iot)	09.05.2022	Nein	IEEEExplorer, WebOfScience	13, 22	3, 3	0, 0

Tabelle 3.1: Suchprotokoll des Systematischen Literaturreviews

### 3.1.3 Datenextraktion und Synthese

Zu der zielgerichteten Datenextraktion werden in Anlehnung an die Richtlinien des systematischen Literaturreviews die folgenden Einschluss- und Ausschlusskriterien definiert, die dabei helfen, die relevanten Publikationen zu finden:

#### # Einschluss-Kriterien

- |   |   |
|---|---|
| 1 | Die Literatur ist in den wissenschaftlichen Fachdatenbanken veröffentlicht, darunter: WebOfScience, IEEEExplorer, SpringerLink, Elsevier, Addison-Wesley, dpunkt-Verlag, ACM und Google Scholar |
| 2 | Der Beitrag wurde nach 2010 veröffentlicht  |
| 3 | Die Veröffentlichung ist in deutscher oder englischer Sprache   |

#### # Ausschluss-Kriterien

- |   |  |
|---|--|
| 1 | Die Veröffentlichung beinhaltet, bzw. behandelt nicht die Schlagworte usability, architecture, formalized interaction oder iot |
| 2 | Die Publikation gehört zu der Literatur der Grauzone   |
| 3 | Die Veröffentlichung hat weniger als 5 Zitationen  |
| 4 | Die Literatur hat weniger als 5 Seiten Inhalt  |

Tabelle 3.2: Einschluss- und Ausschlusskriterien des systematischen Literaturreviews

Mit der Forschungsfrage, den Suchtermen und den Einschluss- und Ausschlusskriterien zu der Quellenrecherche werden während der Anwendung des systematischen Literaturreview-Templates und deren Richtlinien die erzielten Ergebnisse synthetisiert und zusammengefasst. Im Rahmen dieser Forschungsfrage und der Auslegung der Suchterme, die in Tabelle (3.1) zu entnehmen sind, ergab sich nicht die Menge an Literatur, die notwendig ist, um ein umfangreiches Literaturreview im Stil des Templates durchzuführen. Demnach ist kein vollständiges systematisches Literaturreview möglich. Die wenigen relevanten Ergebnisse des Literaturreviews geben einen Einblick in Bereiche, die als Gedankenanstoß und zum transferieren von Wissen, Gedanken und Ideen geeignet sind.

Unter der Betrachtung des Resultats des durchgeführten Literaturreviews gilt die Forschungsfrage als innovativ und eröffnet eine neue Sichtweise basierend auf der Literatur. Um dennoch eine fundierte wissenschaftliche Grundlage zu repräsentieren, wird in folgendem Abschnitt auf Referenzen und Literatur eingegangen, die Teile der Forschungsfrage abdecken und mehr als Gedankenanstoß anzusehen sind, beziehungsweise auch einen partiellen Einblick in den Stand der Technik bieten.

## 3.2 Zusammenfassung

Speziell auf die Forschungsfrage in Kapitel (1.2) gibt es in dem heutigen Stand der Technik keine explizite Literatur, welche den Gedanken aufgreift und behandelt. Die Literatur, die bei dem systematischen Literaturrecherche erarbeitet wurde, deckt trotz dessen manche Aspekten ab, die interessante Anhaltspunkte und Ideen thematisieren. Diese geben Impulse und helfen bei den weiteren Schritten in dieser Arbeit hinsichtlich der Konzeption.

### 3.2.1 Publikationen

Im Folgenden wird die relevante Literatur aufgegriffen und zusammengefasst, sodass ein Einblick in den Prozess der Informationserhebung, der Sammlung von Erfahrungswerten und Ideen gewährleistet wird.

#### **Design and Realization of a Framework for Human-System Interaction in Smart Homes**

In dem Artikel von [WU und FU 2012] wird zu Anfang die Beziehung zwischen Benutzern, Räumlichkeiten und Diensten analysiert. Basierend auf den daraus gewonnenen Erkenntnissen wird ein Framework und ein entsprechender Algorithmus vorgestellt, welcher die Interaktionsbeziehungen modelliert. Basierend auf den Ergebnissen wird ein Framework entwickelt, welches die Interaktionsanforderungen abdeckt. Hauptmerkmale waren dabei Komfort, Bequemlichkeit und Sicherheit. Zur abschließenden Überprüfung des Designkonzeptes und der Implementierung wurden Probanden zum Testen der Anwendung ausgewählt und anschließend ein Interview durchgeführt. Das Evaluierungsergebnis zeigt, dass das Framework eine gute Einführung in die Verbesserung der Mensch-System-Interaktionen darstellt.

#### **Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes**

Der Artikel von [KIM u. a. 2012] erarbeitet einen Vorschlag einer ganzheitlichen und erweiterbaren Softwarearchitektur, welches Dienste und heterogene protokoll- und herstellerspezifische Geräte nahtlos integrieren lässt und Sicherheit über das Internet gewährleistet. Grundlegend wird hierbei auf das OSGi Framework gesetzt. Dadurch wird die semantische Interoperabilität hervorgehoben. Dies ist die Fähigkeit, neue Anwendungen und Treiber zur Laufzeit in das bereitgestellte System zu integrieren [KIM u. a. 2012]. Zusätzlich zu dem System wird im Rahmen dieser Publikation ein Zugangskontrollmodell für spezielle Smart Home Szenarien integriert. Zur Beweisführung wird das Konzept anhand von semantischen Erkennungen von Heimgeräten zur Laufzeit demonstriert. Dafür werden mehrere Protokolle, darunter X10, ZigBee und Insteon, in einen realen Test integriert. Die Arbeit behandelt die folgenden Schwerpunkte [KIM u. a. 2012]:

- Analyse einer Reihe von Heimnetzprotokollen hinsichtlich ihrer Erkennungs- und Integrationsanforderungen.
- Eine erweiterbare Home-Gateway-Architektur, die es ermöglicht, heterogene Geräte während der Laufzeit flexibel zu installieren, zu verwalten und darauf zuzugreifen.
- Ein neuartiger Zugangskontrollmechanismus speziell für Smart-Home-Systeme.
- Die Umsetzung des vorgeschlagenen Konzepts, indem gezeigt wird, wie verschiedene Geräte integriert und von Endbenutzern aufgerufen werden können.

Die Architektur und die damit eingesetzte semantische Abstraktionsschicht unterstützt laut den Ergebnisse erheblich die Anwendungsentwicklung.

Mit der Zugriffskontrollrichtlinie wird den Hausbesitzern eine stabile Kontrolle darüber gegeben, mit der die Benutzer auf die smarte Geräte zugreifen können. Das Resultat dieser Arbeit scheint zu zeigen, dass damit die Barriere für Smart Home Systeme gesenkt wird.

## Wireless Architectures for Heterogeneous Sensing in Smart Home Applications: Concepts and Real Implementation

Dieser Beitrag von [VIANI u. a. 2013] diskutiert die aktuellen Trends von drahtlosen Architekturen für Anwendungen im Smart Home Bereich. Aus der Diskussion wurden Vorteile erarbeitet, die anschließend über die Verwendung der drahtlosen Architektur analysiert wurden. Schwerpunkt dabei lag jedoch auf der Schätzung des Benutzerverhaltens.

Allgemein behandelt der Artikel die Vorstellung einer drahtlosen Architektur für intelligentes Energiemanagement und die Überwachung von älteren Menschen, indem die Anwesenheit, die Bewegung und das Verhalten der Bewohner analysiert wird [VIANI u. a. 2013]. Interessant dabei ist das entstandene Konzept der konkreten Softwarearchitektur.

## My House, My Rules: A Private-by-Design Smart Home Platform

Dieses Whitepaper von [ZAVALYSHYN u. a. 2020] stellt eine *Private-by-Design-IoT-Plattform* für Smart Home Umgebungen vor. Mit dem Konzept wird eine typische Architektur für bestehende IoT Plattformen als Grundlage verwendet, die über ein alternatives Design mehr Sicherheit und Kontrolle für den Hauseigentümer bietet. Genutzt wird dabei die von Intel entwickelte Software Guard Extensions (SGX)<sup>1</sup>. Diese Erweiterung ermöglicht es, eine intuitive Sicherheitsabstraktion einzuführen, die den unbefugten Zugriff auf Daten durch nicht vertrauenswürdige IoT Cloud-Anbieter verhindert. Das Konzept wurde in einen Prototyp umgesetzt und evaluiert. Dabei wurden auch eine quantitative Forschung durchgeführt, die aus mehr als 40 Probanden bestand, welche den Prototypen verwendet und anschließend bewertet und beurteilt haben. Die Mehrzahl der Teilnehmer der Feldstudie hielten die Softwareplattform als benutzerfreundlich und die unterstützen Richtlinien durch die Sicherheitsabstraktion für nützlich [ZAVALYSHYN u. a. 2020]. Mit den Richtlinien kann verstärkt die Privatsphäre der Anwender in ihrem Wohnraum geschützt werden. Der Sicherheitsmonitor der Software ermöglicht Endbenutzern eine granulare Kontrolle und Überwachung der Datenflüsse, die durch die IoT Geräte generiert werden, sowie die Verhinderung von potentiellen Datenschutzverletzungen der Hersteller durch die Verwendung einer Datenschutzrichtlinie.

## Fast-prototyping Approach to Design and Validate Architectures for Smart Home

Inhalt des Artikels von [MONTANARO u. a. 2021] ist das Entwickeln eines komplexen Smart Home Systems. Hintergrund dafür ist die kontinuierliche Entwicklung und Kommerzialisierung sämtlicher IoT Geräte und die damit einhergehende Änderung oder Anpassung der Nutzeranforderungen. Dadurch benötigt die Community eine schnelle Lösung, bzw. einen Prototypen, um die Anforderungen der Nutzer zu erfüllen und auf schnell entstehenden Notwendigkeit vorbereitet zu sein.

Grundlage für die Entwicklung der Plattform ist eine aktuelle und solide Studie, die ebenso im Rahmen des Artikels durchgeführt wurde. Die Benutzeranforderungen wurden aus der Studie extrahiert und sind bei der Planung des Konzeptes mit eingeflossen. Bestandteil dieser Arbeit ist unter

---

<sup>1</sup>Eine hardwarebasierte Verschlüsselung von Speicherinhalten, die bestimmten Programmiercode und Daten im Speicher isoliert. <https://www.intel.de/content/www/de/de/architecture-and-technology/software-guard-extensions.html> Abgerufen am 15.05.2022.

anderem die Verwendung von Node-RED<sup>2</sup> und dem MQTT Kommunikationsprotokoll.

Die in der Arbeit vereinfachten formalisierten Interaktionen werden speziell dem Anwender gegenüber durch Node-RED visualisiert und die komplexe Logik somit vereinfacht dargestellt. Die Idee mit dem Verwenden von Node-RED ist ein guter Gedanke der formalisierten Interaktionen und kann auch dem Entwickler programmatisch Schritte erleichtern. Da selbst das Framework verstanden werden muss, ist eine gewisse Komplexität nicht zu verhindern. Dennoch kann der Gedanke in weiteren Forschungsschritten weitergedacht oder optimiert werden. Mit diesem Grundsatz sind gute Lerneffekte zu erzielen, indem diese Technologie zu praktischen Arbeiten im Forschungs- und Bildungsbereich eingesetzt wird.

Alle oben aufgeführten Artikel und Forschungsarbeiten sind keine explizite Referenz, bzw. stehen in keinem vollumfänglichen Verhältnis zu der in dieser Arbeit gestellten Forschungsfrage, da mit dem systematischen Literaturreview keine Literatur gefunden werden konnte, die sich ausschließlich dem zu behandelnden Thema widmet. Die zusammengefassten Arbeiten dienen lediglich als stichpunktartige Übersicht, um einen Einblick zu gewährleisten, welche Forschungsarbeiten mit den Suchtermen (siehe Tabelle 3.1) erzielt wurden. Basierend auf diesen Grundlagen und dem damit vermittelten Wissen konnten unter anderem die Anforderungen verfeinert und gefestigt werden, sowie Ideen für das folgende Konzept in Kapitel (5) aufgegriffen werden.

Durch das systematische Literaturreview wird deutlich, dass der Stand der Technik in die Richtung, in die die Arbeit abzielt, keine aussagekräftigen Ergebnisse zeigt. Mittels weiteren Experten Interviews wird deutlich, dass viele mit der Thematik des Smart Home vertraut sind, jedoch überwiegend nur als Benutzer bestehender Plattformen und Geräte gelten.

Die mit den Experten Interviews erlangten Informationen sind nicht repräsentativ, lediglich eine Erkenntnis im Rahmen dieser Arbeit.

Um einen Gesamtüberblick bezüglich dem Technikstand rundum Smart Home zu vermitteln, wird in folgendem aus einer etwas pragmatischeren Sichtweise der Stand der Technik skizziert.

---

<sup>2</sup>Ein von IBM entwickeltes grafisches Entwicklungswerkzeug mit Baukastenprinzip. Funktionsbausteine können per Verbindungen miteinander verknüpft und als Prozess bearbeitet werden. <https://nodered.org/> Abgerufen am 15.05.2022.

### 3.2.2 Stand der Technik aus Nutzer- und Produktsicht

Der aktuelle Markt bietet bereits viele Möglichkeiten und Alternativen der Umsetzung eines intelligenten Zuhauses. Darunter z.B. kommerzielle in sich geschlossene Systeme, die ausschließlich dem Nutzer als Anwendung verkauft werden, einzelne Intelligente Geräte, die nichts zwangsläufig eine zentrale Plattform benötigen und Plattformen, wie bspw. Home Assistant und openHAB, die eine gewisse Technikaffinität erfordern und voraussetzen, um diese nach belieben einsetzen und konfigurieren zu können. Viele große Unternehmen, darunter Bosch, Telekom, Siemens, Samsung und viele mehr bieten Geräte und Softwarelösungen, die der Nutzer als reine Anwendung benutzt, an. Diese geben keinen Einblick in ihre Software und deren Umsetzung, daher sind diese zum Vergleich nicht weiter zu berücksichtigen.

Neben den bekanntesten, frei verfügbaren Plattformen openHAB und Home Assistant gibt es weitere Lösungen, die eine Plattform zur Verfügung stellen<sup>3</sup>:

- OpenMotics
- Jeedom
- ioBroker
- AGO Control
- Domoticz
- Homebridge.io

Neben den aufgelisteten Produkten gibt es noch weiter, die jedoch nicht weiter aufgeführt werden. Durch die Vielzahl der Angebote und den immer größer werdenden Mehrwert, den eine solche Plattform bietet, nimmt die Forschung und Entwicklung in dem Bereich des IoT zu. Dadurch können immer mehr Anwendungsfälle abgedeckt und realisiert werden, wodurch dem Nutzer die Einsatzmöglichkeiten deutlicher und effektiver erscheinen. Dies wird in Kapitel (4) durch die konkrete Marktanalyse (siehe Abschnitt 4.1) deutlich.

---

<sup>3</sup>Aufstellungen von verfügbaren Open Source Smart Home Plattformen im Jahr 2020. <https://ubidots.com/blog/open-source-home-automation/> Abgerufen am 16.05.2022

## Kapitel 4

# Anforderungsanalyse

Dieses Kapitel befasst sich im Allgemeinen mit der Anforderungsanalyse und -erhebung. Hierbei wird eine Marktanalyse repräsentiert, um das Potential rundum Smart Home aufzuzeigen und ein Gefühl dafür zu vermitteln, welche Anforderungen dabei entstehen können, bzw. bereits bestehen. Die Analyse ist mit repräsentativen Statistiken, Studien und Umfragen belegt. Hauptsächlich wird im Rahmen der Anforderungserhebung auf die Methodiken und Techniken eingegangen, die verwendet werden, um Anforderungen zu identifizieren. Diese dienen als Grundlage für die Konzeption der Steuerzentrale. Bestandteile der Anforderungserhebung sind unter anderem zentrale Prozesse des Requirements Engineering, ein *user-centered Design*, im Deutschen nutzerzentriertes Design, eine *Target Group Analysis*, im Deutschen Zielgruppenanalyse, und die Durchführung von Experteninterviews. Diese Interviews sind nicht repräsentativ und dienen lediglich der weiträumigeren Informationsgewinnung.

Vorab wird sichergestellt, dass im Rahmen des benutzerzentrierten Designs der Softwareentwickler als Nutzer und Anwender im Vordergrund steht, da dieser die Plattform betreibt, bzw. für die Erweiterung der Software als auch für die Anpassungen auf die eigenen Anwendungsfälle zuständig ist.

Damit ein Eindruck entsteht, welches Marktpotential Smart Home Anwendungen haben und welche Anforderungen somit verbunden sind, wird basierend auf gegebenen Studien, Statistiken und Umfragen eine Marktanalyse durchgeführt.

### 4.1 Marktanalyse

Der Markt rundum Smart Home nimmt immer weiter zu. Sei es die Entwicklung von neuen intelligenten Geräten, die Massentauglichkeit von Geräten oder die stetig wachsende Abdeckung von Anwendungsfällen und Übernahme von Aufgaben und Prozessen. Durch die Vielzahl an Produktanbietern und diversen Kommunikationsmöglichkeiten, ist es schwierig eine Lösung für alle Alternativen und Produktausprägungen anzubieten. Hersteller versuchen mit der angebotenen Produktpalette ihr eigenes Ökosystem im Bereich Smart Home zu erstellen, um die Nutzer abhängig zu machen. Der repräsentativen Studie von Deloitte zufolge ist jedoch eine Insellösung bei den Nutzern in Deutschland nicht gefragt [WAGNER u. a. 2018]. Befragt wurden 2000 Konsumenten zwischen 19 und 75

Jahren. Einem geringen Anteil von 22 Prozent der Befragten ist die Erweiterbarkeit des Systems mit Produkten anderer Hersteller weniger, bzw. nicht wichtig. Im Gegensatz dazu empfinden 43 Prozent der Befragten die Erweiterbarkeit als wichtig und 28 Prozent als sehr wichtig [WAGNER u. a. 2018]. Demnach müssen die Hersteller eine flexiblere Einsetzbarkeit gewährleisten, damit solche Systeme den Marktdurchbruch erlangen. Dadurch wird die Entwicklung von Plattformen komplexer und umfangreicher. Beispielsweise sind die am weit verbreitetsten Open Source Plattformen, openHAB und Home Assistant, sehr komplex und bilden ein großes Ökosystem ab, da stetig der Zuwachs an integrierbaren Geräten zunimmt und damit der Funktionsumfang steigt.

#### 4.1.1 Allgemeine Marktsituation und Marktprognose

Derzeit gibt es viele Anbieter für intelligente Produkte. Diese bieten zum einen einzelne Geräte an, die in beliebige Plattformen integriert werden können und zum anderen ein eigenes Ökosystem, sofern der Anwender mehrere Produkte des Anbieters nutzen möchte. Dennoch ist in den meisten Fällen die Konfiguration der Geräte nur auf den hauseigenen Plattformen möglich. Somit kann der Nutzer nicht alle Komponenten ausschließlich über eine Plattform konfigurieren und steuern.

Eine repräsentative Umfrage der Statista Global Consumer Survey (SGCS) mit 1384 Teilnehmern, welche im April 2022 veröffentlicht wurde, zeigt, welche Anbieter in Deutschland am meist verbreitetsten sind, bzw. welche die Nutzer am häufigsten kaufen. An oberster Stelle steht Philips und Samsung mit jeweils 25 Prozent und an dritter Stelle Bosch mit 23 Prozent. Weitere Anbieter können dem Diagramm im Anhang (siehe C) entnommen werden. Dabei sind jedoch weitaus nicht alle Hersteller und Anbieter aufgelistet. Detaillierter wird an dieser Stelle jedoch nicht eingegangen.

Der aktuelle Markt für intelligente Produkte ist in sechs primäre Segmente aufgeteilt, die jeweils andere Anwendungsfälle abdecken [LASQUETY-REYES 2021]:

- Kontrolle und Konnektivität: Gateways die alle Geräte jeglicher Segmente kontrollieren, intelligente Lautsprecher mit dem Fokus zur Kontrolle und der digitalen Unterstützung und bspw. intelligente Steckdosen
- Intelligente Geräte (Smart Appliances): Kühlschrank, Waschmaschine und Geschirrspüler, Kaffeemaschine, Mikrowelle und Staubsaugerroboter
- Sicherheit: Bewegungs-, Wasser- und Rauchmelder, Kameras und Türschlösser
- Heimunterhaltung (Home Entertainment): Fernseher, Entertainment-Systeme
- Komfort und Licht: Intelligente LEDs, Fenster- und Tür-Sensoren etc.
- Energiemanagement: Thermostate und Regler, Luftqualitätsmesser etc.



Die aufgelisteten Segmente werden von vielen Herstellern bedient. Darunter sind der folgenden Abbildung die repräsentativen Schlüsselanbieter zu entnehmen:

Representative Smart Home key players by type and segment<sup>1</sup>

	Control and Connectivity	Comfort and Lighting	Security	Home Entertainment	Energy Management	Smart Appliances
Dedicated segment companies	Control4 FIBARO Home Intelligence INSTEON LOXONE GIRA	LEDVANCE LIFX BeON home moodnode COMFYLIGHT	ALARM.COM CHUANGO canary ADT Security LUPUS ring eugust vivint.SmartHome	SONOS PURE Roku D. DEFINITIVE TECHNOLOGY	tado° ecobee climote I nest netatmo	ECOVACS ROBOTICS Robot neato robotics
Players entering the market from foreign industries	HomeKit T... Amazon Echo mi NETGEAR	link somfy. hue LEEDARSON SmartThings	AT&T ASSA ABLOY SCHLAGE Gigaset	B&O BANG & OLUFSEN Apple tv logitech BOSE DENON	hive Danfoss innogy BOSCH Honeywell Schneider Electric	B/S/H/ Haier LG Whirlpool

11 1: Key player overview does not represent the entire market landscape  
Sources: Statista Digital Market Outlook, as of April 2021

Abbildung 4.1: Übersicht der repräsentativen Schlüsselanbieter [LASQUETY-REYES 2021]

Die Übersicht deckt jedoch nicht alle Anbieter ab und spezialisiert sich in diesem Fall auf die bekanntesten und die am Markt etablierten.

Laut den von Statista veröffentlichten Daten war die USA mit einem Umsatz von 28,86 Milliarden US-Dollar der größte Smart Home Markt im Jahr 2021, wogegen Deutschland eine Umsatz von 6,59 Milliarden US-Dollar erzielte. Zu berücksichtigen sind dabei jedoch die Demographische Lage als auch die Bevölkerungsdichte. Diese Aufstellung steht in keinem direkten Vergleich und dient lediglich zur Veranschaulichung und zur Unterscheidung der Marktanteile. Deutlich wird dabei jedoch, dass das Marktwachstum prozentual ähnlich ansteigt.

Der Markt-Prognose in Abbildung (4.2) ist zu entnehmen, dass bis 2026 sich der Umsatz nahezu verdoppeln wird. Der Darstellung (4.3) der einzelnen Segmente kann entnommen werden, dass der Weltmarkt von 2021 bis 2026 um ca. 100 Prozent zunimmt. Die Zeitspanne von 2019 bis 2026 stellt einen durchschnittlichen Zuwachs von 17,4 Prozent dar. Anhand der Prognose und des Berichts von Statista ist deutlich zu sehen, dass der Smart Home Markt in den nächsten Jahren erheblich wachsen wird. Prognostiziert ist ein globaler Marktwert bei ca. 207,8 Milliarden US-Dollar bis 2026.



Abbildung 4.2: Umsatz-Prognose von Deutschland und den USA [LASQUETY-REYES 2021]

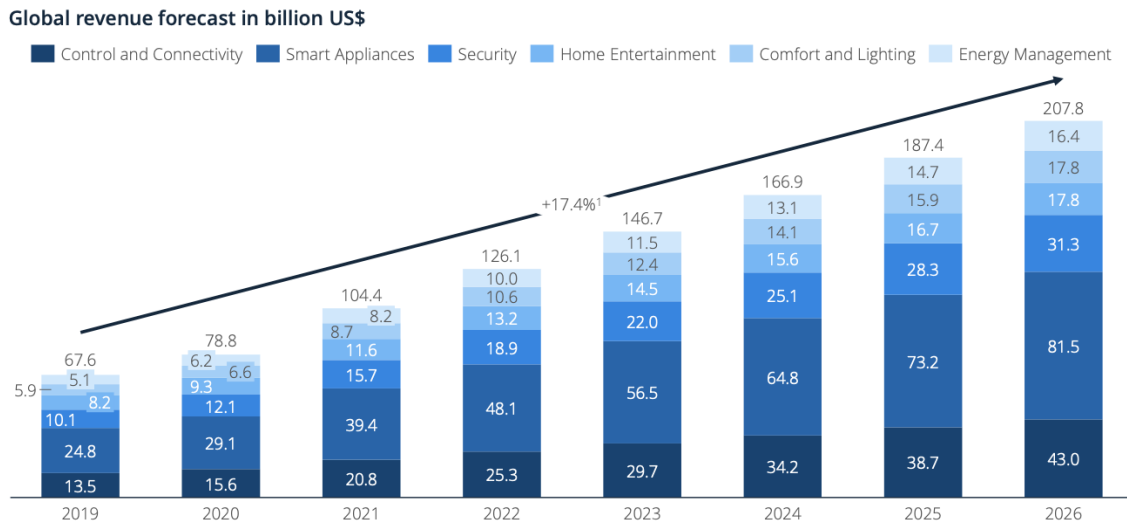


Abbildung 4.3: Globaler Smart Home Marktwert [LASQUETY-REYES 2021]

## Schlüsseltechnologien und Barrieren

Unter den Schlüsseltechnologien im Smart Home sind Komponenten zu verstehen, die den Gedanken eines intelligenten Wohnraumes und Gebäudes forcieren. Dazu zählen unter anderem die Spracherkennung, die bei Sprachassistenten, darunter bspw. Amazon Alexa, Apple HomePod (Siri) und dem Google Nest, eingesetzt wird, sowie Artificial Intelligence (AI) und Künstliche Intelligenz (KI) zur Analyse, Auswertung und Optimierung von Verhaltensmustern und weiteren Analysezielen [LASQUETY-REYES 2021]. Obwohl die Spracherkennung das Wachstum des Marktes ankurbelt, werden die herkömmlichen Interaktionsmöglichkeiten, z.B. die Kontrolle über Berührung durch Touch-Displays, weiter bestehen und weiterhin eine wichtige Art des Gerätezugriffs darstellen [AHNER und PROBST 2022]. Mit dem Einsatz von KI und AI werden Prozesse noch autonomer und können den Komfort aus den Analysen je nach Bedürfnis individuell gestalten. Zu berücksichtigen ist dabei jedoch die dafür geeigneten Anwendungsfälle und die Bereitschaft der Nutzer, in wie fern diese das Analysieren von Verhaltensmustern akzeptieren und zulassen.

### Fehlender Interoperabilität

Die Kommunikation von Smart Home Geräten findet über drahtlose Netzwerke auf Brandbreiten statt, die oft nicht miteinander kompatibel sind. Der hauptsächliche Grund dafür ist, dass Unternehmen ein Protokoll für einen bestimmten Zweck entwickeln und dieses darauf abgestimmt ist den Anwendungsfall abzudecken und um eine Markteintrittsbarriere zu schaffen, sodass der Wechsel zwischen Anbietern erschwert wird [LASQUETY-REYES 2021]. Die damit einhergehende Schwachstelle eines Smart Home ist, dass die Geräte mit den bereits entwickelten Protokollen genutzt werden. So ist die Kommunikation über verschiedene Protokollen nicht vorgesehen. Dadurch entsteht die fehlende Interoperabilität, die der Anwender jedoch für eine Smart Home Lösung ansieht und auch als äußerst sinnvoll betrachtet. Die Einführung von Bluetooth LE Mesh ist eine aktuelle Entwicklung, um der Herausforderung der Bewältigung der Interoperabilität einen Schritt näher zu kommen. Dennoch muss der Anwender prüfen, welche Geräte miteinander kompatibel sind [LASQUETY-REYES 2021]. Bei cloudbasierten Sprachdiensten muss ebenso die Kompatibilität geprüft werden. Dadurch wird dem Nutzer neben der Ungewissheit der Datensicherheit und der Privatsphäre ein weiterer Aspekt geliefert, der die Nutzung von Smart Home Lösungen für bestimmte Anwendergruppen immer noch in Frage stellt.

Derzeit häufig verwendete Protokolle sind unter anderem Bluetooth, Wireless Local Area Network (WLAN) (WiFi), KNX, ZigBee, Z-Wave, MQTT und weitere (siehe 2.4). Um Beispielsweise mit ZigBee über MQTT kommunizieren zu können, gibt es ein Framework, welches die Interoperabilität der beiden Protokolle ermöglicht. Dies ist das sogenannte *ZigBee2MQTT* Framework<sup>1</sup>.

---

<sup>1</sup>Erschafft eine Brücke zwischen ZigBee und MQTT. <https://www.zigbee2mqtt.io/> Abgerufen am 23.05.2022

## 4.2 Zielgruppenanalyse

Zur Analyse der Zielgruppen, die Smart Home Lösungen nutzen, bzw. die als Anwender des im Rahmen dieser Arbeit entstehenden Konzeptes gelten, erfolgt in diesem Abschnitt eine Zielgruppenanalyse. Hierbei wird zwischen zwei Gruppen stark differenziert. Zum einen erfolgt die Benutzer-Analyse, die aufzeigt, welche Zielgruppen im allgemeinen Kontext Smart Home adressiert werden, und zum anderen die Anwender-Analyse, die sich konkret der Zielgruppe widmet, die in dieser Arbeit adressiert wird.

### 4.2.1 Ziel der Zielgruppenanalyse

Ziel einer Zielgruppenanalyse<sup>2</sup> ist die Identifizierung der Personengruppen, die als potentielle Nutzer eines Produktes oder eines Marktsegmentes gelten. Diese Methodik ist ein relevantes Werkzeug in der Produktkonzeption und -entwicklung als auch in der Marktforschung. Maßnahmen und Anforderungen können aus der Zielgruppenanalyse abgeleitet und erarbeitet werden.

Ein Weiteres Ziel ist das bessere Kennenlernen der Zielgruppe, um dadurch deren Bedürfnisse und Interessen genauer zu identifizieren und zu betrachten.

### 4.2.2 Zielgruppe Benutzer

Die in der Marktanalyse (4.1) identifizierten Segmente werden in der Abbildung (4.4) nochmals aufgegriffen. Hierbei wird in der repräsentativen Statistik und Prognose der Statista GmbH die Nutzung des jeweiligen Segments veranschaulicht. Der Fokus dieser Prognose liegt auf der verstärkten Vertretung eines Segments in einem Smart Home in Deutschland. Die derzeit am meisten eingesetzten Segmente sind *Vernetzung und Steuerung* (rot gekennzeichnet) und *Komfort und Licht* (gelb gekennzeichnet). Das am wenigsten genutzt Segment stellt die *Gebäudesicherheit* (schwarz gekennzeichnet) in der Prognose dar. Im Jahr 2021 lag die Nutzung von Geräten des Segments *Vernetzung und Steuerung* bei 6,6 Millionen Nutzer, dicht gefolgt von dem Segment *Komfort und Licht* mit 6,5 Millionen Nutzer. Dagegen liegt das Segment *Home Entertainment* im Jahr 2021 bei 3,8 Millionen Nutzer.

Die bis 2026 veröffentlichte Prognose gibt vor, dass die jeweiligen Segmente stark zunehmen und sich jeweils vervierfachen.

---

<sup>2</sup>Beschreibung der Zielgruppenanalyse und mögliche Durchführungsschritte. <https://www.eology.net/magazine/target-group-analysis> Abgerufen am 24.05.2022.



Quelle: Statista



Abbildung 4.4: Anzahl Smart Home nach Segment [LASQUETY-REYES 2021]

Das Resümee der Prognose zeigt, dass eine breitere Masse an Personen bereits Komponenten der Segmente *Vernetzung und Steuerung* und *Komfort und Licht* nutzen.

Im Hinblick auf die demographischen Statistiken wird in der Zielgruppenanalyse deutlich, welche Gesellschaftsschicht im Jahr 2021 am stärksten vertreten ist. Diese erstreckt sich über eine Altersspanne von 25 bis 54 Jahren. Wobei der Abbildung (4.5) zu entnehmen ist, dass der Schwerpunkt im Alter zwischen 45 und 54 Jahren mit 22,7 Prozent. Die Balance nach Geschlecht liegt bei einem Anteil von 45,6 Prozent weiblichen Befragten und 54,4 Prozent männlichen Befragten, demnach eine geringe Ungleichheit. Die Auswertung nach Einkommen zeigt, dass 35,4 Prozent der Befragten mit mittlerem Einkommen Komponenten eines Smart Home nutzen, dicht gefolgt von Personen mit hohem Einkommen, diese liegen bei 34,9 Prozent.

Demzufolge ist eine große Masse adressiert, die jedoch nach den Segmenten (siehe Abbildung 4.4) wiederum eingeschränkt werden kann.

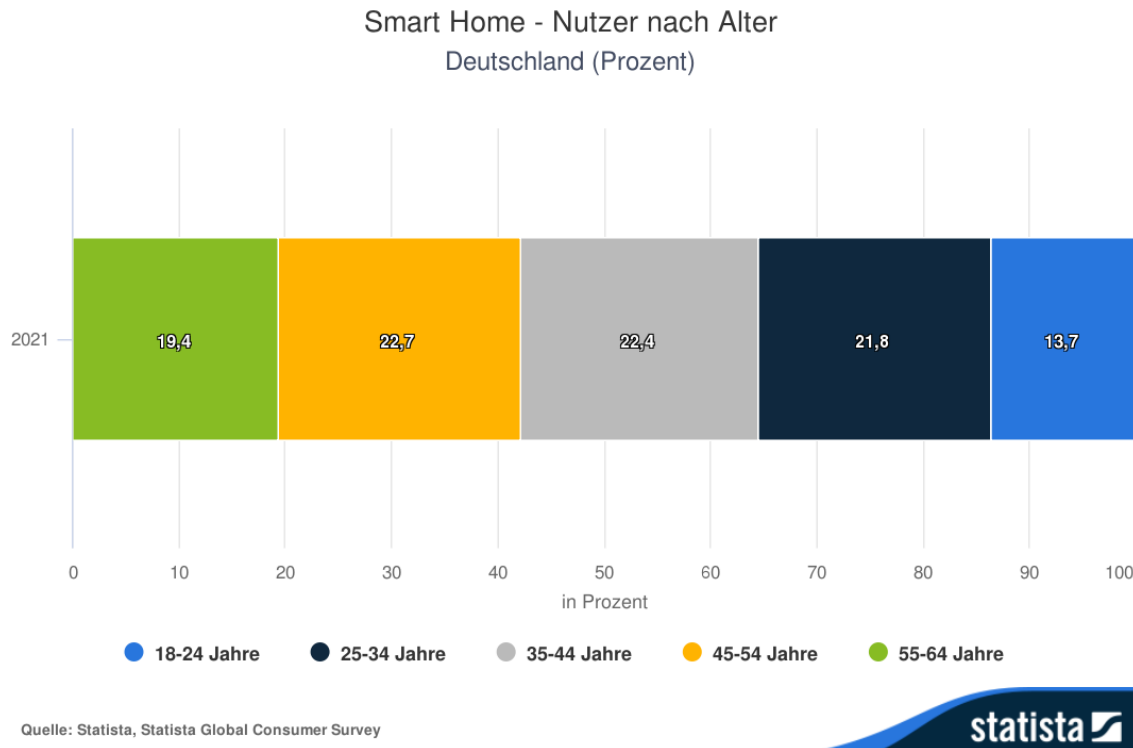


Abbildung 4.5: Smart Home Nutzer nach Alter [LASQUETY-REYES 2021]

Die aufgeführten Statistiken und Prognosen zeigen den Markt rundum Smart Home auf und welches Potential für die nächsten Jahre prognostiziert wird. Im Hinblick auf die Zielgruppe lässt sich sagen, dass eine breite Masse fokussiert werden kann, die jeweils unterschiedliche Voraussetzungen und Bedürfnisse haben. Diese Analyse zeigt jedoch den gesamten Markt auf, um einen Einblick zu gewährleisten, wie stark Smart Home momentan in Deutschland, den Vereinigten Staaten (USA) und dem Rest der Welt vertreten ist. Um einen konkreteren Einblick zu gelangen, wird in nachfolgendem Abschnitt auf die Zielgruppe der Anwender, die in dieser Arbeit fokussiert werden, eingegangen.

### 4.2.3 Zielgruppe Anwender

Der Benutzergruppe wird die Anwenderzielgruppe gegenübergestellt, diese können jedoch ebenso Benutzer der Anwendung werden. Im Umkehrschluss ist ein gewisses Maß an IT-Affinität vorausgesetzt, sodass Benutzer nicht gleich Anwender sein können. Es wird eine grundlegende Kenntnis der Programmierung erfordert, im Rahmen dieser Arbeit der Umgang mit der Programmiersprache Java. Einer Umfrage der Developer Nation<sup>3</sup> zufolge, benutzen von ca. 12.5 Tausend Befragten im Jahr 2021 35.8 Prozent die Programmiersprache Java. Somit kann ein Rückschluss erfolgen, dass Java eine der am häufigsten verwendete Programmiersprache ist.

<sup>3</sup>Umfragen, Statistiken und Berichte rundum Softwareentwickler <https://www.developernation.net/developer-reports/dn21> Abgerufen am 27.05.2022

Im Fokus der Arbeit steht der Softwareentwickler als Anwender, welcher das System betreibt und auf die individuellen Bedürfnisse anpasst.

Laut des Berichts der Developer Nation<sup>4</sup> des dritten Quartals 2021 gibt es zu diesem Zeitpunkt 26.8 Millionen Softwareentwickler weltweit. Davon sind dem Bericht von Daxx<sup>5</sup> zufolge 0.9 Millionen in Deutschland angesiedelt.

Eine vergleichbare Anwendergruppe ist die der Softwarelösung openHAB. Diese ist weltweit bekannt und zählt unter die beliebtesten und am meist genutzt open source Lösungen. Anhand den eigenen Statistiken kann die Community<sup>6</sup> der openHAB Foundation ca. 42 Tausend Benutzer und Anwender vorweisen. Dem GitHub Repository der *openhab-core* Software sind 73 Mitwirkende zugeschrieben, die dazu beitragen die Software weiterzuentwickeln, bzw. zu verbessern und zu stabilisieren.

Damit die adressierte Zielgruppe etwas greifbarer wird und die Anwendungsschritte der *target group analysis* und dem *user-centered design* eingehalten werden, sind Persona<sup>7</sup> entwickelt worden. Diese geben die Anwenderzielgruppe wieder und geben einen Eindruck über die Personen, die sich mit dem Softwareprodukt, bzw. mit dem Konzept auseinandersetzen. Die entstandenen Persona sind dem Anhang beigelegt (siehe REFERENZ TBD).

### 4.3 Anwendungsfälle - Use Cases

Für die Erhebung und Ausarbeitung von Anforderungen an die einfache Handhabung der formalisierten Interaktionen des Softwareentwicklers zur Erweiterung des Systems und dessen Regeldefinitionen werden Anwendungsfälle, sogenannte Use Cases, definiert, bewertet und auf ihre Funktionalität geprüft. Diese wurden im Rahmen des Requirements Engineering dokumentiert. Das RE nach [POHL und RUPP 2021] gibt Schablonen, Vorgehensmodelle und Aufgaben vor, die dabei helfen, den Kontext des Projektes zu erläutern und aus den anliegenden Sachverhalten und Anwendungsfällen die Anforderungen abzuleiten. Nach der strukturierten RE Methode Task-oriented Requirements Engineering (TORE) [ADAM, RIEGEL und DOERR 2014] werden die Aufgaben, die Systemfunktionen und die Interaktionen deutlich. Zum besseren Verständnis des Kontextes und der daraus resultierenden Anforderungen werden die jeweils generierten Anwendungsfälle in den folgenden Abschnitten erläutert.

---

<sup>4</sup>Umfragen, Statistiken und Berichte rundum Softwareentwickler <https://www.developernation.net/developer-reports/dn21> Abgerufen am 27.05.2022

<sup>5</sup>Zusammenfassung mehrerer Berichte. <https://www.daxx.com/de/blog/entwicklungstrends/anzahl-an-softwareentwicklern-deutschland-weltweit-usa> Abgerufen am 27.05.2022

<sup>6</sup>Website Statistik der openHAB Community. <https://community.openhab.org/about> Abgerufen am 27.05.2022

<sup>7</sup>Eine repräsentative Vorstellung einer Person zu bestimmtem Kontext. <https://www.romanpichler.com/the-persona-template/> Abgerufen am 28.05.2022

### 4.3.1 Check In mit einem Service-Roboter

Der Anwendungsfall des Check Ins mit einem Service-Roboter skizziert das Szenario und den Kontext des Sachverhaltes, als auch die Komponenten, die benötigt werden, um den grundlegenden Aufbau der Regeldefinition und -implementierung sowie der Steuerzentrale darzustellen. Daran sind die Bausteine und Anforderungen zu entnehmen, die für das Konzept notwendig sind. Anhand dessen können die Schritte des Anwenders identifiziert werden, die notwendig sind, um die Interaktionen für den Softwareentwickler zu identifizieren und so zu verallgemeinern, dass die formalisierten Interaktionen einfach zu handhaben sind. Des weiteren soll durch die Komplexität der Anwendungsfälle auch ersichtlich werden, welche fundamentalen Komponenten und Konzeptentscheidungen notwendig sind, damit über die Steuerzentrale solche Anwendungsfälle abgebildet werden können.

Der Check In erfordert drei Teilnehmer und eine vorausgesetzte Bedingung, die bereits implementiert ist. Als Teilnehmer zählt eine Person, die den Check In erfährt, ein Service-Roboter, der die Begrüßung und das Check In als Anweisung durchführt und die Steuerzentrale, die den Ablauf koordiniert. Als bereits implementierter Vorgang wird die Authentifizierung der Person über eine Kamera an der Eingangstür vorausgesetzt. Diese veröffentlicht nach erfolgreicher Authentifizierung eine Nachricht an einen Nachrichten-Broker. An dieser Stelle hört die Steuerzentrale auf die veröffentlichte Nachricht und konsumiert diese. Mit dem Ereignis ist der Auslöser für die weiteren Schritte gegeben. Nach dem Erhalt der Nachricht wird diese zugeordnet und die dafür vorgesehene Regel angestoßen. Mit Beginn der Anweisung wird der Service-Roboter an die Tür geschickt, um die Person zu begrüßen. Parallel dazu wird über eine Schnittstelle zu einer internen Software zur Büroplatzbuchung abgefragt, ob die authentifizierte Person bereits einen Büroplatz gebucht hat. Die Information wird in weiterem Schritt für den Check In benötigt. Sobald der Service-Roboter an der Tür angekommen ist, öffnet sich diese und die Person kann eintreten. Wenn die zu begrüßende Person vor dem Service-Roboter steht, kann dieser mithilfe einer integrierten Kamera erkennen, dass eine Person gegenübersteht. Zu diesem Zeitpunkt wird zur Einfachheit damit gerechnet, dass die Person, die vor dem Service-Roboter steht auch diejenige ist, die an der Tür authentifizierte wurde. In zukünftiger Arbeit könnte die Person auch über die Kamera des Roboters erneut authentifziert werden, damit die Korrektheit gegeben ist. Dies ist zu diesem Anwendungsfall jedoch nicht erforderlich und wird deshalb an dieser Stelle nicht weiter verfolgt.

Nachdem der Roboter die Person erkannt hat, startet die formale Begrüßung. Anschließend wird die Information der vorherigen Büroplatzbuchung verwendet, um das Einchecken der Person zu starten. Wurde bei der Abfrage eine Buchung für die Person gefunden, so kann sich diese als anwesend einchecken und an den Arbeitsplatz gehen. Ist jedoch keine Buchung gefunden worden, so kann ein freier Platz über den Roboter reserviert, bzw. gebucht werden. Die Buchung wird anschließend an die Steuerzentrale zurückgegeben und über diese in das Buchungsportal eingepflegt. Nachdem dieser Schritt abgearbeitet ist, endet der Anwendungsfall und der Service-Roboter wird an seine Ausgangsposition, bzw. an dessen Ladestation geschickt. Somit ist das Szenario abgearbeitet und der Roboter steht für weitere Aufgaben zur Verfügung.

Zur besseren Veranschaulichung des Anwendungsfalls sind im Anhang (SIEHE REFERENZ TBD) eine weitere Aufgabenbeschreibung, sowie die User Story als auch ein Use Case-, Sequenz- und



Ablaufdiagramm zu finden.

### 4.3.2 Notfall-Evakuierung mit einem Service-Roboter

Mit dem Szenario der Notfall-Evakuierung wird ein weiterer Anwendungsfall definiert, welcher dabei helfen soll, die Anforderungen zu identifizieren, die für die Bereitstellung des Frameworks der Steuerzentrale benötigt werden. Objektiv betrachtet gibt es bei diesem Fall ebenso einen Auslöser, als Beispiel einen Sensor, bspw. einen Rauch- oder Gasmelder, der eine Nachricht veröffentlicht, die über die Steuerzentrale konsumiert, verarbeitet und dadurch die weiteren notwendigen Schritte eingeleitet werden. Nachdem die Nachricht von der Steuerzentrale erhalten wurde, wird darauf basierend die Regel und die darauffolgende Aktion angestoßen. Die Steuerzentrale muss über die API Schnittstelle des internen Büroplatzbuchungssystems alle eingetragten Personen und Platzbuchungen abfragen und zwischenspeichern. Die Informationen werden genutzt, um den Service-Roboter an die Arbeitsplätze der jeweilig eingetragten Personen zu schicken, sodass dadurch diese über den Notfall informiert, bzw. gebeten werden, das Gebäude zu verlassen. Wird eine Person an einem Arbeitsplatz erkannt, so kann diese über den Sachverhalt informiert werden. Ist jedoch der Arbeitsplatz leer, soll der Service-Roboter den nächsten Arbeitsplatz ansteuern. Nachdem alle Plätze von dem Service-Roboter abgefahren wurden, soll dieser über die restliche Bürofläche fahren und nach Personen suchen. Die Erkennung der Person wird durch die Kamera des Roboters durchgeführt. Abschließend, wenn alle Plätze und die Bürofläche abgefahren wurden, soll der Roboter an eine zentrale Stelle im Büro fahren und ohne Unterbrechung eine Durchsage starten und diese solange wiederholen, bis eine Person den Vorgang beendet. Mit der Beendigung der Dauerschleife ist das Szenario abgeschlossen und der Roboter kann an seine Ausgangsposition zurückgeführt werden, sofern dies umgebungsbedingt noch möglich ist. Somit ist der zweite Anwendungsfall definiert. Die textuelle Schilderung des Szenarios wird mit im Rahmen des RE vorgesehenen Diagrammen und Visualisierungen gestützt. Diese sind dem Anhang (SIEHE REFERENZ TBD) zu entnehmen.

Die Anwendungsfälle wurden so gewählt, da diese eine gewisse Komplexität mit sich bringen, die es mit der Steuerzentrale abzudecken gilt.

Aus dem Kontext beider Anwendungsfälle, der vorangestellten Zielgruppenanalyse und den Experteninterviews werden in Abschnitt (4.5) die daraus identifizierten Anforderungen für die Steuerzentrale aufgeführt.

## 4.4 Experteninterview

Zur Analyse und Erhebung von Anforderungen, die sich an das System richten, werden Experteninterviews durchgeführt. Dabei wird sich, wie in dem Abschnitt (1.4.1) beschrieben, an dem unstrukturierten Ansatz der Führung eines solchen Interviews orientiert [ROBSON 2002]. Infolgedessen werden keine konkreten offenen oder geschlossenen Fragen gestellt. Der Aufbau des Interviews wird in weiteren Schritten aufgegriffen.

Die Ergebnisse der Interviews sind nicht repräsentativ, da sie im Rahmen dieser Arbeit keine große Masse abdecken. Diese dienen lediglich der weiträumigeren Informationsgewinnung und dem

Sammeln mehrerer Meinungsbilder, um aus allen Ideen, Gedankenanstößen und Meinungen ein objektives Gesamtbild zu erzeugen und daraus viele Anforderungen zu gewinnen und abzuleiten.

#### **4.4.1 Ziel des Experteninterviews**

Ziel des Experteninterview ist die Informationsgewinnung aus bestimmten Sachverhalten, die nicht leicht auf anderem Wege zu beschaffen sind. Der Experte dient als Informationsträger und kann dabei helfen seine Sicht auf den Sachverhalt wiederzugeben. Dadurch können Meinungen eingeholt und Erfahrungen ausgetauscht werden. Diese sind oft wichtige Informationen zur Erhebung von Anforderungen einem bestimmten Sachverhalt gegenüber.

#### **4.4.2 Aufbau des Experteninterviews**

Die Experteninterviews wurden im Gesamten als unstrukturierte Interviews durchgeführt. Lediglich die Rahmenbedingungen sowie der Einstieg in das Interview waren vorgegeben und bei jeder interviewten Person ähnlich. Zu Anfang des Gesprächs wurde der Kontext und die Intension erläutert, damit der Experte die Situation und die Absichten kennen lernt und die eigentliche Herausforderung erkennt. Grundlage dafür war die Erläuterung der Zielsetzung (1.3) der Arbeit, um so die Intension zu verdeutlichen. Mit den identifizierten Anwendungsfällen (4.3), die als Basis zur Extraktion von Anforderungen und als potentiell umsetzbare Funktionalitäten gelten, wurden Szenarien veranschaulicht, die dabei helfen, das Anwendungsumfeld zu konkretisieren. Nach der Schilderung des Kontextes und der zugrundeliegenden Ausgangspunkten wurde das Gespräch in Richtung Anforderungen gelenkt. Hierbei lag der Fokus auf der Erhebung der Ideen, Sichtweisen und Meinungen, die als Grundlage für Anforderungen dienten oder gar direkte Anforderungen an das System ergaben. Dabei war der Ausgang des Gesprächs offen. Falls während eines Gesprächs der Fokus verloren ging, bzw. Exkurse ein zu großes Ausmaß annahmen, wurde wieder auf die vorliegende Sachlage aufmerksam gemacht und der Fokus erneut auf die Anforderungen geworfen. Schwerpunkte bei den Interviews waren zum einen, welche Anforderungen gelten, um eine einfache Handhabung der formalisierten Interaktionen für den Softwareentwickler zu gewährleisten und zum anderen die Funktionalitäten, die der Experte dem System gegenüber sieht, um ein Regelwerk für ein intelligentes Büro zu erstellen. Ebenso wurden nichtfunktionale Anforderungen, die ein solches System mit sich bringen soll, adressiert. Die zusammengefassten Anforderungen und daraus abgeleiteten Bedingungen sind dem Abschnitt (4.5) zu entnehmen.

#### **4.4.3 Zusammenfassung der Experteninterviews**

In Summe wurden insgesamt vier Experteninterviews durchgeführt. Jedes Gespräch war individuell und hatte dementsprechend einen anderen Verlauf, bzw. ein anderes Ergebnis. Dennoch konnte der inhaltliche Fokus gewahrt und verschiedene Meinungsbilder eingeholt werden. Jeder befragte Experte konnte zu dem anliegenden Sachverhalt seine Meinung äußern und wichtige Informationen und Sichtweisen mitteilen. Die erhobenen Informationen wurden analysiert, aufbereitet und als Anforderungen dokumentiert. Die dabei entstandenen Informationen und Anforderungen werden in folgendem Abschnitt aufgezeigt.

## 4.5 Anforderungen

In Folge der vorangestellten Literaturrecherche, der Markt- und, Zielgruppenanalyse, der entwickelten Anwendungsfälle und der durchgeführten Experteninterviews ergaben sich Anforderungen an das System als auch Bausteine, die in der Konzeption und der anschließenden prototypischen Implementierung einer solchen Software essentiell sind. Da der Fokus dieser Arbeit auf der einfachen Handhabung der formalisierten Interaktionen für den Softwareentwickler liegt, werden spezifische Anforderungen, die in diese Richtung adressiert sind, höher priorisiert als allgemeingültige Anforderungen, die mehr das Verhalten einer solchen Software beschreiben.

Den folgenden Tabellen sind die Anforderungen zu entnehmen:

Anforderung	Priorität	Quelle
Es soll eine Struktur vorgegeben werden, mit der einfach und formell Regeln und Prozesse implementiert werden können.	Essentiell	Experteninterview
Das Konstrukt von Regeln soll allgemeingültig sein, dass der Entwickler die Logik (Bedingung, Aktion & Zusätze) implementieren muss.	Essentiell	Experteninterview
Nachrichten und Auslöser über Nachrichten sollen empfangen und verarbeitet werden können.	Essentiell	Experteninterview, Anwendungsfall
Es soll eine allgemeingültige Programmiersprache verwendet werden.	Mittel	Experteninterview, Zielgruppenanalyse
Komponenten müssen digital abgebildet werden.	Mittel	Experteninterview, Anwendungsfall
Komponenten und dessen Zustände müssen abgebildet werden.	Hoch	Experteninterview, Anwendungsfall
Aktionen und Regeln sollen nach einem bestimmten Auslöser (Trigger) ausgeführt werden.	Hoch	Experteninterview, Anwendungsfall
Es sollen Auslöser gegeben sein, bzw. implementiert werden, die definierte Aktionen anstoßen	Essentiell	Experteninterview
Aktionen sollen parallel ausgeführt werden können.	Mittel	Experteninterview
Aktionen, welche die gleiche Komponente beanspruchen, sollen nacheinander ausgeführt werden können.	Mittel	Experteninterview
Die Zustände der Komponenten sollten persistiert werden, damit bei einem Systemausfall oder -fehler diese nicht verloren gehen.	Mittel	Experteninterview

Tabelle 4.1: Abstrakte Anforderungen

<b>Anforderung</b>	<b>Priorität</b>	<b>Quelle</b>
Als Programmiersprache soll Java verwendet werden.	Mittel	Experteninterview, Zielgruppenanalyse
Damit Komponenten abgebildet werden können, muss ein Zustandsraum definiert sein.	Essentiell	Experteninterview, Anwendungsfall
Der Zustandsraum muss zur Laufzeit immer zur Verfügung stehen.	Hoch	Experteninterview, Anwendungsfall
Regeln und darauf ausgelöste Aktionen sollen zeitbasiert oder bspw. über MQTT ausgelöst werden. Diese dienen als Auslöser	Hoch	Experteninterview, Anwendungsfall
Nutzung eines Thread Pools zur parallelen Ausführung von Regeln.	Mittel	Experteninterview
Aktionen, welche die gleiche Komponente beanspruchen, sollen nacheinander ausgeführt werden können.	Mittel	Experteninterview

Tabelle 4.2: Konkretere Anforderungen

<b>Funktionale Anforderung</b>	<b>Nichtfunktionale Anforderungen</b>	<b>Quelle</b>
Bereitstellung eines Konstruktes zur Implementierung eines Regelsets.	Zuverlässigkeit: Durch einen Auslöser gestartete Aktionen sollen bei gleichem Auslöser immer die gleiche Aktion anstoßen.	Ex.-Interv.
MQTT Nachrichten werden verarbeitet und an einen Auslöser weitergeleitet.	Die Steuerzentrale muss eine Verfügbarkeit von 99,9% vorweisen.	Ex.-Interv.
Implementierung eines MQTT Subscriber and Producer.	Die Kommunikation erfolgt ausschließlich über MQTT.	AW-Fall
Speicherung des Zustandsraumes in eine Datenbanken.	Single Point of Contact (Implementierung, Anpassung und Erweiterung der Regel und der Logik).	Zielgr.-Analyse
Erweiterungen zu API Anbindungen per MQTT.	Sicherheit: Es werden nur Schnittstellen zur Datenabfrage erlaubt.	Ex.-Interv.
Abbildung von Zustandsinformationen.	Die Steuerzentrale ist nur im eigenen Netz erreichbar.	Ex.-Interv.

Tabelle 4.3: Funktionale und Nichtfunktionale Anforderungen

# Kapitel 5

## Konzeption

In diesem Kapitel wird das erarbeitete Konzept dieser Arbeit dargelegt. Basierend auf den Anforderungen, die aus den Anwendungsfällen, Experteninterviews und der Zielgruppenanalyse erhoben wurden, werden die daraus generierten Überlegungen und Entscheidungen transparent dargestellt. Durch die bereits erfolgten Schritte der Anforderungsanalyse (4) sind erste Aufgaben der Konzeption abgeschlossen.

Zu Anfang des Kapitels wird das allgemeine Ziel eines Konzeptes erläutert. Anschließend wird auf das Anwendungsumfeld des Systems, engl. Framework, (5.2) eingegangen. Darauf folgend wird anhand den zugrundeliegenden Informationen das Architekturkonzept (5.3), sowie das Softwarekonzept (5.4) erläutert. Des Weiteren werden die Hintergründe der Wahl des Frameworks (5.5) aufgezeigt.

### 5.1 Ziel der Konzeption

Das Ziel einer Konzeption ist die Veranschaulichung von abstrakten Ideen, geistiger Entwürfe und Leitideen. Hierbei werden aus den zugrundeliegenden Problemstellungen, Szenarien und Anforderungen Entwürfe und Lösungsmöglichkeiten erarbeitet und identifiziert. Diese helfen bei der Aufstellung von notwendigen Schritten und dienen als Grundlage zur Untermauerung und Darlegung von Entscheidungen. Somit wird Dritten der Kontext, die Domäne und das zu lösende Problem, bzw. die Lösung dargestellt.

Im Rahmen dieser Arbeit ist das Ziel des Konzeptes die Veranschaulichung der Entscheidungsfindung zur Lösung des vorliegenden Sachverhaltes. Das Konzept erarbeitet eine Lösung zur Implementierung einer Anwendung zur Koordination von Regeln und Prozessen innerhalb eines Firmenbüros. Hierbei wird der allgemeine Aufbau der Architektur skizziert und demonstriert, wie eine solche Lösung aussehen kann. Unter Berücksichtigung der Forschungsfrage (siehe Abschnitt 1.2) wird eine Möglichkeit offengelegt, mit der ein Softwareentwickler neue Regeln entwickeln und dem System hinzufügen kann, ohne ein weiteres zu erlernendes Framework zu verwenden. Dabei sollen die notwendigen Schritte und Interaktionen formalisiert und für den Entwickler vereinfacht werden. Mit der Definition der Zielsetzung der Arbeit (siehe Abschnitt 1.3) wird die Abgrenzung deutlich. In folgender Darlegung des Konzeptes wird nochmals konkreter auf den Kontext als auch auf die Intension eingegangen.

## 5.2 Anwendungsumfeld

Grundsätzlich ist der Einsatzort des Frameworks variabel, da die eigentliche Implementierung und Nutzung der Regeln und Prozesse stark abhängig von den Anwendern ist. Dadurch kann sowohl in privatem Smart Home Umfeld als auch in Büroräumen eine solche Instanz mittels dem Framework erstellt werden. Basierend auf den vorangestellten Tätigkeiten, darunter die Anforderungsanalyse und die Eingrenzung auf den Einsatz im Smart Office, liegt der Schwerpunkt auf dem Einsatz in einem smarten Büro.

## 5.3 Architekturkonzept

### 5.3.1 Überlegungen, Anstöße und Herausforderungen

### 5.3.2 Schnittstellen

### 5.3.3 Datenbanken

## 5.4 Softwarekonzept

## 5.5 Auswahl des Frameworks

Im Bereich der Java-Entwicklung gibt es mittlerweile viele Möglichkeiten, um Applikationen, Anwendungen und Frameworks zu entwickeln, die unterschiedliche Präferenzen und Einsatzmöglichkeiten bieten. Dadurch sind, auf den Einsatzbereich bezogen, Vor- und Nachteile im Vergleich ähnlicher Systeme nicht auszuschließen. Eine kleine Auswahl an Systemen wurde getestet und auf deren Brauchbarkeit analysiert und evaluiert. Für diese Evaluation wurden Kriterien ausgearbeitet, mit der die Auswahl des Frameworks eingeschränkt und nach Möglichkeit das passende ergeben soll. Diese sind nach ihrer Relevanz aufgelistet:

1. Freiheiten bei der Nutzung (Anwendung und Gewährleistung von Entwurfsmustern).
2. Nutzung und Bereitstellung von Bibliotheken (Libraries).
3. Bereitstellung einer Plattform (Full Stack).
4. Aktive Community und stetige Weiterentwicklung des Systems.
5. Nutzung des Frameworks in bereits bestehenden Projekten im Bereich Smart Home.
6. Open Source-Projekt, um Flexibilität und weitestgehende Unabhängigkeit zu gewährleisten.
7. Ausschluss von Frameworks, die ausschließlich für die Web-Entwicklung gedacht sind.

Aufgrund der Vielzahl an Frameworks war es im Rahmen dieser Arbeit nicht möglich, alle vorhandenen und in Betracht gezogenen Systeme detailliert aufzuführen. Lediglich die engere Auswahl wird aufgegriffen. Nach ausführlicher Recherche und unter Berücksichtigung von Frameworks, wie bspw. Grails, Quarkus, Blade, und Play, wurden schließlich genau zwei Systeme gegenübergestellt,

die die vorangestellten Kriterien in Gänze erfüllen. Viele in Betracht gezogenen Frameworks finden überwiegend in der Web-Entwicklung Anwendung, bzw. basieren auf dem Spring Framework. Durch diese Erkenntnis wurden viele Systeme nicht weiter berücksichtigt und betrachtet. Die beiden Kernsysteme, die ihren Einsatz und ihre Möglichkeiten rechtfertigen werden in den folgenden Abschnitten kurz erläutert.

### 5.5.1 OSGi

Das OSGi Framework der OSGi<sup>1</sup> Alliance, welches in der openHAB Software eingesetzt wird, klassifiziert eine dynamische Softwareplattform, mit der die Modularisierung und Verwaltung von Applikationen und den dazugehörigen Diensten mittels Komponentenmodell realisiert werden kann [FUNKE 2009]. Bekannte Produkte, die auf der OSGi Plattform laufen, sind neben openHAB unter anderem die Entwicklungsumgebung Eclipse der Eclipse Foundation, Produkte und Softwarelösungen von IBM, Oracle, Adobe und weitere.

Nennenswerte Eigenschaften und Vorteile der Software sind die Modularisierung und Versionierung, das zur Laufzeit organisierte Abhängigkeitsmanagement, das Fernmanagement des laufenden Systems über sogenannte Management Agents und die Nutzung des Serviceorientierten Programmiermodells, Service-Oriented Architecture (SOA)<sup>2</sup>. An dieser Stelle wird das Framework nicht technisch vertieft. Die Funktionsweise und die technisch fundierte Erläuterung kann dem Buch [WUETHERICH u. a. 2008], sowie der Dokumentation [WUETHERICH u. a. 2009] eines ausgearbeiteten Workshops entnommen werden. Nachteile der Plattform sind zum einen der weniger breite Einsatz des Frameworks und zum anderen die kleine Community. Ebenso findet eine Weiterentwicklung der Plattform nur mäßig statt. Dies hat zur Folge, dass die Erweiterung, bzw. die Entwicklung des darauf aufbauenden Systems träge vorstangeht.

### 5.5.2 Spring Boot

---

<sup>1</sup>Ursprung der OSGi Plattform. <https://www.osgi.org/about/> Abgerufen am 19.06.2022

<sup>2</sup>Erläuterung des SOA Modells. <https://www.ibm.com/cloud/learn/soa> Abgerufen am 20.06.2022

# Kapitel 6

## Umsetzung

### 6.1 Implementierung

#### 6.1.1 Auswahl der genutzten Technologien

#### 6.1.2 Aufbau der Architektur

#### 6.1.3 Einbindung der Funktionen abgeleitet von der Konzeption

### 6.2 Ergebnis



# Kapitel 7

## Diskussion und Evaluation

### 7.1 Analyse des Konzepts der Eigenentwicklung

### 7.2 Usability Test

#### 7.2.1 Ziel des Usability Tests

### 7.3 Experten Interview

#### 7.3.1 Ziele des Experten Interviews

### 7.4 Vergleich zwischen Eigenentwicklung und bestehenden Softwarelösungen

## Kapitel 8

## Fazit

Draft

## Kapitel 9

### Ausblick

Draft

# Abbildungsverzeichnis

2.1	Technologische Einordnung von IoT [SIEPERMANN und LACKES 2018]	9
2.2	Exemplarische Darstellung eines IoT-Systems [GILLIS 2022]	10
2.3	Client-Server, Peer-to-Peer und Message Passing Interaktionsparadigma [MINERVA, BIRU und ROTONDI 2015]	12
2.4	Mögliche Anwendungsszenarien im Smart Home [STRESE u. a. 2010]	17
2.5	Technologische Einordnung von Smart Home in Verbindung zu IoT [BENDEL 2021]	18
2.6	Aufbau und Funktionsweise einer Smart Home Infrastruktur	19
2.7	Themenbasiertes Pub/Sub Kommunikationsmodell [HUNKELER, TRUONG und STANFORD-CLARK 2008]	25
2.8	Architektur des Home Assistant Supervisors [SCHOUTSEN 2020]	29
2.9	Architektur des Home Assistant Cores [LI u. a. 2018]	30
2.10	Architektur der openHAB Plattform [KREUZER 2013]	36
2.11	Architektur der openHAB 2.0 Plattform [KREUZER 2016]	37
2.12	OSGi Schichtenarchitektur [KREUZER 2022]	38
3.1	Strategie der Literatursuche	42
4.1	Übersicht der repräsentativen Schlüsselanbieter [LASQUETY-REYES 2021]	51
4.2	Umsatz-Prognose von Deutschland und den USA [LASQUETY-REYES 2021]	52
4.3	Globaler Smart Home Marktwert [LASQUETY-REYES 2021]	52
4.4	Anzahl Smart Home nach Segment [LASQUETY-REYES 2021]	55
4.5	Smart Home Nutzer nach Alter [LASQUETY-REYES 2021]	56

# Tabellenverzeichnis

2.1	Interaktionsparadigmen nach [MINERVA, BIRU und ROTONDI 2015]	12
2.2	Historische Entwicklung vom Internet der Dinge [DURGA, PROF und KUMAR 2020]	14
2.3	Teilsysteme des Smart Home [STRESE u. a. 2010]	16
2.4	Übertragungsmethoden des Smart Home	23
2.5	Stärken und Schwächen der Home Assistant Plattform	31
3.1	Suchprotokoll des Systematischen Literaturreviews	43
3.2	Einschluss- und Ausschlusskriterien des systematischen Literaturreviews	44
4.1	Abstrakte Anforderungen	61
4.2	Konkretere Anforderungen	62
4.3	Funktionale und Nichtfunktionale Anforderungen	62
A.1	Vergleich von MQTT und AMQP [NAIK 2017]	VI
B.1	Vergleich der Plattformen [BRICE 2022] [SUTTNER 2022] [BARCLAY 2020]	VIII

# Liste der Code-Beispiele

2.1	Erzeugung und Veröffentlichung einer Nachricht . . . . .	25
2.2	Empfang und Konsum einer Nachricht . . . . .	25
2.3	Konstrukt zur Regeldefinition über Home Assistant . . . . .	32
2.4	Konstrukt zur Regeldefinition über Xbase . . . . .	39

Draft

# Abkürzungsverzeichnis

<b>IoT</b>	Internet of Things . . . . .	7
<b>IdD</b>	Internet der Dinge . . . . .	7
<b>SH</b>	Smart Home . . . . .	3
<b>IT</b>	Informationstechnologie . . . . .	7
<b>IP</b>	Internetprotokoll . . . . .	8
<b>IIoT</b>	Industrial Internet of Things, dt. Industrielles Internet der Dinge . . . . .	11
<b>P2P</b>	Peer to Peer . . . . .	12
<b>REST</b>	Representational State Transfer . . . . .	13
<b>SOAP</b>	Simple Object Access Protocol . . . . .	13
<b>M2M</b>	Machine to Machine . . . . .	13
<b>API</b>	Application Programming Interface . . . . .	13
<b>HTML</b>	Hypertext Markup Language . . . . .	13
<b>XML</b>	Extensible Markup Language . . . . .	13
<b>JSON</b>	JavaScript Object Notation . . . . .	13
<b>P&amp;G</b>	Procter & Gamble . . . . .	14
<b>RFID</b>	Radio-Frequency Identifitaction . . . . .	14
<b>ITU</b>	International Telecommunication Union . . . . .	14
<b>IIS</b>	Fraunhofer Institut für Integrierte Schaltungen . . . . .	14
<b>RE</b>	Requirements Engineering . . . . .	4
<b>BMWK</b>	Bundesministerium für Wirtschaft und Klimaschutz . . . . .	20
<b>IFA</b>	Internationale Funkausstellung . . . . .	21
<b>MQTT</b>	Message Queue Telemetry Transport . . . . .	23
<b>TCP</b>	Transmission Control Protocol . . . . .	23
<b>TLS</b>	Transport Layer Security . . . . .	23
<b>SSL</b>	Secure Sockets Layer . . . . .	23

<b>AMQP</b>	Advanced Message Queue Protocol . . . . .	26
<b>HTTP</b>	Hypertext Transfer Protocol . . . . .	26
<b>CoAP</b>	Constrained Application Protocol . . . . .	26
<b>PAN</b>	Personal Area Network . . . . .	27
<b>UI</b>	User Interface . . . . .	29
<b>DoD</b>	Definition of Done . . . . .	30
<b>openHAB</b>	open Home Automation Bus . . . . .	33
<b>OSGi</b>	Open Service Gateway initiative . . . . .	33
<b>JVM</b>	Java Virtual Machine . . . . .	37
<b>JAR</b>	Java Archive . . . . .	37
<b>SGX</b>	Software Guard Extensions . . . . .	46
<b>SGCS</b>	Statista Global Consumer Survey . . . . .	50
<b>AI</b>	Artificial Intelligence . . . . .	53
<b>KI</b>	Künstliche Intelligenz . . . . .	53
<b>WLAN</b>	Wireless Local Area Network . . . . .	53
<b>DSL</b>	Domain Specific Language . . . . .	39
<b>TORÉ</b>	Task-oriented Requirements Engineering . . . . .	57
<b>SOA</b>	Service-Oriented Architecture . . . . .	65



# Literatur

- ADAM, Sebastian, Norman RIEGEL und Joerg DOERR [2014]. *A Framework for Systematic Requirements Development in Information Systems*. URL: <https://re-magazine.ireb.org/articles/tore> [besucht am 30.05.2022].
- AHNER, Dr.-Ing. Nicole und Marint PROBST [2022]. *All-Electronics. Marktentwicklung, Player und Trends beim Smart Home*. URL: <https://www.all-electronics.de/markt/marktentwicklung-player-und-trends-beim-smart-home-580.html> [besucht am 23.05.2022].
- ASCHENDORF, B. [2014]. *Energiemanagement durch Gebäudeautomation: Grundlagen - Technologien - Anwendungen*. Springer Fachmedien Wiesbaden, S. 55–57. ISBN: 9783834820327. URL: <https://books.google.de/books?id=-y8pBAAAQBAJ>.
- BALAKRISHNAN, Sumathi, Hemalata VASUDAVAN und Raja Kumar MURUGESAN [Dez. 2018]. „Smart home technologies: A preliminary review“. In: Association for Computing Machinery, S. 120–127. ISBN: 9781450366298. DOI: 10.1145/3301551.3301575.
- BARCLAY, Lewis [2020]. *Home Assistant vs OpenHAB - Which one is better?* URL: <https://everythingsmarthome.co.uk/home-assistant-vs-openhab-which-one-is-better/> [besucht am 20.05.2022].
- BENDEL, Oliver [2021]. *Smart Home*. URL: <https://wirtschaftslexikon.gabler.de/definition/smart-home-54137> [besucht am 23.03.2022].
- BRAJKOVIC, Sandra [2014]. *Telekom arbeitet am intelligenten Haus der Zukunft*. URL: <https://www.welt.de/wirtschaft/webwelt/article125324076/Telekom-arbeitet-am-intelligenten-Haus-der-Zukunft.html> [besucht am 06.04.2022].
- BRICE, Alex [2022]. *Best of open source smart home: Home Assistant vs OpenHAB*. URL: <https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/> [besucht am 01.05.2022].
- BUSCH-JAEGER [2021]. *Busch-Jaeger - Geschichte. 1990er-Jahre - Das intelligente Haus wird Wirklichkeit*. In: *busch-jaeger.de*. URL: <https://www.busch-jaeger.de/geschichte/> [besucht am 06.04.2022].
- CAO, Keyan u. a. [2020]. „An Overview on Edge Computing Research“. In: *IEEE Access* 8, S. 85714–85728. ISSN: 21693536. DOI: 10.1109/ACCESS.2020.2991734.

- DURGA, P V Vijaya, Asst PROF und T Sujith KUMAR [2020]. „A Literature Survey on Internet of Things“. In: *International Journal for Research in Engineering Application Management (IJREAM)* 06, S. 2454–9150. DOI: 10.35291/2454-9150.2020.0717.
- FRAUNHOFER [2021]. *Historie - Die Geschichte des Fraunhofer-inHaus-Zentrums. Das Fraunhofer-inHaus-Zentrum*. URL: <https://www.inhaus.fraunhofer.de/de/ueber-uns/Historie.html> [besucht am 06.04.2022].
- FUNKE, Holger [2009]. *Das OSGi Framework*. URL: <https://entwickler.de/java/das-osgi-framework> [besucht am 19.06.2022].
- GARTNER [März 2022]. *Gartner Hype Cycle*. <https://www.gartner.com/en/research/methodologies/gartner-hype-cycle>.
- GILLIS, Alexander [2022]. *What is the internet of things (IoT)?* URL: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> [besucht am 25.03.2022].
- HUNKELER, Urs, Hong Linh TRUONG und Andy STANFORD-CLARK [2008]. „MQTT-S - A publish/-subscribe protocol for wireless sensor networks“. In: IEEE Computer Society, S. 791–798. ISBN: 9781424417971. DOI: 10.1109/COMSWA.2008.4554519.
- KIM, Ji Eun u. a. [2012]. „Seamless integration of heterogeneous devices and access control in smart homes“. In: S. 206–213. ISBN: 9780769547411. DOI: 10.1109/IE.2012.57.
- KITCHENHAM, Barbara [2007]. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. URL: <https://www.researchgate.net/publication/302924724>.
- KREUZER, Kai [Juni 2014]. *openHAB 1.5 - Release Highlights*. URL: <http://kaikreuzer.blogspot.com> [besucht am 29.04.2022].
- [Jan. 2016]. *openHAB 1.8 and 2.0 beta 1 Release*. URL: <http://kaikreuzer.blogspot.com> [besucht am 30.04.2022].
- [2013]. *openHAB Dokumentation*. URL: <https://github.com/openhab/openhab1-addons/wiki> [besucht am 29.04.2022].
- [2020]. *openHAB Strength*. URL: <https://www.openhab.org/docs/> [besucht am 01.05.2022].
- [Dez. 2012]. *OSGi Connects the World*. URL: <http://kaikreuzer.blogspot.com> [besucht am 29.04.2022].
- [2022]. *OSGi Overview*. URL: <https://www.openhab.org/docs/developer/osgi/osgi.html> [besucht am 01.05.2022].
- LASQUETY-REYES, Dr. Jeremiah [2021]. *Statista. Smart Home Report 2021. Statista Digital Market Outlook - Market Report. (Artikel-Nr. did-41155-1)*. URL: <https://de.statista.com/statistik/studie/id/41155/dokument/smart-home-report/> [besucht am 23.05.2022].
- LI, Yang u. a. [2018]. „Home Assistant-Based Collaborative Framework of Multi-Sensor Fusion for Social Robot\*“. In: ISBN: 9781538673461.
- LUBER, Stefan und Nico LITZEL [2019]. *Definition - Was ist Smart Home*. URL: <https://www.bigdata-insider.de/was-ist-smart-home-a-809018/> [besucht am 08.05.2022].

- [2016]. *Was ist das Internet of Things?* URL: <https://www.bigdata-insider.de/was-ist-das-internet-of-things-a-590806/> [besucht am 22.03.2022].
- MADAKAM, Somayya, R. RAMASWAMY und Siddharth TRIPATHI [2015]. „Internet of Things (IoT): A Literature Review“. In: *Journal of Computer and Communications* 03 [05], S. 164–173. ISSN: 2327-5219. DOI: 10.4236/jcc.2015.35021.
- MELL, Peter und Timothy GRANCE [Sep. 2011]. *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology*, S. 2–3. URL: <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>.
- MINERVA, Roberto, Abyi BIRU und Domenico ROTONDI [2015]. „IEEE<sub>IoT</sub>TowardsDefinitionInternet<sub>o</sub>fThingsRevision1“. In: *Towards a definition of the Internet of Things (IoT)*, S. 59–70. URL: [https://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revision1\\_27MAY15.pdf](https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf).
- MONTANARO, Teodoro u. a. [2021]. „Fast-prototyping approach to design and validate architectures for smart home“. In: *Journal of Communications Software and Systems* 17 [2], S. 177–184. ISSN: 18466079. DOI: 10.24138/jcomss-2021-0005.
- NAIK, Nitin [Okt. 2017]. „Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP“. In: Hrsg. von Nitin NAIK. 2017 IEEE International Systems Engineering Symposium (ISSE), S. 1–6. ISBN: 9781538634035. DOI: 10.1109/SysEng.2017.8088251.
- POHL, K. und C. RUPP [2021]. *Basiswissen Requirements Engineering: Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt.verlag. ISBN: 9783969102473. URL: <https://books.google.de/books?id=iEgqEAAAQBAJ>.
- PRANZ, Bernhard und Markus SCHILLER [Juni 2018]. „Smart Home with openHAB“. In: S. 2–4. URL: [https://www.researchgate.net/publication/342171578\\_Smart\\_Home\\_with\\_openHAB](https://www.researchgate.net/publication/342171578_Smart_Home_with_openHAB).
- ROBSON, Colin [2002]. *Real world research: A resource for social scientists and practitioner-researchers*. Wiley-Blackwell. ISBN: 978-0631213055.
- SCHOUTSEN, Paulus [2020]. *Home Assistant Supervisor*. URL: <https://developers.home-assistant.io/docs/supervisor> [besucht am 01.05.2022].
- SIEPERMANN, Markus und Richard LACKES [2018]. *Internet der Dinge*. URL: <https://wirtschaftslexikon.gabler.de/definition/internet-der-dinge-53187> [besucht am 23.03.2022].
- STRESE, Hartmut u. a. [2010]. *Smart Home in Deutschland. Untersuchung im Rahmen der wissenschaftlichen Begleitung zum Programm Next Generation Media (NGM) des Bundesministeriums für Wirtschaft und Technologie*. Institut für Innovation und Technik (iit). ISBN: 9783897501652.
- SUTTNER, Matthias [2022]. *VERGLEICH: OPENHAB VS. HOME ASSISTANT*. URL: <https://smarthome.msuttner.de/openhab-2/vergleich-openhab-vs-home-assistant/> [besucht am 02.05.2022].
- V., VDE e. [2013]. *Die deutsche Normungs-Roadmap Smart Home + Building*, S. 16–45. URL: [www.dke.de](http://www.dke.de).

- VIANI, Federico u. a. [2013]. „Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation“. In: *Proceedings of the IEEE* 101 [11], S. 2381–2396. ISSN: 00189219. DOI: 10.1109/JPROC.2013.2266858.
- WAGNER, Dr. Gunher u. a. [2018]. *Deloitte. Smart Home Consumer Survey 2018. Ausgewählte Ergebnisse für den deutschen Markt*. URL: [https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/Deloitte\\_TMT\\_Smart\\_Home\\_Studie\\_18.pdf](https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/Deloitte_TMT_Smart_Home_Studie_18.pdf) [besucht am 22.05.2022].
- WU, Chao Lin und Li Chen FU [2012]. „Design and Realization of a Framework for Human–System Interaction in Smart Homes“. In: *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans* 42 [1], S. 15–31. ISSN: 15582426. DOI: 10.1109/TSMCA.2011.2159584.
- WUETHERICH, Gerd u. a. [2008]. *Die OSGi Service Platform*. dpunkt.verlag. ISBN: 978-3-89864-457-0.
- [2009]. *Einführung in die OSGi Service Plattform*. URL: <https://nilshartmann.net/uploads/Powerworkshop-OSGi-JAX2009.pdf> [besucht am 19.06.2022].
- ZAVALYSHYN, Igor u. a. [Dez. 2020]. „My house, my rules: A private-by-design smart home platform“. In: Association for Computing Machinery, S. 273–282. ISBN: 9781450388405. DOI: 10.1145/3448891.3450333.

# Anhang A

Kriterium	MQTT	AMQP
Jahr	1999	2003
Architektur	Client/Broker,	Client/Broker Client/Server
Abstraktion	Publish/Subscribe	Publish/Subscribe Request/respon- se
Header Größe	2 Byte,	8 Byte
Nachrichtengröße	max. 265 MB	Abhängig von Broker/Server
Semantik / Metho- den	Connect, Disconnect, Publish, Subs- cribne, Unsubscribe, Close	Consume, Deliver, Publish, Get, Se- lect, Ack, Delete, Nack, Recover, Re- ject, Open, Close
Zwischenspeicher (Cache) und Proxy	Teilweise	Ja
Standards	OASIS, Eclipse Foundation	OASIS, ISO/IEC
Transport- Protokoll	TCP,	TCP, SCTP
Sicherheit	TLS/SSL	TLS/SSL
Standard-Port	1883/8883 (TLS/SSL)	5671 (TLS/SSL), 5672
Format	binär	Binär
Lizenzmodell	Open Source	Open Source
Support	IBM, Facebook, Eurotech, Cisco, Red Hat Amazon (AWS) etc.	Microsoft, JP Morgan, Bank of Ame- rica, Barclays, Goldman Sachs Cre- dit Suisse

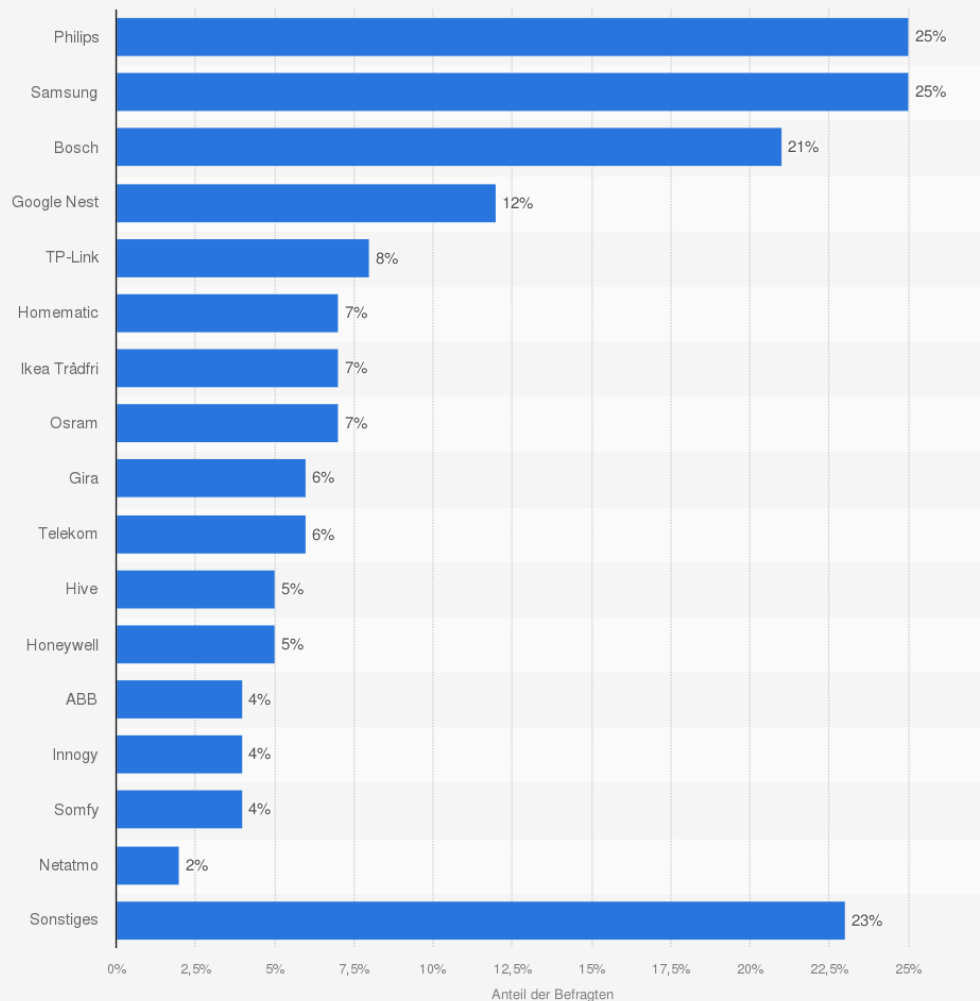
Tabelle A.1: Vergleich von MQTT und AMQP [NAIK 2017]

## Anhang B

Attribut	openHAB	Home Assistant
Architektur	Robustheit und Starrheit, sowie die gewissenhafte Entwicklung	Erfordert häufigere Updates, bietet jedoch eine schnelle Entwicklung und eine viel modernere und anspruchsvollere Architektur [BRICE 2022].
Installation und Wartung	Installation ist genauestens dokumentiert. Initiale Konfiguration muss über die Kommandozeile durchgeführt werden, Updates ebenso.	Ein-Klick-Installationsprozess. Initiale Konfiguration benötigt das Verständnis von YAML-Dateien. Updates können über die UI verwaltet werden [BRICE 2022].
Unterstützte Geräte und Verknüpfungen	Durch die gängigen Smart Home Protokolle werden über 1000 Komponenten unterstützt.	Home Assistant unterstützt ebenso über 1000 Komponenten, wobei die Benutzerfreundlichkeit beim Anlegen und Verwalten bei Home Assistant deutlich angenehmer und leichter erscheint [SUTTNER 2022].
Automatisierungsregeln	Bereitstellung von Automationen durch Xtend. Xtend ist ein flexibler und ausdrucksstarker Java-Dialekt, der sich in Quellcode basierend auf Java 8 kompilieren lässt. Ebenso können über Blockly Automatisierungsregeln erstellt werden. Blockly ist eine clientseitige JavaScript Bibliothek, die visuelle Blöcke mit Anweisungen, Bedingungen uvm. erstellen und editieren lässt. Eine weitere Alternative ist die Verwendung von Node-RED.	Home Assistant bietet die Möglichkeit, Automatisierungen per YAML zu erstellen. Alternativ mit Node-RED. YAML-Dateien sind flexibler, jedoch nicht unbedingt benutzerfreundlich.
User Interface	openHAB hat in den User Interface viele Duplikate und zu viele ähnliche Optionen zur Verfügung bei unterschiedlicher Oberfläche.	Die Benutzeroberfläche der Home Assistant Plattform ist für Einsteiger und Anfänger deutlich benutzerfreundlicher und weniger komplex als openHAB.
Mobile Anwendungen	openHAB bietet eine dedizierte App für iOS und Android. Diese ist mit innovativen Optionen ausgestattet.	Home Assistant bietet ebenso eine App für iOS und Android, jedoch bei weitem nicht so stark entwickelt als die von openHAB. Die Benachrichtigungsdienste sind jedoch mit Home Assistant besser genutzt.
Community und Dokumentation	openHAB bietet eine sehr starke und repräsentative Community. Die Dokumentation ist ausführlich und beschreibt die Prozesse, Konzept und Architektur ausführlich.	Home Assistant unterscheidet sich in diesen Punkten nicht von openHAB. Eine starke Community und eine solide Dokumentation der Plattform.
Vergleich von Xtend & YAML	XII	

Tabelle B.1: Vergleich der Plattformen [BRICE 2022] [SUTTNER 2022] [BARCLAY 2020]

## Anhang C Beliebteste Smart-Home-Marken in Deutschland im Jahr 2022



**Quelle**  
Statista Global Consumer Survey (GCS)  
© Statista 2022

**Weitere Informationen:**  
Deutschland: vier Wellen zwischen 1. April 2021 und 17. März 2022; 1.384 Befragte; 18 bis 64 Jahre;  
Beleuchtung, Gebäudesicherheit oder Energiemanagement besitzen



## Erklärung

Ich versichere, dass ich diese Master-Thesis mit dem Thema: „*Konzeption und prototypische Umsetzung einer Steuerzentrale eines smarten Büros mit dem Fokus einer einfachen Handhabung der formalisierten Interaktionen für Softwareentwickler*“ selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlichen oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet habe. Die Arbeit wurde noch keiner Kommission zur Prüfung vorgelegt und verletzt in keiner Weise Rechte Dritter.

---

Ort, Datum

Unterschrift