

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

Teilnehmernummer

Kurs

ErstprüferIn

ZweitprüferIn

24 Wochen

800864

PSEJG20

Prof. Dr. Natividad Martinez Madrid

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

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 **SH!** 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.2022m]

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 Entwicklungsphasen¹ 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.2022m]
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.2022m]
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.2022m]
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.2022m]
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 Markteinsetzbarkeit und Relevanz der Technologie zahlen sich eindeutig aus. [gartner.2022m]

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

¹Die 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. **SH!** (**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. **IoT!**, 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.

Mit der erzielten Marktreife entstehen Produkte und Lösungen, die bestimmte Teile der anfänglichen Idee abdecken. Mit zunehmender Entwicklung und anfallenden Anforderungen, werden viele Produkte zu groß und haben dadurch die grundlegende Konzeption und Architektur nicht vorausschauend entwickelt. Daher ist die anfängliche Überlegung und Konzeption essentiell.

Daher ist ein weiterer Punkt meiner Motivation den Schritt zu gehen, ein Konzept zu entwickeln, dass die Erweiterung eines solchen Systems basierend auf der Konzeptgrundlage vereinfacht und so die Nutzung für Entwickler zur Weiterentwicklung verbessert. Somit soll dem Entwickler bei einer stetigen Erweiterung der Plattform Zeit und Aufwand erleichtert werden. Dadurch können weitere Anwendungsszenarien und Objekte integriert werden, ohne einen zu großen Entwicklungsaufwand zu erzeugen.

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 **SH!** zum **IdD!** und der damit einhergehenden Technologie bringt diese Vorteile mit sich, wie z.B. die Modernisierung von Wohngebäuden 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 den formalisierten Interaktionen eines Softwareentwicklers bei der Weiterentwicklung eines bestehenden Softwarekonzeptes. In der Phase der Anforderungserhebung werden Experten Interviews 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 **RE! (RE!)** und der Anwendung des *user-centered design*-Prinzips² ermittelt. Dabei wird die Nutzerorientierung auf den Softwareentwickler ausgelegt. Zur anschließenden Evaluation wird erneut ein Experten Interview sowie ein Usability Test durchgeführt, damit Eindrücke über das entstandene Konzept entstehen und somit die Forschungsfrage beantwortet werden kann. Die soeben genannten Forschungsmethoden werden im Nachfolgenden näher erklärt.

1.4.1 Experten Interview

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 [robson2002real]. 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.

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

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, anschließend die Durchführung und abschließend die Dokumentation und Offenlegung der Ergebnisse [Kitchenham2007].

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 (??) beschreibt die Motivation (??), welche die Intension kundtut, diese Thematik rund um **IoT!** und Smart Home zu bearbeiten. Darauf folgend werden die Forschungsfragen (??), die im Rahmen der Thesis behandelt werden, erläutert. Nach der Beschreibung der Forschungsfragen wird im anknüpfenden Abschnitt (??) 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 (??) widmet sich den essentiellen und wichtigen Grundlagen dieser Arbeit. Zu Anfang wird dem Leser der Terminus des **IoT!** (??) offenbart, um zum Teil den Kontext im Bezug zu dieser Arbeit zu begreifen, gefolgt von einer Einführung in die Thematik des **SH!** (??), 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 **SH!**, werden Technologien (??) aufgegriffen, die im Rahmen dieser Arbeit erwähnenswert sind und verwendet werden. Abschließend werden in Kapitel (??) die Softwarelösungen, Home Assistant und openHAB (?? & ??), dargestellt. Diese dienen zur Grundlage für die Evaluation als auch zur Gegenüberstellung der Lösungen in Kapitel (??) Diskussion und Evaluation.

Die theoretischen und methodischen Hintergründe sowie den Stand der Technik wird in Kapitel (??) 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 (??) befasst sich mit den Anforderungen, engl. Requirements, die für die eigentliche Konzeption relevant sind. Innerhalb dieses Kapitels wird anhand von Informationen und den umzu-

setzenden 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 (??) 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 (??) 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 Ergebnis wird aus objektiver Sicht in dem darauf folgenden Kapitel (??) erläutert.

Nach Abschluss der Umsetzung und dessen Ergebnisdokumentation befasst sich das Kapitel (??) 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 (??), 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 (??) 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 **IdD!** (**IdD!**), im Englischen **IoT!** (**IoT!**), zählt als eines der Schlagworte in der **IT!** (**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 **IdD!** lassen sich Anwendungen sowie Prozesse automatisieren und Aufgaben erledigen ohne das von außen Eingegriffen werden muss [**bigdatainsider2016**]. Die Prozessautomatisierung findet sich auch im Kontext des **SH!** wieder, welches in nachfolgendem Kapitel genauer aufgegriffen wird.

In der einschlägigen Literatur gibt es für das **IoT!** 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 Ashton¹, 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 [**Madakam2015**]. Eine aus dem Zusammenhang hervorgehende, aus dem wissenschaftlichen Artikel entnommene Definition ist folgende:

¹Britischer Technologie-Pionier, Mitgründer des Auto-ID Centers am Massachusetts Institute of Technology (MIT). 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” [Madakam2015]

Daraus kann die Ableitung erfolgen, dass der Begriff des **IdD!** 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 **IP!** (**IP!**)-Adresse, im Netzwerk belegt und mit elektronischer Intelligenz ausgestattet [bigdatainsider2016]. 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 **SH!** verwendet. Die Funktion als auch die Umsetzung wird im Kapitel (??) näher beleuchtet.

Das **IdD!** 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 **IoT!** 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.² Die Vereinigung beider Welten ist die Verknüpfung physischer Objekte, die eindeutig identifizierbar sind, mit einer virtuellen Repräsentation in einer vergleichbaren Internet-Struktur.

Gesamtbild des IoT!

Der folgenden Abbildung (??) ist zu entnehmen, welche Technologien rund um das **IdD!** liegen und in Verbindung damit stehen.

²Das 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.

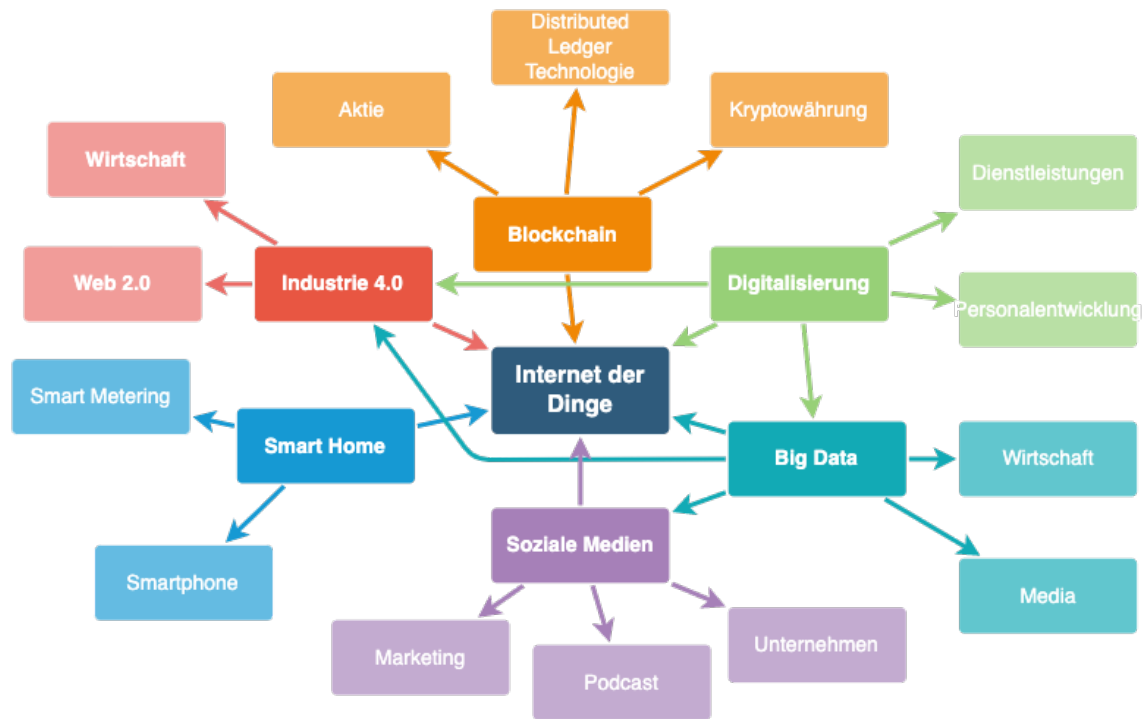


Abbildung 2.1: Technologische Einordnung von IoT [iotmindmap2018]

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 Data³ 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 (??).

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

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

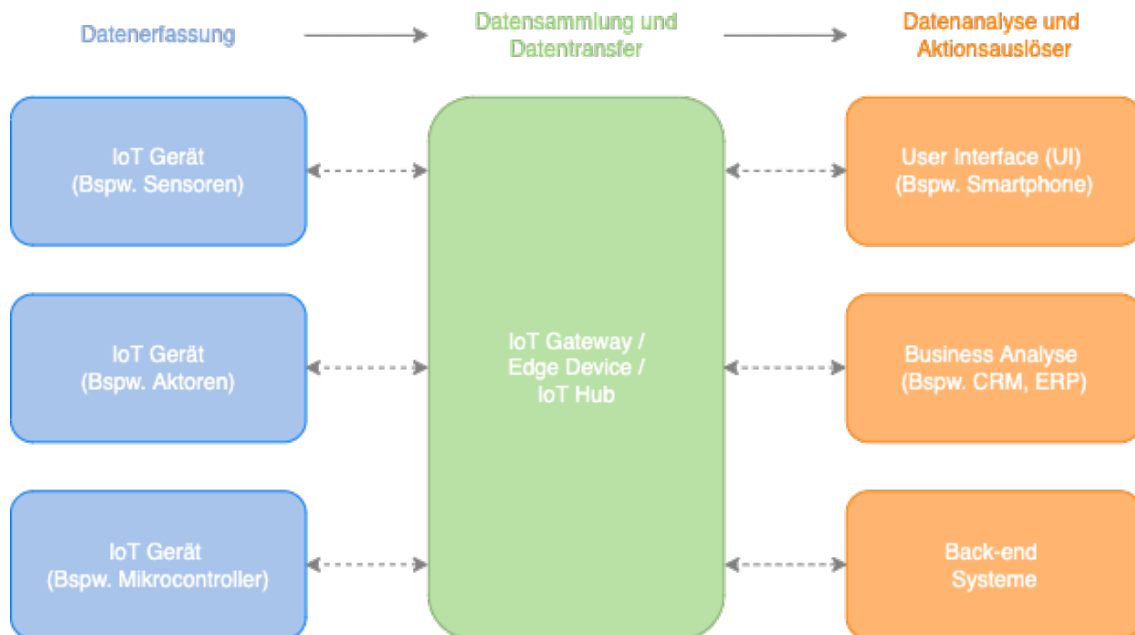


Abbildung 2.2: Exemplarische Darstellung eines **IoT**-Systems [iotskizze2022]

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 **IoT** 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 **SH**-Konzeptes, welches in Abschnitt (??) 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*⁴, auch bekannt als *IIoT!* (*IIoT!*). Im Rahmen dieser Arbeit wird zwischen den beiden Anwendungsbereichen differenziert und den Fokus auf den privaten Bereich gelegt.

Mit dem **IdD!**-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 [...]” [Cao2020]

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 angeboten⁵. 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.” [Mell2011]

Mit dem Cloud Computing wird im Vergleich zum Edge Computing ein zentraler Ansatz verfolgt, bei dem die erzeugten Daten von den Akteuren 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.

⁴Definition 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

⁵Einblick 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 (??) zu entnehmen.

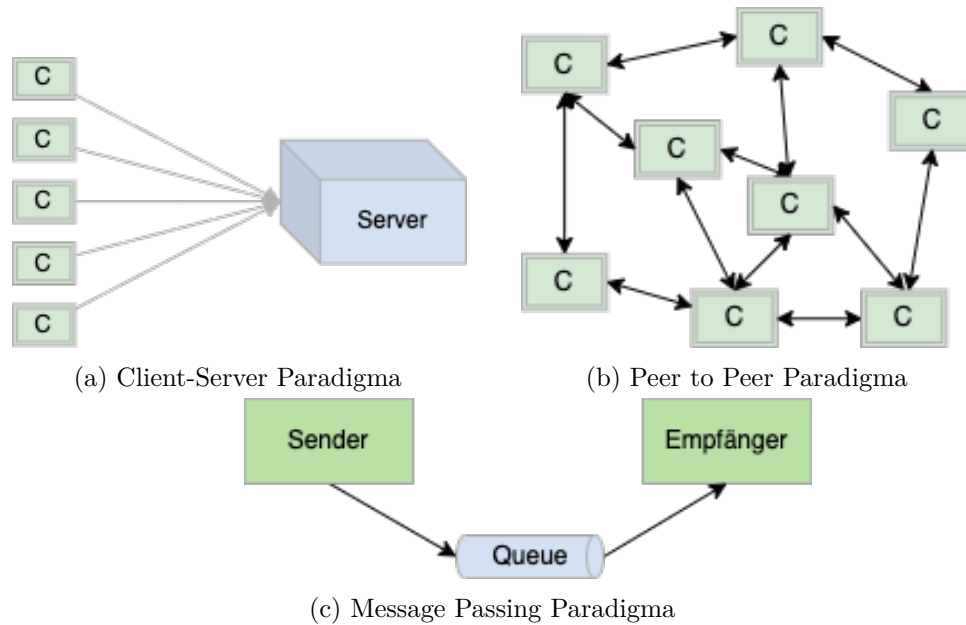


Abbildung 2.3: Client-Server, Peer-to-Peer und Message Passing Interaktionsparadigma [IEEE2015]

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

Tabelle 2.1: Interaktionsparadigmen nach [IEEE2015]

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 **MtoM!** (**MtoM!**) finden, kategorisiert:

- **SOAP! (SOAP!)**⁶: Dabei handelt es sich um ein Standardprotokoll, das dafür entwickelt wurde, um verschiedene Programmiersprachen auf verschiedenen Plattformen untereinander kommunizieren zu lassen.
- **REST! (REST!)**⁷: 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 **HTML! (HTML!)**, **XML! (XML!)** und **JSON! (JSON!)**.

Beide verfolgen das Ziel der Datenübermittlung zwischen Web-Anwendungen über eine Schnittstelle, das sogenannte **API! (API!)**. **REST!** Architekturen werden bei einem **IoT!**-Szenario aufgrund ihrer Simplität und ihres Komforts bevorzugt. [IEEE2015]

2.1.2 Historische Entwicklung

Die ersten Konzepte zu dem heute bekannten **IoT!** liegen schon einige Jahrzehnte zurück. Die historische Definition des **IoT!** 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:

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

⁷Detailliertere 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. ⁸
1995	Das Unternehmen Siemens leitete das erste Mobilfunkmodul ein, welches für die Kommunikation zwischen Maschinen zuständig ist (MtoM! -Technologie).
1999	Die Definition des Begriffs IoT! von Kevin Ashton wurde veröffentlicht und von der Gesamtheit akzeptiert, als er bei PG! (PG!) an den sogenannten RFID! (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. ITU! (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 **IdD!** [Durga2020]

Mittlerweile ist das **IdD!** 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 **FraunhoferIIS!** (**FraunhoferIIS!**)

2.1.3 Ziele von IoT!

Die Definition von dem **IdD!** 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 **SH!** 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

SH!, im Deutschen *“intelligentes Zuhause“*, ist ein wesentliches Anwendungsgebiet des **IoT!**. Diese Rubrik der Anwendung widmet sich überwiegend dem Gebrauch im privaten Umfeld und 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 **SH!** ist eine Weise zu verstehen, mit der die Erhöhung der Wohn- und Lebensqualität, Energienutzung unter Verwendung vernetzter und fernsteuerbarer Geräten 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.2010m]

Eine vergleichbare Definition wurde zu späterem Zeitpunkt durch eine Literaturrecherche publiziert. Diese beschreibt die zugrundeliegende Thematik weniger aus Anwendersicht sonder 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[...]“ [Balakrishnan2018]

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 Definition gibt. Aus der einschlägigen Literatur wird ersichtlich, dass viele Synonyme für die Benennung der Thematik verwendet werden, darunter beispielsweise [strese.2010m]:

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

- Home of the Future

Eine elementare Information im Zusammenhang zu dieser Arbeit ist, dass die Verwendung des Begriffs *intelligentes Büro* ebenso in den Kontext des **SH!** gehört. Hierbei wird lediglich die Räumlichkeit im unternehmerischen Jargon verwendet, die ebenso eine Grundlage für die Verwendung von Komponenten des **SH!** bietet.

An dieser Stelle 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 SH!

Der Zentrale Punkt des **SH!** 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	Haustierversorgung, Aquarienmanagement, etc.
Mobilität	PKW mit Diagnostik, Navigationssystem mit local based services, Info-/Entertainmentangebote etc.

Tabelle 2.3: Teilsysteme des Smart Home [strese.2010m]

Ein weiterer wichtiger Anhaltspunkt zum Verständnis der Definition von Smart Home ist die Ausstattung der Komponenten mit Intelligenz und die Vernetzung der Teilsysteme. Dadurch steht als Ziel im Vordergrund vielmehr die verteilte Intelligenz, um Aufgaben möglichst autonom (eigenständig) abzuarbeiten und weniger die übergeordnete zentrale Steuerung. Die dabei erzeugten als auch erforderlichen Daten mit anderen Komponenten des Gesamtsystems auszutauschen, ist ebenso ein vorangestelltes Ziel, welches eine intelligente Umgebung schafft.

Eine mögliche Vernetzung und auch Verwendung solcher Komponenten wird in folgender Abbildung

(??) skizziert. Diese Grafik dient als grobe Übersicht potentieller Anwendungsszenarien, repräsentiert jedoch nicht alle Möglichkeiten der Anwendung. Darüber hinaus gibt es weitaus mehrere

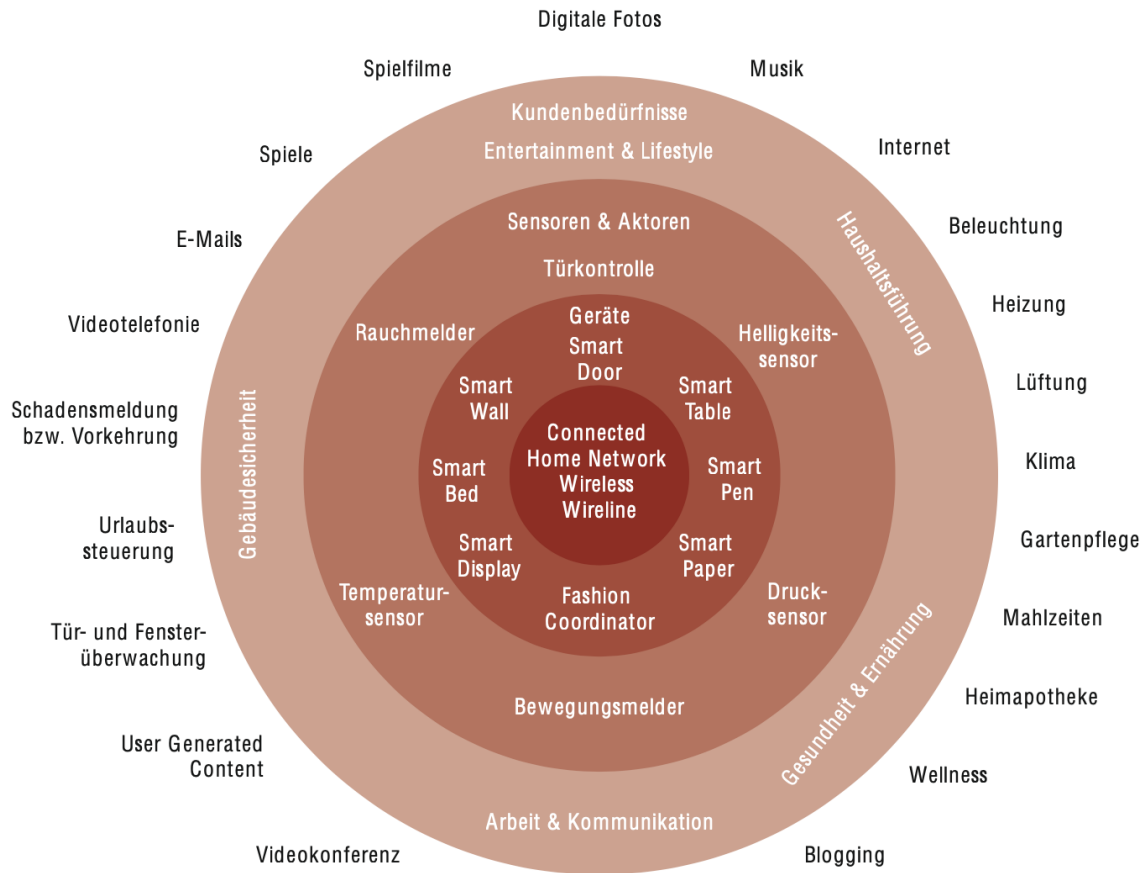


Abbildung 2.4: Mögliche Anwendungsszenarien im Smart Home [strese.2010m]

Anwendungsszenarien, beziehungsweise werden diese in der Abbildung (??) in einem Überbegriff zusammengefasst. Ein Anwendungsszenario, welches immer mehr Zuwendung findet, ist die Koppelung von Robotern jeglicher Art, darunter Staubsaugerroboter, wobei diese schon weiter verbreitet sind, oder vor allem jedoch Service-Roboter, die immer mehr in die Thematik des **SH!** versucht zu integriert werden.

Einordnung von Smart Home in das Internet der Dinge

Im Konsumentenmarkt wird die Technologie des **IoT!** in Produkten eingesetzt, die das Konzept des **SH!** 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. und Übertragungstechnologien, die im Abschnitt (??) aufgegriffen werden.

Weitere Beispiele sind der Tabelle (??) zu entnehmen.

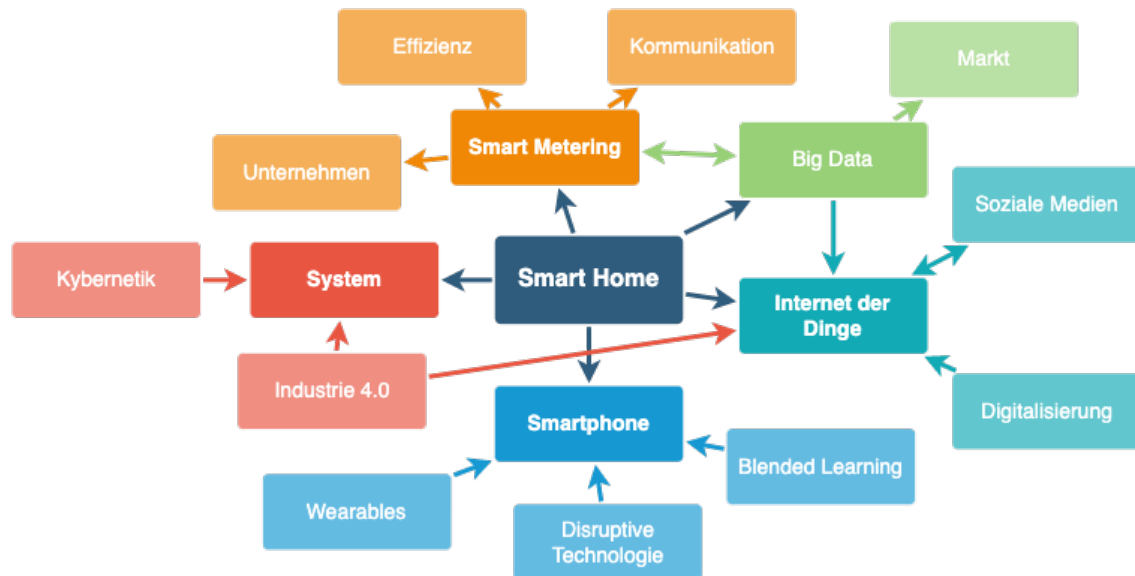


Abbildung 2.5: Technologische Einordnung von Smart Home in Verbindung zu IoT [shmindmap2021]

Die Abbildung (??) zeigt die Verbindungen als auch die Beziehungen von **SH!** zu anderen Technologien. Hier wird deutlich, dass im Bereich des **SH!** 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 Metering⁹ 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 Anbieter zur Rechnungsstellung genutzt werden. Ein Beispiel dazu ist ein digitaler intelligenter Stromzähler mit direkter Kommunikationsmöglichkeit zum Anbieter selbst.

Der Aufbau eines **SH!** ist architektonisch ähnlich zu dem Grundprinzip einer **IoT!**-Lösung. Die Veranschaulichung des zugrundeliegenden Aufbaus ist der Abbildung (??) zu entnehmen.

Um die Bezugspunkte zu **IoT!** und die Einordnung zu untermauern, wird in folgendem Abschnitt auf die Funktionsweise von **SH!** eingegangen.

Funktionsweise eines SH!

Ein **SH!** 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.

⁹Smart 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

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 möglich. 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 dafür zuständig, die Verbindung aller Elemente. Hierfür kommen verschiedene Protokolle, die unter anderem in Abschnitt (??) beschrieben werden, per Funk, Kabel oder Stromnetz zum Einsatz. Ein exemplarischer Aufbau der Komponenten ist der folgenden Abbildung zu entnehmen:

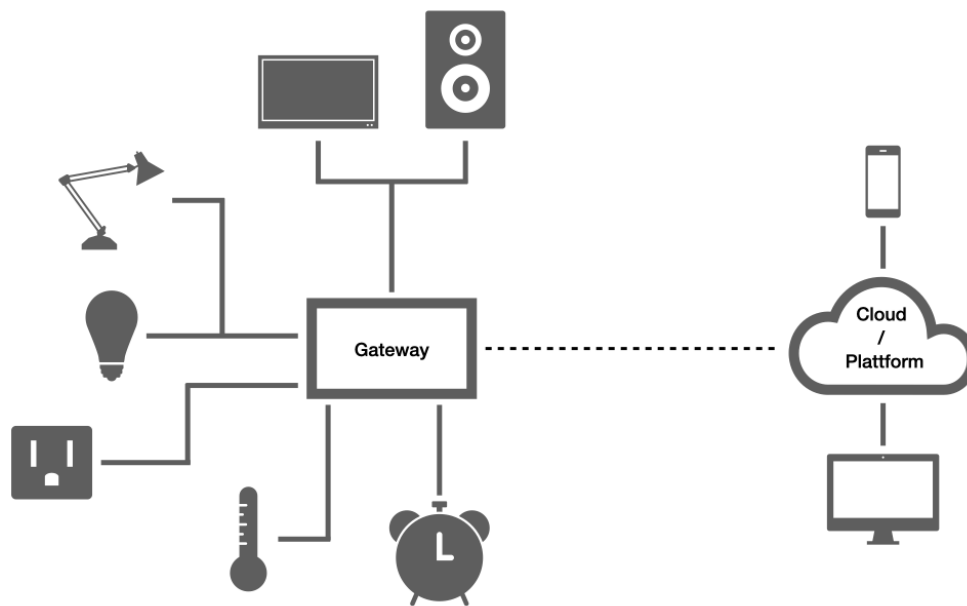


Abbildung 2.6: Aufbau und Funktionsweise einer Smart Home Infrastruktur

2.2.1 Historische Entwicklung

Im Bereich des **SH!** wurde im Jahr 1975 die erste Netzwerktechnologie für Hausautomationen präsentiert und vermarktet. Bekannt wurde diese unter dem Namen *xx0*¹⁰. Dabei handelt es sich um ein stromleistungsbasiertes 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 [aschendorf2014energiemanagement]. 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 basierte ebenso auf dem grundlegenden Netzwerkprotokoll *X10* und fügte sich nahtlos in den europäischen Installationsbus (*EIB/KNX*) ein [busch-jaeger]. *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-forschungsanlage]. Innerhalb dieser Institution erforschen, entwickeln, testen und demonstrieren 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 **SH!** Bereich aktiv und präsentierte der Öffentlichkeit ein vollständig vernetztes „intelligentes“ Musterhaus in Berlin, das sogenannte *T-Com-Haus*. Anfang des Jahres 2012 wurde das **BMWK!** (**BMWK!**) im Sektor des **SH!** aktiv. Seitdem fördert die Behörde das „Zertifizierungsprogramm Smart Home + Building“, bei dem Forschungen im Bereich des **SH!** von Vertretern akademischer Einrichtungen und Industrieunternehmen, die in diesem Segment unterwegs sind, durchgeführt werden. Ziele dabei sind unter anderem die Erstellung von Standards und Prüfsiegeln für systemübergreifende Interoperabilität der Geräte eines **SH!** [vde-smartAndBuilding]. 2013 bot die deutsche Telekom neue Lösungen in einem weiteren Musterhaus in Darmstadt. Darin sind mehrere Komponenten und Vernetzungen geboten als in dem Vorgänger in Berlin [telekom_SH]. Ein Augenmerk dabei liegt auf der Nutzung verschiedener Funkstandards, die es ermöglichen, intelligente Geräte unterschiedlichster Hersteller zu konfigurieren und über ein Smartphone, Tablet oder Computer zu kontrollieren und steuern [telekom_SH]. Die Auswirkungen der rasanten Entwicklung dieser Technologie treiben das Interesse in die Höhe, sodass 2014 und 2015 die Technologie das Hauptthema der **IFA!** (**IFA!**) bei vielen Ausstellern war. Bis heute wird in dem Segment des **SH!** 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 (??) nochmals aufgegriffen und mit Statistiken und Umfragen belegt.

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

2.2.2 Ziele von Smart Home

Die Ziele der intelligenten Vernetzung sind in erster Linie die offensichtlichsten, die auch jeweils große Domänen abdecken. Die Ziele sind **[goals-smarthome]**:

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

Allgemein lässt sich der Komfort durch die bequeme Steuerung von Licht, Heizung, Unterhaltungselektronik, Service-Robotern und vielen weiteren Geräten aus der Ferne als auch aus unmittelbarer Nähe steigern. Hierfür spielen beispielsweise Smartphones, zentrale Steuerungsgeräte und Tablets eine wichtige Rolle. Diese fungieren in dem Bezug ähnlich zu einer herkömmlichen Fernbedienung. Es können ebenso Prozesse automatisiert werden 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 höher gestellt werden, damit bei Eintreffen in das Gebäude eine optimale Temperatur herrscht. Die Steuerung von Geräten spielt ebenso eine Rolle in den Bereichen der Energie- und Kosteneffizienz und der Sicherheit.

Um Energiekosten zu verringern, sind zum Beispiel mehrere Komponenten durch eine Automation miteinander verbunden. Dadurch ist es möglich anhand der Informationen, bspw. des Temperaturfalls bei offenem Fenster anhand von Sensoren zu erkennen, ob ein Fenster offen ist. Dementsprechend kann die Heizung für den Zeitraum ausgeschaltet oder reguliert werden, damit die Heizung nicht überflüssig Wärme erzeugt. Ein weiteres Sparpotential ist das ausschalten von aktuell nicht benötigten Ressourcentypen, wie z.B. Elektrogeräte oder LED-Lampen, die unter anderem vor Verlassen des Gebäudes vergessen wurden auszuschalten.

Das letzte größere Ziel, welches mit **SH!** verfolgt wird, ist die Erhöhung der Sicherheit und das Überwachen von Eingängen und Fenstern. So können durch Bewegungsmelder und Sensoren Aktivitäten registriert werden, die bei Abwesenheit den Eigentümer informieren. Schutz vor Einbruch, bzw. das direkte Handeln auf bestimmte Ereignisse. Mit Kameras können auch Dinge überwacht werden, beispielsweise, wenn ein Haustier kurze Zeit alleine zuhause ist oder Einbruchversuche stattfinden. Ebenso kann eine Alarmierung über Sensoren erfolgen, wenn Wasser im Keller steht oder ein Kurzschluss eines Gerätes ein Feuer auslöst.

Die Geräte im **SH!** erfüllen weitestgehend diese Ziele und werden stetig weiterentwickelt, verbessert und neue Funktionalitäten entwickelt.

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 **SH!** mehrere Methoden und Möglichkeiten, die Objekte miteinander zu vernetzen. Unter dessen gehören Protokolle über Bluetooth, Ethernet, WLAN, Bussysteme, Funk und Stromleitung. Diese werden Abhängig von den Herstellern eingesetzt. Proprietäre Systeme funktionieren nur über eine Übertragungsmethode. So erzwingen die Hersteller die Nutzung einer Produktlinie, bzw. den Kauf einer einheitlichen Lösung. 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 **SH!** kommunizieren können. Die populärsten werden in folgender Tabelle aufgelistet¹¹:

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

Die Auflistung der zum aktuellen Zeitpunkt am meist verwendeten Übertragungsmethoden dient lediglich als Einblick, damit eine große Gesamtübersicht der Technologien und der Thematik **SH!** entsteht. Demnach wird im Rahmen dieser Arbeit das Thema oberflächlich erläutert und nicht ausführlich vertieft.

¹¹Auswahl von derzeit verwendeten Übertragungsmethoden. https://de.wikipedia.org/wiki/Smart_Home#Übertragungsmethoden Abgerufen am 02.04.2022

2.3.2 MQTT

Das **MQTT!** (**MQTT!**)-Protokoll ist eines der ältesten offenen Netzwerk- und Nachrichtenprotokolle der **MtoM!**-Kommunikation. Dies wurde 1999 von IBM Mitarbeiter Andy Stanford-Clark¹² und von Cirrus Link Solutions Mitarbeiter Arlen Nipper¹³ 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 [Naik2017]. **MQTT!** ist ein klassisches Client-Sever-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 [Naik2017]. **MQTT!**-Clients können mehrere Topics abonnieren und erhalten jede Nachricht, die an das jeweilig abonnierte Topic gesendet wird.

Die Leichtgewichtigkeit 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 [Naik2017] enthält. Grundlegend ist **MQTT!** auf der Basis des Transportprotokoll **TCP!** (**TCP!**) aufgebaut und nutzt zur Verstärkung der Sicherheit die **TLS!** (**TLS!**)/**SSL!** (**SSL!**)-Verschlüsselung. Dadurch sind Client und Broker mit ihrer Kommunikation verbindungsorientiert. Die Anwendung von **MQTT!** im Bereich des **SH!** 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 simplen und leichtgewichtig [Naik2017].

Ein interessanter Aspekt des **MQTT!**-Brokers ist, dass er die gesamte Datenlage seiner Kommunikationspartner aufbewahrt und so die Option bietet, zusätzlich als Zustand-Datenbank betrieben werden kann. Dadurch können weniger leistungsfähige Geräte mit einem **MQTT!**-Broker verbunden werden, bei denen die Geräte Befehle und Daten entgegennehmen können, zugleich aber ein komplexeres Lagebild auf dem Broker entsteht. So können Daten an einen leistungsfähigeren Kommunikationspartner weitergeleitet und dort ausgewertet werden.¹⁴

Mit diesem Aspekt können durch das **MQTT!** Protokoll Automatisierungslösungen geschaffen werden und findet dadurch im Segment **IoT!** und **SH!** einfache Verwendung und dementsprechend eine große und schnelle Verbreitung.

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

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

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

Zusammengefasst ist der **MQTT**!-Broker die Kommunikationsschnittstelle der **SH**!-Geräte und der **SH**!-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 daran interessiert sind, bestimmte Informationen zu konsumieren, ihr Interesse anmelden [Hunkeler2008]. 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 an bestimmte Abonnenten weitergeben. Diese Vermittlung der Informationen zwischen Herausgeber (publisher) und Abonnent (subscriber) erfolgt über den Markler (broker), dieser koordiniert sämtlichen Abonnements. Alle Abonnent müssen sich explicit bei dem Broker anmelden, um die Informationen zu erhalten [Hunkeler2008].

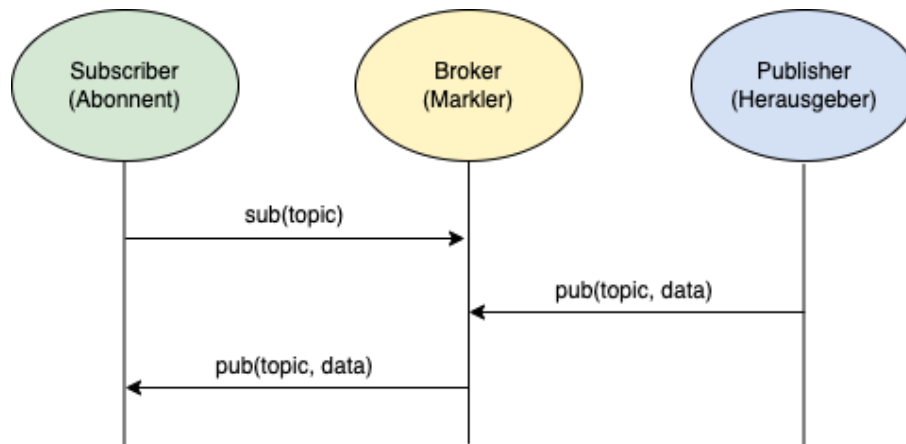


Abbildung 2.7: Themenbasiertes Pub/Sub Kommunikationsmodell [Hunkeler2008]

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

Beispiel

IST NOCH AUSZUARBEITEN. BEISPIELBESCHREIBUNG FOLGT!!!!

```
1 subscribe -topic: test/test/
```

Code-Beispiel 2.1: Test test

2.3.3 AMQP

Das **AMQP!** (**AMQP!**)-Protokoll ist ebenso ein leichtgewichtiges **MtoM!**-Protokoll, welches im Jahre 2003 von John O'Hara JPMorgan Chase in London, Großbritannien, entwickelt wurde [Naik2017]. Der Fokus dieses Protokolls liegt auf der Unternehmens-Messaging Ebene und legt hohen Wert auf die Zuverlässigkeit, Sicherheit, Bereitstellung und Interoperabilität der Kommunikation. **AMQP!** unterstützt neben der Publish/Subscribe- auch die Request/Response Architektur. Es bietet eine breite Palette von Funktionen im Zusammenhang mit Messaging, wie z. B. zuverlässiges Queuing, themenbasiertes Publish-and-Subscribe-Messaging, flexibles Routing und Transaktionen [Naik2017]. 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 [Naik2017]. Die beiden Komponenten, Empfänger und Herausgeber, nutzen den Namen des Austausch, um eine Verbindung aufzubauen. 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 [Naik2017].

Das binäre Protokoll **AMQP!** erfordert einen Header von acht Byte mit Nachrichtennutzlasten, die die Größe der Nachricht ist abhängig von dem Broker, bzw. dem Server. Die Verbindungsorientierte Kommunikation von **AMQP!** basiert auf dem Standard-Transportprotokoll **TCP!** und zur Sicherheit auf dem **TLS!**/**SSL!**-Protokoll. Ein Kernmerkmal des **AMQP!** Kommunikationsmodells ist die Zuverlässigkeit [Naik2017].

Neben den beiden aufgeführten Kommunikationsmodellen gibt es noch weitere, darunter das klassische **HTTP!** (**HTTP!**) und das **COAP!** (**COAP!**), die allerdings nicht weiter ausgeführt werden. Der Fokus in dieser Arbeit liegt auf dem **MQTT!**-Protokoll.

2.4 Home Assistant

Eines der populärsten **SH!** Plattformen ist das sogenannte Home Assistant System. Die Open-Source-Software ist ein zentrales Steuerungssystem von Heimautomationen und der Verwaltung von intelligenten Geräten mit dem Fokus der lokalen Steuerung und gesicherter Privatsphäre. 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 die Steuerung per Sprachbefehlen ermöglichen. Kompatible Plattformen sind unter anderem Google Assistant, Amazon Alexa und Apple HomeKit. Dies sind weitaus nur eine Selektion von bekannten Herstellern. 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 **PAN!** (**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.¹⁵

Die Plattform ist in Python geschrieben und 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 im November 2013 erstmals auf GitHub veröffentlicht.¹⁶

Vier Jahre nach den ersten Entwicklungen der **SH!** Plattform wurde im Juli 2017 ein verwaltetes Betriebssystem mit dem Namen *Hass.io* entwickelt.¹⁷ Dadurch gelang der Durchbruch der vereinfachten Verwendung von der Home Assistant Plattform auf kleineren Computern, sogenannten Einplatinencomputern, wie beispielsweise einem 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 ermöglicht. Ein weiteres Feature des Supervisors ist die Möglichkeit der Plattform über Add Ons weitere Funktionalitäten zu Verfügung zu stellen.¹⁸

Die Software wird stetig weiterentwickelt und verbessert. Mittlerweile gehört sie zu den am meist genutzten Open-Source-Plattformen im Bereich **SH!**.

¹⁵Grundlegende Ableitung der Definition von Home Assistant siehe https://en.wikipedia.org/wiki/Home_Assistant Abgerufen am 16.04.2022

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

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

¹⁸Einstieg 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.¹⁹

Damit der offene Ansatz von Home Assistant Anbietern gegenüber nicht eingeschränkt ist, bietet die Software Möglichkeiten, um viele Geräte zu vereinheitlichen. Somit begegnet Home Assistant der Heterogenität des offenen Marktes, in sofern, dass diese Geräte auf gemeinsame Konzepte gebracht werden. Diese sind in vier Konzepte²⁰ 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

¹⁹Konzept und Architektur von Home Assistant. https://developers.home-assistant.io/docs/architecture_index Abgerufen am 19.04.2022

²⁰Erläuterung 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 definiert²¹:

- 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

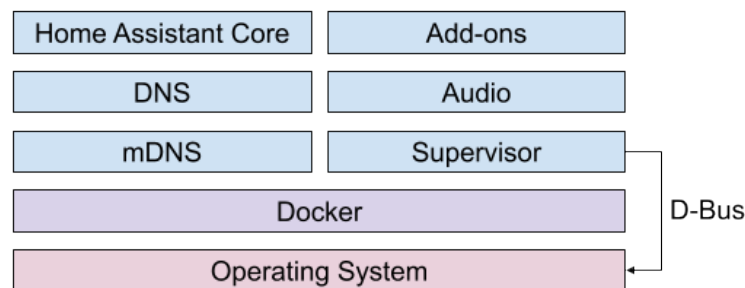


Abbildung 2.8: Architektur des Home Assistant Supervisors [**haos-supervisor**]

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 Timer²².

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 **UI!** (**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 SQLite²³ Datenbank verwendet.

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

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

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

Diese ist nur auf dem 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 [HAOSarchitecture2018].

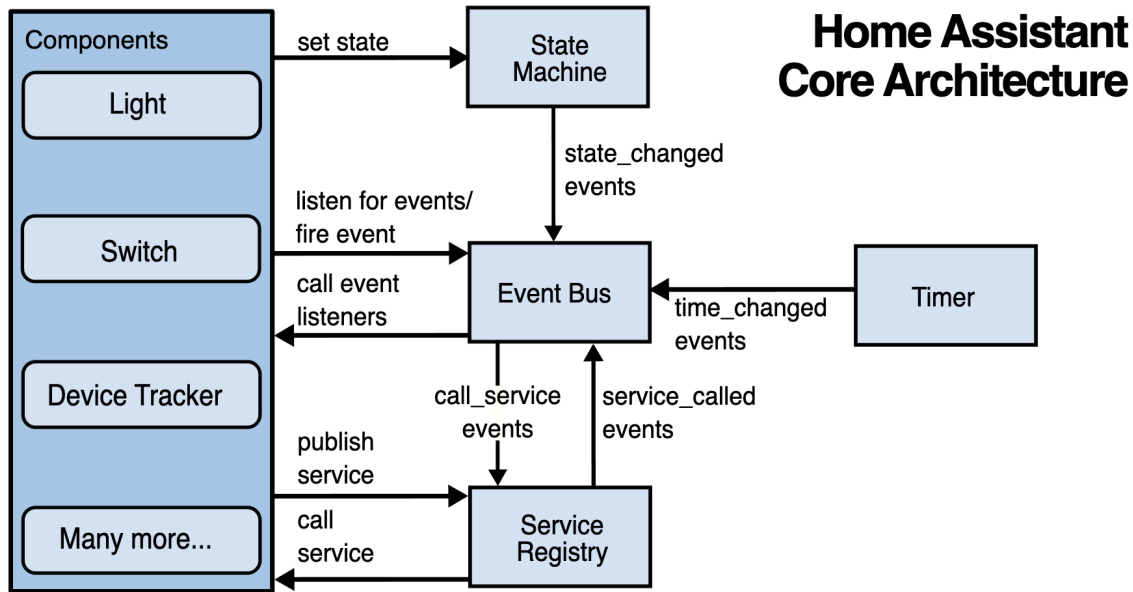


Abbildung 2.9: Architektur des Home Assistant Cores [HAOSarchitecture2018]

2.4.2 Ziele und Schwerpunkte

Jede Softwarelösung verfolgt bestimmte Ziele und Schwerpunkte, damit die Zielerreichung (**DOD!** (**DOD!**)) messbar ist. Die mit der Home Assistant Plattform verfolgten Ziele sind Privatsphäre und Sicherheit, die Wahlmöglichkeiten von Geräten und die Haltbarkeit der Plattform²⁴.

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

²⁴Schwerpunkte 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.

2.5 openHAB

Neben der so eben erläuterten Home Assistant Plattform zählt ebenso die openHAB Plattform als bekannt und in der Anwendung populär. Der **OPENHAB!** (**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 **OSGI!** (**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.²⁵

Historie

Das Smart Home Projekt begann im November 2009 und wurde von Kai Kreuzer entwickelt²⁶. 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.[**kaikreutzer2012**]“

²⁵Einleitung zu openHAB. <https://www.openhab.org/docs/> Abgerufen am 25.04.2022

²⁶Chronologie und Blogbeiträge von Kai Kreutzer. <http://kaikreutzer.blogspot.com> Abgerufen am 28.04.2022.

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 [kaikreutzer2014]. Seitdem kamen weitere Unterstützungen zu anderen Geräten und Kommunikationsprotokollen dazu. Im Laufe der Entwicklung wurde eine mobile Anwendung entwickelt, mit 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 [openHAB-article]. 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 [openHAB-article].
- 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 allein-stehender 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 [openHAB-article]. 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 Konzepte²⁷. 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.

²⁷Historie und Konzepte von openHAB. <https://medium.com/smartsmarthome/openhab-41c30d50fc42> Abgerufen am 28.04.2022

- 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 drunter liegende OSGi-Framework (grün) und die Erweiterungen (Add-Ons) (gelb).

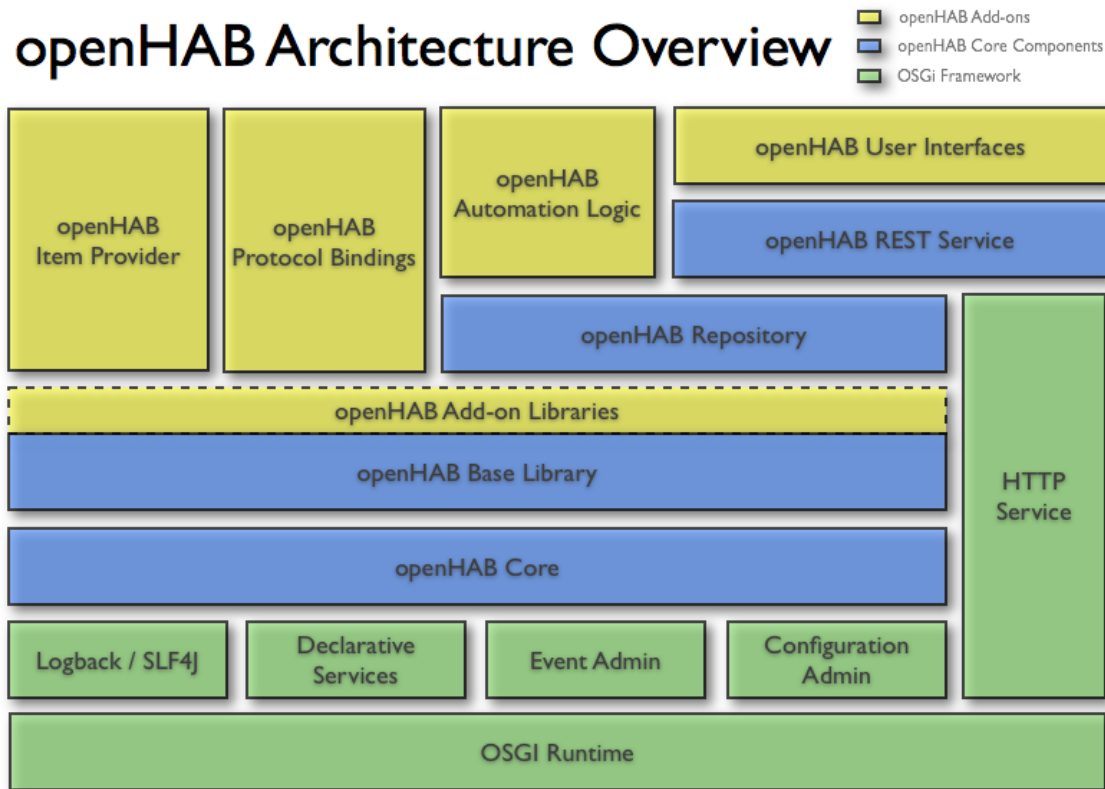


Abbildung 2.10: Architektur der openHAB Plattform [openHAB-architecture2013]

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*²⁸ 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

²⁸Eclipse SmartHome. <https://projects.eclipse.org/projects/iot.smarthome> Abgerufen am 29.04.2022

openHAB Plattform. Das Framework ist mittlerweile archiviert und erhält keine weiteren Aktualisierungen.

*Apache Karaf*²⁹ 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*³⁰ ist eine modulare Java basierte Laufzeitumgebung und eine Implementierung 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 [**openHAB-article**].

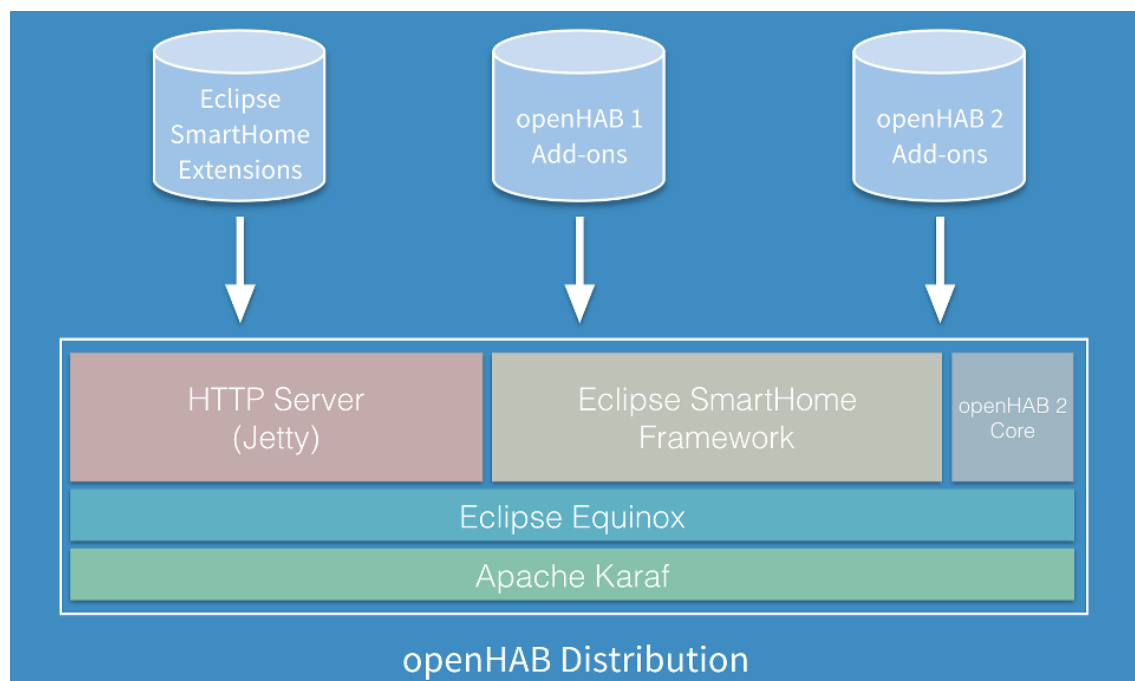


Abbildung 2.11: Architektur der openHAB 2.0 Plattform [kaikreutzer2016]

OSGi fördert den modularen Aufbau von Software, die mit der Programmiersprache Java geschrieben ist. Der Open-Source Standard bietet einen Ansatz, der eine modulare Plattform beschreibt und eine **JVM!** (**JVM!**) voraussetzt. Die **JVM!**³¹ 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

²⁹ Apache Karaf. <https://karaf.apache.org/> Abgerufen am 29.04.2022

³⁰ Eclipse Equinox. <https://www.eclipse.org/equinox/> Abgerufen am 29.04.2022

³¹ Detaillierte Beschreibung der JVM. <https://www.javatpoint.com/jvm-java-virtual-machine> Abgerufen am 01.05.2022

technischer Sicht ist dieses Konstrukt eine **JAR! (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 [openHAB-article].

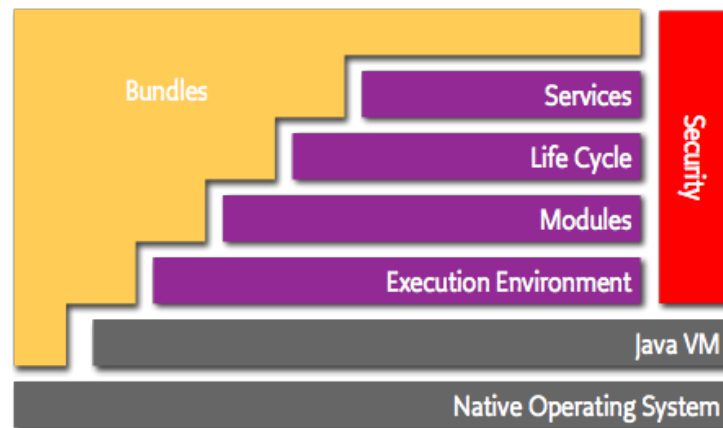


Abbildung 2.12: OSGi Schichtenarchitektur [openhab-osgi]

Weitere Details der **OSGI!**-Architektur³² sind der openHAB Dokumentation zu entnehmen.

2.5.2 Ziele und Schwerpunkte

„openHAB empowering the smart home³³“

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!³⁴“

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 [openhab-strength].

³² Architektur und Erläuterung OSGi. <https://www.openhab.org/docs/developer/osgi/osgi.html> Abgerufen am 01.05.2022

³³ Log und Label von openHAB <https://www.openhab.org/> Abgerufen am 01.05.2022

³⁴ Einleitung und Übersicht der openHAB Dokumentation. <https://www.openhab.org/docs/> Abgerufen am 01.05.2022

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 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 [openhab-strength].

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.

2.6 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 Erfahrungen und Eindrücke von Nutzern und Experten. Die Tabelle (??) zum Vergleich der Plattformen ist dem Anhang ... 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.

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 [sh-uni-comparison].

Tabelle 2.6: Vergleich der Plattformen [sh-uni-comparison] [msuttner-comparison] [barclay-comparison]

2.7 Requirements Engineering

In der heutigen Softwareentwicklung ist zu Anfang jedes Projekts die Frage offen, welche Anforderungen soll das Produkt erfüllen, sodass daraus eine gute, stabil und effiziente Software entsteht und der Umfang und das Ziel klar gestaltet sind. Mit dem **RE!** wird der Gedanke verfolgt, die Bedürfnisse des Kunden, bzw. des Stakeholders zu erfüllen und eine erfolgreiche, den Kunden zufriedenstellende Entwicklung des Systems zu erzielen. Ein Stakeholder ist eine Person oder Organisation mit Einfluss auf die Anforderungen des Systems oder die Auswirkungen auf das System hat [pohl2021basiswissen]. Diese Anforderungen werden durch das **RE!** ermittelt und dokumentiert.

Unter dem Begriff **RE!** ist folgendes zu verstehen:

Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit dem Ziel, die Wünsche und Bedürfnisse der Stakeholder zu verstehen und die Gefahr zu minimieren, ein System auszuliefern, das diese Wünsche und Bedürfnisse nicht erfüllt. [pohl2021basiswissen]

Oft wird darunter auch ein kooperativer und inkrementeller Prozess verstanden, dessen Ziele die Gewährleistung folgender Punkte darstellt:

- Alle Anforderungen sind bekannt und werden auch in dem erforderlichen Detaillierungsgrad verstanden.
- Alle involvierten Stakeholder haben eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielt.
- Alle Anforderungen sind zu den Dokumentationsvorschriften konform, bzw. zu den Spezifikationsvorschriften konform spezifiziert.

Damit ist von System-Design, Architekturen oder die nachfolgenden Tests zu differenzieren. Dieser Prozessschritt deckt die Formalitäten ab, um allen beteiligten eine grundlegende Übersicht zu geben, welche Ziele verfolgt werden, bzw. welche Anforderungen erfüllt werden sollen. Auch ist das **RE!** ein wesentlicher Bestandteil des Entwicklungsprozesses nach Scrum. Unter dem Begriff Scrum ist ein Vorgehensmodell zu verstehen, welches zum Projekt- und Produktmanagement in agilen Softwareentwicklungen eingesetzt wird.

Innerhalb der Anforderungen, die im **RE!** ermittelt werden, wird zwischen drei Arten von Anforderungen unterschieden:

- Funktionale Anforderungen: Diese legen die Funktionalität fest, die durch die Entwicklung des Systems erreicht werden soll. Typischerweise werden diese in Funktions-, Verhaltens- und Strukturanforderungen unterteilt.
- Qualitätsanforderungen, sogenannten **NFA! (NFA!)**: Diese legen die Qualitäten des Systems fest und beeinflussen die Gestaltung der Systemarchitektur in größerem Maß als die funktionalen Anforderungen.

- Rand- oder Rahmenbedingungen (Constraints): Diese legen die Randbedingungen des Systems fest. (Beispielsweise die zu nutzenden Frameworks, Programmiersprache, Konventionen, etc.)

NFA!s werden über das Produktqualitätsmodell, im Englischen **PQM!** (**PQM!**), nach der Norm ISO/IEC 25010:2011 abgebildet. [**ISO-IEC-PQM2011**]

Ein wichtiger Bestandteil des **RE!** ist die Spezifikation der Anforderungen, welche in einem Dokument, der sogenannten Anforderungsspezifikation, im Englischen **SRS!** (**SRS!**), beschrieben sind. Dieses Dokument unterliegt der Norm ISO/IEC 29148-2011 und deckt die Qualitätskriterien ab, damit die Anforderungen einem gewissen Standard entsprechen.

Das **RE!** spielt für die Anforderungsanalyse in Kapitel (??) und die darauf folgende Konzeption in Kapitel (??) eine wichtige Rolle. Alle Anforderungen und Spezifikationen wurden mit Zuhilfenahme der Standrads im **RE!** ermittelt. In den jeweiligen Kapiteln wird drauf genauestens eingegangen.

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 [Kitchenham2007] 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 (??) die Stichpunkte aufgegriffen und als Suchterm formuliert. Die daraus resultierenden Suchterme, die der tabellari-schen Darstellung (??) 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.

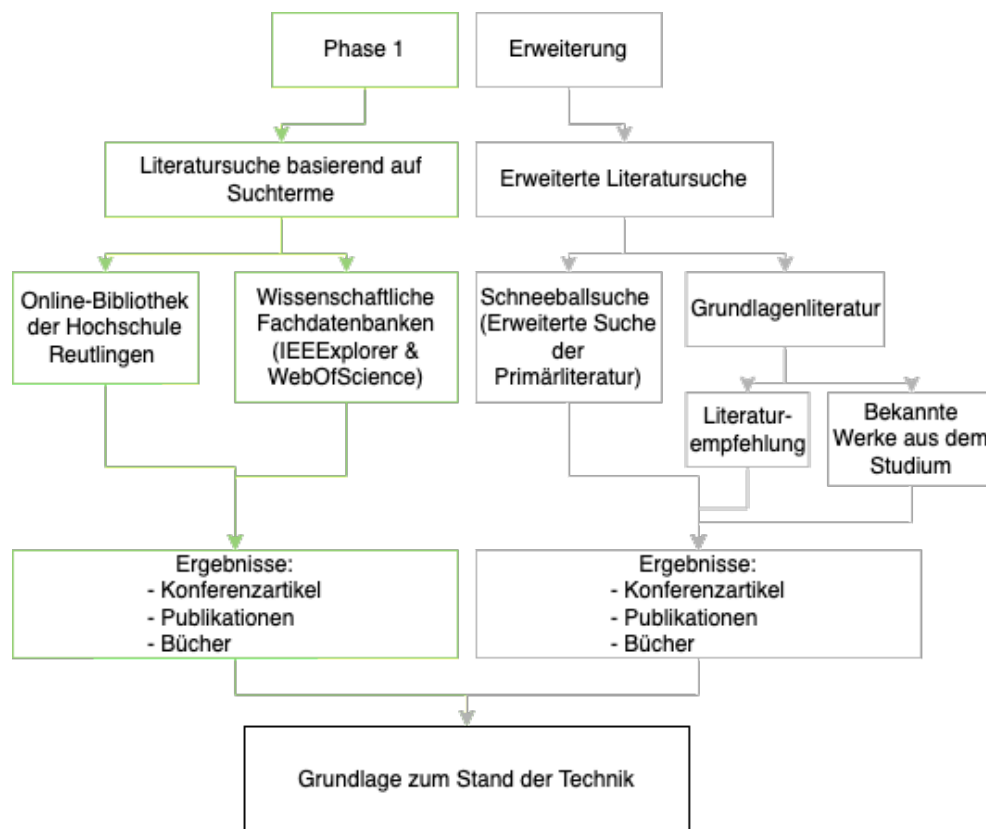


Abbildung 3.1: Strategie der Literatursuche

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

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: Mit den Forschungsfragen, den Suchtermen und den Einschluss- und Ausschlusskriterien zu der Quellenrecherche werden die dabei erzielten Ergebnisse studiert und anhand ihres Inhaltes synthetisiert.

3.1.4 Ergebnisse

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 gateway OR iot)	09.05.2022	Nein	IEEEExplorer, WebOfScience	13, 22	3, 3	0, 0

43
Tabelle 3.1: Suchprotokoll des Systematischen Literaturreviews

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 2000 veröffentlicht |
| 3 | Die Veröffentlichung ist in deutscher oder englischer Sprache |

Ausschluss-Kriterien

- | | |
|---|---|
| 1 | Die Veröffentlichung beinhaltet, bzw. behandelt nicht die Schlagworte usability oder architecture oder formalized interaction |
| 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

Kapitel 4

Anforderungsanalyse

4.1 Marktanalyse

4.2 Use Cases

4.2.1 Check in mit Temi

4.2.2 Notfallevakuierung mit Temi

4.3 Experten Interview

4.3.1 Ziele des Experten Interviews

4.4 Anforderungen

Kapitel 5

Konzept

5.1 Abzudeckende Funktionen

5.2 Architektur

5.2.1 Schnittstellen

5.2.2 Interfaces

Kapitel 6

Umsetzung

6.1 Implementierung

6.1.1 Aufbau der Architektur

6.1.2 Einbindung der Funktionen abgeleitet von der Konzeption

Kapitel 7

Ergebnis

Kapitel 8

Diskussion und Evaluation

8.1 Analyse des Konzepts der Eigenentwicklung

8.2 Vergleich zwischen Eigenentwicklung und bestehenden Softwarelösungen

Kapitel 9

Fazit

Kapitel 10

Ausblick

Abbildungsverzeichnis

Tabellenverzeichnis

Liste der Code-Beispiele

Abkürzungsverzeichnis

IoT	Internet of Things
IdD	Internet der Dinge
SH	Smart Home
IT	Informationstechnologie
IP	Internetprotokoll
IIoT	Industrial Internet of Things, dt. Industrielles Internet der Dinge
P2P	Peer to Peer
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
M2M	Machine to Machine
API	Application Programming Interface
HTML	Hypertext Markup Language
XML	Extensible Markup Language
JSON	JavaScript Object Notation
P&G	Procter & Gamble
RFID	Radio-Frequency Identifitaction
ITU	International Telecommunication Union
IIS	Fraunhofer Institut für Integrierte Schaltungen
RE	Requirements Engineering
SRS	Software Requirements Specification
NFA	Nichtfunktionale Anforderung
PQM	Product Quality Model
BMWK	Bundesministerium für Wirtschaft und Klimaschutz
IFA	Internationale Funkausstellung
MQTT	Message Queue Telemetry Transport

TCP	Transmission Control Protocol
TLS	Transport Layer Security
SSL	Secure Sockets Layer
AMQP	Advanced Message Queue Protocol
HTTP	Hypertext Transfer Protocol
CoAP	Constrained Application Protocol
PAN	Personal Area Network
UI	User Interface
DoD	Definition of Done
openHAB	open Home Automation Bus
OSGi	Open Service Gateway initiative
JVM	Java Virtual Machine
JAR	Java Archive

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