

# Smart-Home-Lösungen mittels Sprachsteuerung und Raspberry Pi

## STUDIENARBEIT

für die Prüfung zum

Bachelor of Science

des Studienganges Angewandte Informatik

an der

Dualen Hochschule Baden-Württemberg Karlsruhe

Abgabedatum 14.05.2017

Bearbeitungszeitraum

Theoriesemester 5 und 6

Autoren

Maximilian Hirte, Robin Warth

Matrikelnummern

8994521, 6028632

Kurs

TINF15B4

Ausbildungsfirma

Siemens AG

Östl. Rheinbrückenstr. 50

76187 Karlsruhe

Betreuer der DHBW

Prof. Dr. Jürgen Röthig

## Erklärung

Wir versichern hiermit, dass wir unsere Studienarbeit mit dem Thema: „Smart-Home-Lösungen mittels Sprachsteuerung und Raspberry Pi“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

---

Ort    Datum

---

Maximilian Hirte

---

Ort    Datum

---

Robin Warth

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Abkürzungsverzeichnis</b>	<b>VII</b>
<b>1 Einleitung - erster Entwurf</b>	<b>1</b>
<b>2 Aufgabenstellung</b>	<b>2</b>
<b>3 Grundlagen</b>	<b>3</b>
3.1 Smart Home . . . . .	3
3.2 Sprachverarbeitung . . . . .	3
3.3 Analyse und Vergleich von digitalen Assistenten . . . . .	4
3.3.1 Amazon Alexa . . . . .	5
3.3.2 Apple Siri . . . . .	6
3.3.3 Google Assistant . . . . .	7
3.3.4 Microsoft Cortana . . . . .	9
3.3.5 Samsung Bixby . . . . .	10
3.4 Raspberry Pi . . . . .	11
3.5 Funkstandards . . . . .	11
3.6 Development-Tools . . . . .	11
3.6.1 Frameworks . . . . .	11
3.6.2 Amazon Developer Services . . . . .	11
3.7 Gebrauchstauglichkeit . . . . .	12
<b>4 Konzept</b>	<b>14</b>
4.1 Anforderungsanalyse . . . . .	14
4.1.1 Grund der Umsetzung . . . . .	14
4.1.2 Funktionale Anforderungen . . . . .	14
4.1.3 Nicht-Funktionale Anforderungen . . . . .	15
4.2 Herangehensweise . . . . .	16
4.3 Architektur . . . . .	16
4.4 Design . . . . .	16
<b>5 Implementierung</b>	<b>18</b>
5.1 Einrichtung . . . . .	18
5.1.1 Raspberry Pi . . . . .	18
5.1.2 Python . . . . .	18
5.1.3 Node.js . . . . .	19
5.1.4 Cloud 9 . . . . .	19

---

5.1.5	Git . . . . .	19
5.1.6	Amazon Developer Konto . . . . .	19
5.1.7	Alexa . . . . .	19
5.1.8	Amazon Web Services (AWS) Lambda . . . . .	25
5.2	Entwicklung . . . . .	26
5.2.1	AWS Lambda . . . . .	26
<b>6</b>	<b>Ergebnis, Fazit und Ausblick</b>	<b>34</b>
	<b>Literatur</b>	<b>VIII</b>
	<b>Anhang</b>	<b>XI</b>

# Abbildungsverzeichnis

Abb. 4.1: Raspberry Pi-Alexa-Kommunikation . . . . .	17
Abb. 5.1: Setup Skill-Information . . . . .	20
Abb. 5.2: Setup Interaction Modell . . . . .	21
Abb. 5.3: Setup Configuration Endpoints . . . . .	22
Abb. 5.4: Setup Configuration Account Linking . . . . .	23
Abb. 5.5: Setup Test . . . . .	25

## Quellcodeverzeichnis

5.1	Node.js Installieren . . . . .	19
5.2	GitHub Repository . . . . .	19
5.3	Request: Report-State . . . . .	27
5.4	Response: Report-State . . . . .	27
5.5	Request: Discovery . . . . .	28
5.6	Response: Discovery . . . . .	29
5.7	Request: Power-Controller . . . . .	31
5.8	Response: Power-Controller . . . . .	32

# Abkürzungsverzeichnis

<b>API</b>	Application Programming Interface
<b>ARN</b>	Amazon Resource Name
<b>AWS</b>	Amazon Web Services
<b>GPIO</b>	General Purpose Input/Output
<b>HTTP</b>	Hypertext Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>JSON</b>	JavaScript Object Notation
<b>LED</b>	lichtemittierende Diode
<b>LWA</b>	Login With Amazon
<b>SSH</b>	Secure Shell

# 1 Einleitung - erster Entwurf

Wir leben mittlerweile in einer Welt, in der jegliche Teilprozesse des täglichen Lebens bereits automatisiert sind bzw. automatisiert werden sollen, so auch das eigene Zuhause. Seit Anfang des Jahrtausends stellen sich Firmen und Institute die Aufgabe, das intelligente Zuhause zu entwickeln [VOE16] und schon mehrere Jahre ist der Begriff Smart Home in aller Munde. Es können jegliche Systeme innerhalb des Eigenheims zentral gesteuert und mit einander verknüpft werden. Über eine App kann dann der Benutzer seine persönlichen Einstellungen vornehmen und das Lebensgefühl somit auf eine neue Ebene setzen.

In letzter Zeit geht der Trend dahin, diese Hausautomatisierung per Sprachbefehle zu koordinieren. Zu diesem Zweck gibt es verschiedene unabhängige digitale Sprachassistenten, die dem Bewohner das Leben vereinfachen. Besonders Amazon-Alexa und der Google-Assistent ringen im Kampf um die Marktvormachtstellung und überzeugen beide im Bereich der Sprachverständnis und -verarbeitung, aber vor allem auch mit den Fähigkeiten, die sie bieten.

Jedoch bringt eine komplette Wohneinrichtung mit Systemen, die von Haus aus die Verbindung zu den digitalen Assistenten gewährleisten, immense Kosten mit sich. Die ließen sich reduzieren, wenn sich z. B. nicht nur die von Amazon und Google vorgesehenen und zu hohen Preisen angepriesenen Geräte koppeln lassen würden, sondern jegliche Systeme, Geräte und Anlagen, die für den Smart Home-Gebrauch geeignet sind.

Die Motivation dieser Studienarbeit ist somit, eine solche Kostenersparnis zu erzielen. Amazon bietet mit seinem Developer Services die Möglichkeit, Alexa mit einem Raspberry Pi zu verbinden oder sogar direkt auf Kleincomputer zu installieren. Die Aufgabe besteht somit darin den Raspberry Pi mit Alexa über die gängigen Funkstandards mit Geräten zu verbinden, die ansonsten über keinerlei Fähigkeiten verfügen, sich mit digitalen Sprachassistenten zu verknüpfen. Neben der Verringerung der Kosten bietet eine solche Lösung eine weitere Stufe der Individualisierbarkeit und vor allem Geräte- bzw. Herstellerunabhängigkeit für die persönliche Smart Home-Zusammenstellung in den eigenen vier Wänden.



## 2 Aufgabenstellung

- Vergleich von digitalen Assistenten
- mindestens Funksteckdosen
- eigene (Überwachungs-)Kamera
- Kostenersparnis aufzeigen
- eigene Skills zur Ansteuerung

## 3 Grundlagen

In diesem Kapitel werden der Arbeit zu Grunde liegende Begriffe erklärt, Zusammenhänge zwischen diesen aufgezeigt sowie verschiedene Programme vorgestellt und verglichen.

### 3.1 Smart Home

Unter dem Begriff Smart Home wird eine Menge von technischen Systemen, Geräten und Anlagen im Haushalt verstanden, die miteinander interagieren und zentral gesteuert werden. Die damit verbundenen Ziele sind:

- Automatisierung von Alltagsvorgängen,
- Verbesserung von Wohn- und Lebensqualität,
- Erhöhung der häuslichen Sicherheit sowie
- effiziente Energienutzung. [DEV17]

Die gesamte Haushaltstechnik wird somit vernetzt und kann individuell angepasst und automatisiert werden. So können zum Beispiel Geräteeinstellungen vorgenommen werden und über eine Schnittstelle, wie Computer oder Smartphone, verwaltet werden. Es ist somit zum Beispiel möglich seine Heizung per Zeitsteuerung zu kontrollieren oder auch per App auf dem Smartphone die Heizung, das Licht, den Fernseher oder jedes andere angeschlossene Gerät an- und abzuschalten, ohne zu Hause zu sein. Prinzipiell lassen sich alle Geräte ansteuern, die über WLAN, Bluetooth, ZigBee oder andere Funkstandards verfügen. [SCH17]

Die nächste Stufe der Entwicklung ist dann die endgerätlose Steuerung. D. h., die Geräte auch zu steuern, ohne jedes mal den PC oder die App zu benutzen. Der Fokus wird immer mehr auf Gesten- und vor allem auf Sprachsteuerung gelegt, weswegen letzteres auch das Thema dieser Arbeit ist.

### 3.2 Sprachverarbeitung

Seit der zügigen Entwicklung der Computertechnik in der zweiten Hälfte des 19. Jahrhunderts gibt es das Problem und den Wunsch, natürliche Sprache mit Hilfe einer Maschine zu erkennen und zu verarbeiten. Denn in der heutigen Zeit ist Interaktion mit Maschinen so gut wie unumgänglich und der herkömmliche Weg über Tastatur, Maus und Touchscreen ist für den Menschen immer noch unzureichend, denn das Hauptkommunikationsmittel ist die Sprache. [RAO17, S. 1]

Die Sprachverarbeitung unterteilt sich unter anderem in Sprachsynthese, also eine symbolischen Notation in ein Sprachsignal umzusetzen, und in Spracherkennung, die in dieser

Arbeit hauptsächlich beleuchtet wird. Der Schwerpunkt hierbei liegt darin ein Sprachsignal oder auch Laut in eine textliche Form umzusetzen, sodass der Computer diese erkennt. Dies ist jedoch nach wie vor ein sehr großes Problem und wird auch nicht in den nächsten beiden Jahrzehnten so gelöst werden können, dass der Computer die menschliche Sprachwahrnehmungsfähigkeit erreicht. Der Grund dafür ist die Vielfältigkeit und Komplexität der natürlichen Sprache. Auf der Ebene des Wortschatzes ist der Computer schon sehr gut aufgestellt und kann die meisten Wörter erkennen und unterscheiden, doch auf Ebene der Syntaktik und Semantik hat die Maschine bisher keine Chance. Der PC ist somit noch nicht in der Lage die Fähigkeit des Menschen, fortwährend zu untersuchen und zu entscheiden, ob das Gehörte Sinn ergibt, zu simulieren. [PFI17, S. 21]

Aus diesem Grund werden für Spracherkennungsassistenten spezielle Anwendungsszenarien konzipiert. Ein solcher digitale Assistent kann somit zwar keine vollständigen Konversationen führen, aber er verfügt zum Beispiel über die Fähigkeit einen Wetterbericht aus dem Internet zu suchen, wenn er nach dem morgigen Wetter gefragt wird. Durch solche Spezialisierung auf Anwendungsfälle werden bisher gute Resultate erreicht, da es weniger Erkennungsfehler gibt. Weiterhin ist diese Lösung kompakter, da sie sowohl weniger Speicher als auch Rechenleistung benötigt. [PFI17, S. 28]

Im Jahr 2017 hat die Mozilla Foundation das Projekt Common Voice durchgeführt, welches aus 400.000 validierten, englischen Sprachaufnahmen von 200.000 Personen besteht. Diese Aufnahmen ließen sie durch verschiedene Spracherkennungssoftware laufen, um somit die Fehlerquote bei der automatisierten Übersetzung der Sprachaufnahmen in Text zu minimieren. Ziel war es, eine Quote unter der Zehn-Prozent-Marke zu erreichen. Das erscheint für uns Menschen relativ viel, aber laut Forschungsergebnissen, auf die sich Mozilla beruft, liegt die Fehlerquote des menschlichen Sprachverständnisses bei rund 6%. [MOZ17] Aus diesen Werten lässt sich also schließen, dass vorhandene Software im Bereich der Spracherkennung schon den menschlichen Fähigkeiten annähert.

Die maschinelle Erkennung und Verarbeitung erfolgt über Sprachsignale. Die genauen Algorithmen sowie der gesamte interne technische Ablauf der digitalen Assistenten sind nicht Teil dieser Arbeit und werden nicht weiter erläutert.

### 3.3 Analyse und Vergleich von digitalen Assistenten

Ein digitaler Assistent (oft auch Intelligenter Persönlicher Assistent oder auch Sprachassistent) ist ein Programm, dass durch Hilfe von Sprachverarbeitung gesprochenen Anweisungen Folge leisten kann und zudem noch in menschlicher Sprache, durch eine meist weibliche Stimme, antworten kann. Die häufigsten Anwendungsfälle stellen dabei einfache Sprachsuchen im Internet oder Abarbeiten von einfachen Aufgaben, wie Schreiben und Versenden von Nachrichten oder Erstellen von Kalendereinträgen bzw. Erinnerungen dar.

Im Folgenden werden die zur Zeit fünf bedeutendsten Assistenten genauer untersucht und anhand folgender Kriterien miteinander verglichen:

### 3.3.1 Amazon Alexa

Im Jahr 2015 kam mit den smarten Echo-Lautsprechern auch Alexa auf den Markt und reiht sich somit in der Historie hinter Siri und Google Now ein. Die größte Verbreitung genießt es auf den Echo-Geräten, aber auch auf weiteren Amazon-Geräten, wie Fire-TV oder Fire-Tablets. Der digitale Assistent lässt sich aber auch als App auf dem Smartphone installieren. [BAG17, S. 65]

Die Tester von Heise online, die die vier bekanntesten Sprachassistenten unter die Lupe genommen haben, beschrieben Alexa mit einer fast beängstigend guten Spracherkennung und mit einer angenehmen, freundlichen Stimme. Allgemein punktet der Assistent aus dem Amazonhaus in der Art der Kommunikation. Als tägliche Begleiterin im eigenen Heim ist sie sehr sympatisch und reagiert sogar auch oft mit Humor. Sie kann sogar ein bisschen wie Yoda reden und weiß, dass 42 die Antwort auf alles ist. Durch das alles hat sie auf Basis der Kommunikation einen Vorsprung gegenüber dem Hauptkonkurrenten Google. [BLE17, S. 81]

Auf Wissensfragen kann Alexa meist auch gut und richtig antworten. Jedoch sind Navigationsinformationen, wie der schnellste Weg nach Hamburg, noch sehr ausbaufähig. Sie ist aber auf ihren Echo-Lautsprechern bisher auch größtenteils auf den stationären Betrieb ausgelegt. Somit ist der Assistent im Bereich des Lokalservices mit z. B. dem Kinoprogramm der Stadt besser ausgerüstet, aber manchmal muss dennoch die dazugehörige App geöffnet werden. [BLE17, S. 81] Seit neustem gibt es dazu auch den Echo Show, ein zum Lautsprecher gehörendes Tablet, auf denen die Informationen noch visualisiert werden können. [BAG17, S. 65]

Derartige Wissenslücken lassen sich durch erweiterbare Skills schließen. Mittlerweile gibt es ca. 3000 deutschsprachige und insgesamt über 16000 Skills, die das Leben durch z. B. bestimmte Radiosender, Online-Kochbücher oder Fahrpläne per Sprachzugriff vereinfachen und verbessern. Solche Skills kann auch jeder Benutzer selber implementieren, indem er sich kostenlos bei den Amazon Developer Services registriert und somit zum Entwickler wird. Die Hauptstärke von Alexa liegt also genau in diesen erweiterbaren Skills, wodurch der Assistent immer weiter dazulernt. [BLE17, S. 81]

Das Nachsehen gegenüber der Konkurrenz hat Alexa vor allem beim Fremdsprachenverständnis. Versucht der Anwender z. B. bei einer auf Deutsch eingestellten Station einen englischen Titel per Spracheingabe auf dem Musik-Streaming-Dienst zu starten, kann es passieren, dass Alexa nicht das richtige Lied oder den passenden Interpreten abspielt. [BLE17, S. 81]

Weiterhin ist, obwohl der Assistent auf den Online-Shopping-Riesen Amazon optimiert sein soll, nicht wirklich zufriedenstellend. Oftmals lassen sich die Artikel nur in die Merkliste legen und nicht direkt bestellen. Vor allem bei Bekleidung und Schuhen tritt dieses Problem auf. Wenn der Benutzer jedoch einen Artikel bestellen will, den er in der Vergangenheit schon einmal erworben hat, hat Alexa keine Probleme. [BLE17, S. 81]

Ein letztes Manko von Alexa ist das Kontextverständnis. So kann sie nicht aufeinanderfolgende Fragen miteinander verknüpfen, was eine der Stärken von Google und dessen Deep Learning ist. Dennoch hat Amazon im Bereich der digitalen Assistenten noch die Nase vorn, da die Echo-Lautsprecher einige Monate vor dem Google Home auf dem Markt kam, somit länger Erfahrung sammeln konnte und vor allem schon genügend Skills existieren, die Alexa kontinuierlich verbessern. [CLA18]

Amazon hat das Marktpotential im Bereich der Sprachassistenten erkannt und möchte somit sein Produkt so weit wie möglich verbreiten und auf so viele Geräte wie möglich anzubieten. Demnächst soll Alexa demzufolge auch direkt in Autos verbaut werden und somit die Hands-Free-Steuerung während der Fahrt erleichtern. Seat und BMW wollen den Assistenten in mehreren Modellen einbauen. Außerdem sprachen die Firmenbosse davon eine Alexa-Cortana-Verbindung zu erstellen und das hieße, dass Alexa dann auch auf Windows-Geräten erreichbar wäre. [BAG17, S. 66]

Amazon hat somit ein Produkt geschaffen, dass dem Benutzer das Leben vereinfachen kann und auch die Zukunft sein wird. Auch im Bereich von Smart Home bietet Alexa eine gute Unterstützung. Es gibt schon verschiedene Geräte die automatisch mit dem Assistenten gekoppelt werden können, aber durch die benutzerdefinierten Skills können auch eigene Verbindungen über den Raspberry Pi ganz andere Geräte angesteuert werden und aus diesem Grund ist Alexa auch ein möglicher Kandidat zur Verwendung in dieser Studienarbeit.

### 3.3.2 Apple Siri

Siri wurde erstmalig im Jahre 2011 veröffentlicht und ist somit auch die älteste Assistentin. Sie ist nur auf den Apple-eigenen Plattformen iOS, macOS, watchOS sowie tvOS verfügbar, aber seit Februar 2018 gibt es auch analog zu den Amazon und Google Lautsprechern einen Apple HomePod, auf dem mit Siri kommuniziert werden kann. [BAG17, S. 68]

Die Vorreiterin punktet vor allem durch die sehr gute Einbettung in das System an sich und der damit verbundenen Interaktion mit den von Apple mitgelieferten Apps auf dem Smartphone. Ohne Probleme lassen sich durch einfache Fragen bzw. Aufforderungen der Taschenrechner öffnen und Therme berechnen, eine Wettervorhersage einholen, neue Termine in den Kalender eintragen oder die Verabredungen für den heutigen Tag ausgeben lassen. Weiterhin können Emails oder Notizen diktiert, abgespeichert oder versendet werden. Zwar gibt es mittlerweile eine Unterstützung für häufig genutzte Apps wie Messenger-Dienste wie WhatsApp, welche auch ausgereift sind, wenn der Benutzer aber Dienste von externen Anwendungen anfordert, versucht Siri so gut wie immer auf Apple-eigene Apps zuzugreifen, weswegen die angeforderten Ergebnisse oftmals unzureichend sind. Der Assistent ist ursprünglich nur für die Ersetzung der Touchbedienung auf den eigenen Geräten entwickelt wurden und dies ist auch sehr gut gelungen, wodurch auch der Siris Erfolg zu erklären ist. [BLE17, S. 85]

Im Bereich der Sprachausgabe überzeugt durch eine angenehme und natürliche Stimme, welche natürlich über die Jahre hinweg immer weiter verbessert werden konnte. Die Tester von Heise online beschrieben Siri als vertrauenerweckend sogar fast freundlich, was auch mit der Personalisierung zu tun hat. Denn der Benutzer kann sich von dem Assistenten mit seinem Namen ansprechen lassen. Zudem stellt sich Siri auf die Stimme des Anwenders ein, wodurch das Benutzererlebnis zusätzlich verbessert wird. Für Humor ist sie aber eher nicht offen, weswegen sie in normaler Kommunikation minder unterhaltsam abschneidet als ihrer Konkurrenten. Außerdem kann in den Einstellungen, wenn bevorzugt, auch genderkorrekt eine Männerstimme favorisiert werden, was bei weitem nicht alle digitale Assistenten unterstützen. [BLE17, S. 85]

Minuspunkte sammelt Siri im Bereich von Übersetzungen sowie im Fremdsprachenverständnis. In diesen Punkten hinkt sie im Vergleich mit den anderen Assistenten weit hinterher bzw. kann Siri noch gar nicht einzelne Wörter in andere Sprachen übersetzen. Zudem weist die Apple-Lösung gegenüber Google und Amazon klare Wissenslücken auf. Oftmals bekommt der Anwender gar keine oder falsche Antworten (nicht zur Fragestellung passend) und Wissensfragen werden willkürlich sprachlich oder visuell beantwortet, was für den Benutzer unverständlich und auf Dauer nervig ist. Auch ortsbezogene Suchanfragen sind noch nicht wirklich ausgereift. Zwar funktionieren Navigation und Restaurantvorschläge in der Nähe einwandfrei, jedoch können auf Dinge, wie örtliches Kinoprogramm nur selten zugegriffen werden. [BLE17, S. 85]

Zusammenfassend lässt sich sagen, dass Siri als älteste Assistentin ausgereift ist, aber dennoch einige Schwächen aufweist und sowohl Amazon als auch Google sie überholt haben. Wie von Apple bekannt, funktioniert alles solange gut, sofern Apple-Produkte und Software benutzt wird. So auch im Bereich von Smart Home. Auch hier ist der Anwender auf das firmeneigene HomeKit angewiesen, was auch in diesem Bereich eher Minuspunkte sammelt und sich somit nicht für den Gebrauch in dieser Studienarbeit anbietet. [BLE17, S. 85]

### 3.3.3 Google Assistant

Schon im Jahr 2012 kam die Vorversion des heutigen Google Assistant, Google Now, auf den Markt und hat somit sich auch schon einige Jahre auf dem Bereich der Spracherkennung und -verarbeitung entwickeln können. Der Assistent ist auf jeglichen Android-Geräten seit dem 6.0er Update erhältlich und mittlerweile gibt es sogar auch eine iOS-Version. Nach dem der Internetriese immer mehr Marktanteile an Amazon abgeben musste und die Google-Suchanfragen seit der Veröffentlichung des Echo-Lautsprechers zurückgingen, erkannten sie das Potenzial und brachten Ende 2016 (2017 mit deutscher Sprachunterstützung) mit ihrem Google Home den eigenen smarten Lautsprecher auf den Markt. [BAG17, S. 66 f]

Der Assistent punktet vor allem durch herausragende Smartphone-Integration, sodass

die applikationsübergreifende Interaktion sehr gut funktioniert. Per Sprache kann die Kamera geöffnet werden, eine Whatsapp-Nachricht diktiert und versendet werden und sogar der Flugmodus aktiviert werden, was ein eigenes Schließen bzw. Beenden des Assistenten mit sich bringt. [BLE17, S. 84]

Weitere Vorteile von Google sind auch ganz einfach an dessen riesigen Datenfundus zurückzuführen und somit ist der Google Assistant auch im Allgemeinwissen marktführend. Er kann vor allem bei ortsbezogenen Informationen, wie Kinoprogramm oder auch Navigation, punkten und entscheidet dabei auch so gut wie immer richtig, ob das Ergebnis mündlich oder visuell ausgegeben wird. Weiterhin zählt der Assistent genauso wie Cortana zu den Sprachgenies und kann sogar ganze Sätze übersetzen. [BLE17, S. 84]

Ein Alleinstellungsmerkmal ist die Stimmenerkennung, denn Google ist bisher als einziges in der Lage, Stimmen zu unterscheiden und kann somit sogar mehrere Benutzer anlegen. [CLA18] Hinzu kommt, dass der Assistant kontextbezogene Nachfragen ermöglicht. Google analysiert jede einzelne Frage und merkt sich die Informationen darin, um später einen Zusammenhang zu folgenden Fragen zu haben. Das beherrscht der Assistent auch besser als die Konkurrenz und somit lassen sich annähernd vollständige Gespräche führen. [BLE17, S. 84]

Aufgrund der langen Laufzeit seit der Erstveröffentlichung ist auch die Sprachsynthese so gut, dass man alles versteht, aber auch noch in der aktuellen Version eine recht roboterhafte Stimme hat. Die Heise online Tester beschrieben die Stimme als etwas hochnäsig und humorarm. Da macht Alexa einen wesentlich besseren Eindruck, da in der Kommunikation das Gefühl eine große Rolle spielt und ein Benutzer natürlich lieber mit jemanden redet, der eine freundlichere Stimme hat bzw. den er mehr mag. [BLE17, S. 84]

Analog zu den Alexa Skills gibt es bei Google die (Voice) Actions bei denen benutzerdefinierte Anwendungen implementiert und hinzugefügt werden. Bislang gibt es aber bei weitem weniger vorhandene und vor allem auch kaum deutschsprachige Actions, da Amazon mit ihrem Produkt einfach viel früher dran war und hier klare Marktvorteile genießt. [CLA18]

Im Bereich von Smart Home nehmen sich beide Assistenten, also von Google und Amazon, nicht viel. Sie unterstützen ähnlich viele Geräte und lassen sich auch noch erweitern. Vorteile hat nur Google bei der Sprachbedienung. Während bei Google eine einfache Phrase wie: „Hey Google, mach es wärmer.“ ausreicht, muss bei Alexa immer, wie bei jedem anderen Fall auch, der direkte Skill sehr präzise angesprochen werden, wie z. B.: „Alexa, sage Raspberry er soll die Temperatur im Wohnzimmer auf 21,5 Grad einstellen.“ Demzufolge ist Alexas Steuerung in vielen Fällen umständlicher. [CLA18]

Der Google Assistant und Alexa dominieren aktuell größtenteils den Markt. Noch hat Alexa aufgrund der früheren Veröffentlichung gewisse Vorteile an Verbreitungsgrad, die es für Google gilt aufzuholen, was durch den riesigen Deep Learning Mechanismus nicht mehr lange dauern wird. Für diese Studienarbeit kommen somit beide Assistenten durch die freie Erweiterbarkeit in Frage.

### 3.3.4 Microsoft Cortana

Im Jahr 2014 kam mit Cortana Microsofts Antwort im Bereich der Sprachassistenten auf den Markt. Ihr hauptsächliche Verbreitung erstreckt sich über Geräte mit einem Windows 10 Betriebssystem, da der Assistent dabei direkt mitgeliefert wird.

Cortanas Alleinstellungsmerkmal gegenüber ihrer Konkurrenten ist ihre Übersetzungsfähigkeit, denn sie kann neben Wörtern auch Sätze per Sprachbefehl durch den Microsoft Translator in verschiedenste Sprachen übersetzen und in den meisten Fällen, das Ergebnis auch vorlesen. In diesen Tests schneidet der Assistent sogar besser ab, als Googles Produkt. Ähnlich wie Siri ist Cortana auf das Betriebssystem sehr gut zugeschnitten und somit lassen sich Programme ohne Probleme per Sprachsteuerung öffnen. [BLE17, S. 82]

Im Vergleich zu den Konkurrenzprodukten muss Microsoft aber in einigen Punkten zurückstecken. Zwar wirken Cortanas Antworten nicht künstlich, sondern sie hat eher ein Stückweit Empathie und Mitgefühl, aber dennoch klingt ihre Stimme ein etwas hölzern, was für den Benutzer auf Dauer etwas gewöhnungsbedürftig ist. [BLE17, S. 82] Weiterhin ist die Anzahl der unterstützten Sprachbefehle gegenüber der Assistenten der Konkurrenz sehr eingeschränkt. Um diese Lücke zu stopfen gibt es eine von Microsoft bereitgestellte Programmierschnittstelle, aber im Gegensatz zu Amazon und Google ließen sich hier noch nicht viele externe Programmierer finden. Somit gibt es immer noch weniger als 100 Skills, die hinzugekommen sind und das auch nur auf Englisch und bisher keine deutschsprachigen. [BAG17, S. 67]

Auch im Bereich von Sprachsuchen hat Cortana das Nachsehen gegenüber Google, Amazon, Apple und Samsung. Viele Antworten kann sie nicht selbst geben und öffnet dann Browser und versucht ihr Glück via Suchmaschine. Dabei benutzt sie immer Edge und Bing, was sich auch nicht umstellen lässt. Minuspunkte gibt es nicht nur für diesen Zwang, sondern auch dafür, dass sie die Antworten, welche sie nicht selber beantworten kann, oftmals im Browser auch gar nicht vorliest. Edge muss der Benutzer auch selber wieder schließen, was nicht sonderlich benutzerfreundlich ist. [BLE17, S. 82]

Oftmals sind zudem auch die Sprachsuchergebnisse nicht zufriedenstellend. Auf die Frage „Seit wann gibt es dich?“ öffnet sie die Homepage [www.seid-seit.de](http://www.seid-seit.de), was natürlich erstens nichts mit der Fragestellung zu tun hat und zweitens der Benutzer zusätzlich noch das Fenster selbst wieder schließen muss.

Dies alles wird auch der Grund sein, warum Cortana gegenüber den anderen Assistenten wenig Beachtung geschenkt wird. Jedoch soll noch im Laufe des Jahres 2018 eine Kooperation zwischen den beiden Freundinnen Cortana und Alexa ermöglicht werden. Somit könnte Alexa mit einem Sprachbefehl aufgefordert werden Cortana zu öffnen und andersherum. Die Entwickler erhoffen sich dadurch die Schwächen gegenseitig auszumärzen. Somit könnte der Benutzer dann über Cortana bei Amazon einkaufen und einer der wichtigsten Punkte ist, dass Cortana somit erstmalig auf einem intelligenten Echo-Lautsprecher verfügbar sein wird und somit auch zum ersten Mal mit Smart Home in Verbindung gebracht wird. Bis



dato kommt die Verwendung von Cortana aber nicht für diese Studienarbeit in Frage.

### 3.3.5 Samsung Bixby

Der jüngste unter den digitalen Assistenten ist die im Jahr 2017 von Samsung veröffentlichte Software Bixby. Diese ist bisher nur auf den Samsung Smartphones Galaxy S8 und durch eine Zusatzinstallation auf den Samsung Galaxy S7 Geräten verfügbar und genießt damit eine sehr beschränkte Verbreitung. Die Installation besteht aus Bixby Voice, den eigentlichen Sprachassistenten, Bixby Home, einer digitalen Pinnwand und Organisationseinheit und Bixby Vision, welcher ein Objekt mit der Kamera bzw. in der Galerie scannen kann und zu diesem Objekt im Internet Informationen sammelt bzw. Shops sucht, die diesen verkaufen. [WIG17]

Zunächst wurde der Assistent nur auf Koreanisch und Chinesisch veröffentlicht, aber mittlerweile wurde auch die englische Sprache hinzugefügt. Deutsch fehlt noch immer, weswegen der geringe Verbreitungsgrad hierzulande zu erklären ist.

Ähnlich wie Siri kann der Benutzer bei Bixby zwischen verschiedenen Stimmen wählen. Tester beschrieben die Stimmen Stephanie mit einem angenehmen Klang, John mit einer sympathischen Stimme und Julia mit einer tieferen Aussprache. Alle drei zeichnen sich aber jeweils durch eine klare und deutliche Kommunikation aus. [REI17]

Beim Sprachverständnis hat der Assistent jedoch noch seine Probleme. In den Einstellungen kann zwischen drei Empfindlichkeitsstufen ausgewählt werden, aber die höchste kommt bei weitem nicht an Google heran. Somit muss das Schlüsselwort "Hi Bixby" öfters lauter wiederholt werden. Weiterhin gibt es einige Schwierigkeiten mit Dialekt, aber dafür wird die Stimme mit der Zeit gelernt und auch ein manuelles Sprachtraining ist möglich. [REI17]

Im Bereich von übergreifender Interaktion kann Bixby ordentlich punkten. Allgemeine Dinge wie Wecker, Wetter sowie Youtube funktionieren einwandfrei und teilweise sogar besser als alle anderen Konkurrenzprodukte. So schießt der Befehl "take a photo" direkt ein Foto und öffnet nicht nur wie Google die Kamera. In den FeedbackEinstellungen kann der Benutzer zu dem entscheiden, ob die Antworten auf Fragen akustisch, visuell oder ausschließlich textlich sowie kurz oder lang beantwortet werden sollen. [REI17]

Allgemein lässt sich sagen, dass Bixby bisher wenig Zeit hatte sich zu beweisen. Dafür müsste es aber auch für mehr Sprachräume und Geräte zugänglich gemacht werden. Zudem hat es einige Nachteile, da auf denselben Smartphones auf denen Bixby erhältlich ist auch Google mit ihrem Produkt vertreten ist und im direkten Vergleich durch das schlechtere Sprachverständnis und fehlendes Verknüpfen mit vorhergehenden Fragen schlechter abschneidet. Dennoch hat Bixby großes Potential und kann durch mehr Erfahrung auf dem Markt, den Vorsprung der Konkurrenz aufholen. [HAR17] Da bisher auch keine Smart Home Unterstützung verfügbar ist, scheidet der Assistent auch für diese Studienarbeit aus.

## 3.4 Raspberry Pi

[folgt]

## 3.5 Funkstandards

ISM-Band von 433,05 bis 434,79 MHz

Industrial, Scientific, Medical Band

Wir benutzen 433 MHz (70-Zentimeter-Band)

Betriebsfunk und Amateurfunk, Handfunkgeräte

Als ISM-Band für z.B. Funkswitcher, Funkthermometer, Funksteckdosen

Max. Leistung von 10mW

Reichweite 0,3 in Stadt, 2,5 auf freier Fläche (quasioptisch)

69 Kanäle, gebührenfrei, anmeldefrei

«Non specific Short-Range-Devices» (SRD) – Geräte geringer Sendeleistung und kurzer Sendereichweite – dürfen ohne weitere Frequenzuteilung betrieben werden, sofern sie die technischen Voraussetzungen erfüllen

Geräte für Anwendungen kommen sich selten in die Quere, trotz große Verbreitung (Garagentoröffner, Autoschlüssel..), da Aktivierungsradius gering (<100m) und die Nutzungsdauer kurz (0,1 bis 3s). Trotzdem kann vorkommen dass 433,92MHz blockiert ist, wegen z.B. Funkkopfhörer, der im Umkreis von 50m auf derselben Frequenz arbeitet  
Nachteile: Reichweite, Kosten und Stromverbrauch

Nutzung nicht beschränkt -> Störungen durch Nachbarn wahrscheinlich  
Sichere Übertragung nur bei leistungsfähiger Kanalcodierung, Module mit mehreren Kanälen haben deutliche Vorteile, indem sie auf andere Frequenz ausweichen können  
(<http://rn-wissen.de/wiki/index.php?title=Funkmodule>)

## 3.6 Development-Tools

[folgt]

### 3.6.1 Frameworks

[folgt]

### 3.6.2 Amazon Developer Services

Mit den Amazon Developer Services wird jedem Benutzer, der eine Mailadresse besitzt, die mit einem Amazon-Konto verknüpft ist, die Möglichkeit geboten, eigene Applikationen für jegliche Geräte auf Android oder iOS basierend oder für die Amazon eigenen Fire Tablets und Fernseher kostenlos zu entwickeln, zu builden, zu testen und sogar über den Amazon

Appstore zu vertreiben. Nach der Registrierung eines Amazon Developer Accounts stehen dem Entwickler etliche Services und APIs zur Verfügung, die direkt eingebunden und benutzt werden können. Das umschließt beispielsweise vorgefertigte Login-Formulare durch Amazon-Konten, Werbung für die eigene Applikation auf den Amazon-Fire-Geräten zu schalten oder Multi-Screen-Möglichkeiten mit zwei-Wege-Kommunikation zwischen Fire TV und der Smartphone Anwendung. Weiterhin werden verschiedenste Plugins und Extensions wie Unity, Apache Cordova oder Xamarin unterstützt. [ROU17a]

Das für dieses Projekt wichtigste Werkzeug ist das Alexa Skills Kit. Dieses erlaubt eigene Skills, also Fähigkeiten, dass Alexa bestimmte eigens konzipierte Anwendungsfälle durch bestimmte Wörter erkennt und auf diese reagieren kann, selbst zu erstellen bzw. zu programmieren. Diese selbsterstellten Skills können dann auch über den Alexa Skills Store vertrieben werden. [ROU17b]

Über die eigenen Skills können dann über die Verbindung mittels des Raspberry Pi die angeschlossenen Geräte wie Funksteckdosen oder die Kamera, welche sonst nicht von Alexa aus erreichbar wären, angesprochen werden.

### 3.7 Gebrauchstauglichkeit

Gemäß „DIN EN ISO 9241“ [ISO06] sind die Kriterien für die Gebrauchstauglichkeit (Usability) über die sieben „Grundsätze der Dialoggestaltung“ beschrieben:

- Aufgabenangemessenheit
  - „Ein interaktives System ist aufgabenangemessen, wenn es den Benutzer unterstützt, seine Arbeitsaufgabe zu erledigen.“
- Selbstbeschreibungsfähigkeit
  - „Ein Dialog ist in dem Maße selbstbeschreibungsfähig, in dem für den Benutzer zu jeder Zeit offensichtlich ist, in welchem Dialog, an welcher Stelle im Dialog er sich befindet, welche Handlungen unternommen werden können und wie diese ausgeführt werden können.“
- Erwartungskonformität
  - „Ein Dialog ist erwartungskonform, wenn er den aus dem Nutzungskontext heraus vorhersehbaren Benutzerbelangen sowie allgemein anerkannten Konventionen entspricht.“
- Lernförderlichkeit
  - „Ein Dialog ist lernförderlich, wenn er den Benutzer beim Erlernen der Nutzung des interaktiven Systems unterstützt und anleitet.“
- Steuerbarkeit
  - „Ein Dialog ist steuerbar, wenn der Benutzer in der Lage ist, den Dialogablauf zu starten sowie seine Richtung und Geschwindigkeit zu beeinflussen, bis das Ziel erreicht ist.“

- Fehlertoleranz
  - „Ein Dialog ist fehlertolerant, wenn das beabsichtigte Arbeitsergebnis trotz erkennbar fehlerhafter Eingaben entweder mit keinem oder mit minimalem Korrekturaufwand seitens des Benutzers erreicht werden kann.“
- Individualisierbarkeit
  - „Ein Dialog ist individualisierbar, wenn Benutzer die Mensch- System-Interaktion und die Darstellung von Informationen ändern können, um diese an ihre individuellen Fähigkeiten und Bedürfnisse anzupassen.“

## 4 Konzept

In diesem Abschnitt geht es um den grundsätzlichen Entwurf und das Konzept des praktischen Teils der Studienarbeit. Hier wird dargestellt, welches konkrete Ziel mit der Studienarbeit verfolgt und mit welchen Mitteln es erreicht werden soll.

### 4.1 Anforderungsanalyse

In der Anforderungsanalyse wird beschrieben, welche funktionalen und welche nicht-funktionalen Anforderungen an das Projekt gestellt sind.

#### 4.1.1 Grund der Umsetzung

[folgt]

#### 4.1.2 Funktionale Anforderungen

Nach Beendigung des Projekts soll die Möglichkeit bestehen, Geräte per Sprachbefehle ein und aus zu schalten. Einige der Geräte sollen über Funksteckdosen, als Steuerelement, an das Smart Home angebunden werden. Hierfür stehen beliebig viele Geräte zur Verfügung, wie beispielsweise Lampen, LED-Streifen, Ventilatoren und weitere Stromverbraucher.

Zudem soll eine Kamera, angeschlossen an einen Raspberry Pi, über Sprachbefehle bedient werden können. Es sollen verschiedene Einsatzmöglichkeiten für die Kamera geschaffen werden, wie Zeitraffer-Aufnahmen, Überwachungskamera oder stationärer Fotoapparat mit Fernsteuerung über Sprachbefehl und Webinterface.

Die Funksteckdosen, die Kamera und eventuell weitere Geräte sollen neben Sprachbefehlen auch über eine webbasierte Hauszentralsteuerung erreichbar sein. Diese Hauszentralsteuerung muss von beliebigen und webbrowsersfähigen Geräten, mit Anbindung ans Heimnetz, ansteuerbar sein. Die webbasierte Hauszentralsteuerung soll zudem über ein passendes Untermenü die zur Verfügung stehenden Funktionen aufzeigen und eine Übersicht über mögliche Sprachbefehle bieten. Über die webbasierte Hauszentralsteuerung sollen die selben Funktionen vorliegen, welche auch über Sprachbefehle zur Verfügung stehen.

Zur einfacheren Ansteuerung, erleichterter Bedienbarkeit und Komfort sollen personalisierte Befehle und Befehlsprotokolle erstellt werden können. Idealerweise im Webinterface der Hauszentralsteuerung. Zu diesen Befehlen und Protokollen zählen Befehle zur parallelen Ansteuerung von verschiedenen Geräten, Zeitschaltuhren sowie Protokolle für Energiesparmaßnahmen und Sicherheit.

(- Ansteuern von Geräten mittels Infrarotsender)

### 4.1.3 Nicht-Funktionale Anforderungen

Im Folgenden werden die qualitativen Anforderungen bzw. die Qualitätsattribute des Projektes genauer definiert.

#### Systemumgebung

Der Zugang zum System erfolgt über ein internes lokales Heimnetzwerk, über Webbrowser und Sprachbefehle. Funktionsrealisierung erfolgt mittels Node.js Server auf Raspberry Pi, Alexa Voice Service unterstützendem Gerät, wie zum Beispiel Amazon Echo Dot, und mit an Raspberry Pi verbundenen Geräten.

Das Aufrufen der webbasierten Heimzentralsteuerung ist betriebssystemunabhängig und kann von beliebigen Geräten mit Javascript fähigem Webbrowser erfolgen, wie beispielsweise Smartphone, Tablet oder Heimcomputer.

#### Performance

Das System soll direkt auf Sprachkommandos und User-Inputs mittels Webinterface reagieren. Eine maximale Verzögerung von zwei Sekunden, bis der Benutzer ein auditives oder visuelles Feedback bekommt, wird angestrebt.

#### Bedienbarkeit

Das System soll einfach und intuitiv bedienbar sein und daher möglichst genau den Anforderungen (Siehe 3.7 Gebrauchstauglichkeit, ISO 9241-110:2006) gerecht werden.

Das Modul soll den Benutzer unterstützen seine Arbeit zu machen und darf dabei keine unnötigen Informationen anzeigen oder dem Benutzer überflüssige Arbeitsschritte auferlegen. Es soll dem Benutzer in jedem Moment klar sein, was er tun kann und was er zu tun hat, um sein Ziel zu erreichen. Informationen wie Rückmeldungen und Beschriftungen von Bedienelementen sowie Funktionen sollen auf den Benutzer natürlich wirken und ihm allgemein bekannt vorkommen. Hinweise sollen dem Benutzer helfen die Bedienoberfläche der webbasierten Heimzentralsteuerung und der Sprachsteuerung kennenzulernen und ihre Funktionen zu verstehen. Bei falscher Bedienung sollen Fehler vom System erkannt werden, dem Benutzer klar gemacht werden, was schiefgelaufen ist und ihm somit helfen es beim nächsten Versuch besser zu machen. Der Benutzer muss immer die Kontrolle über das System haben um sein Smart Home zu bedienen und ungewünschtes Verhalten abwenden können. Zudem soll der Benutzer sein Smart Home individualisieren können, daher muss die Möglichkeit zur individuellen Benennung von Befehlen und somit auch Geräten geschaffen werden.

## **Zuverlässigkeit**

Das System, inklusive Sprach- und Webservice, muss allzeit verfügbar sein und nach Unterbrechung, wie zum Beispiel durch einen Stromausfall schnell und möglichst autonom anlaufen. Für Aktionen wie Sprachbefehle oder weitere, durch den Benutzer durchgeführte, Aktionen besteht eine angestrebte Fehlertoleranz von unter zehn Prozent, damit das System allzeit zuverlässig seine Arbeit verrichtet.

## **4.2 Herangehensweise**

- Planung: Anforderungsanalyse aufstellen und Design / Technischer Entwurf festlegen
- mit Raspberry Pi Geräte an steuern (zunächst Funksteckdosen)
- Raspberry Pi mit Alexa verbinden
- eigene Skills implementieren und Testen-> Proof of Concept abgeschlossen
- Repertoire auf Kamera erweitern

## **4.3 Architektur**

Das System ist unterteilt in einer Art von Client-Server-Architektur. Der Client wird hierbei repräsentiert von dem Alexa Voice Service und der Heimzentralsteuerung im Webbrowser, da diese in erster Linie dazu dienen, Kommandos vom Benutzer entgegen zu nehmen und zudem Informationen über das System bereit zu stellen.

Die Server-Seite wird vertreten vom Raspberry Pi und dem darauf laufenden Node.js Laufzeitumgebung mit Express.js Webserver. Über den Raspberry Pi werden zudem die Smart Home Geräte angesteuert und deren Logik hinterlegt.

Die Verbindung von Server und Client erfolgt über HTTP- und TCP-Protokoll.

## **4.4 Design**

Auf Abbildung 4.1 kann man die Trennung und den Zusammenhang zwischen Raspberry Pi (Server) und Amazon Echo Dot (Client) [ECHO] erkennen.



Abb. 4.1: Raspberry Pi-Alexa-Kommunikation

Auf dem Raspberry Pi [RASPa] befindet sich die Node.js [NODEa] Laufzeitumgebung mit Express.js Webserver. Der Express.js Webserver kommuniziert über HTTP mit dem Alexa Voice Service über ein neues Alexa Skill [ALEXA], welches speziell für die Anbindung von Raspberry Pi geschaffen wird.



## 5 Implementierung

In diesem Kapitel werden die einzelnen Entwicklungsschritte der in der Architektur und Design vorgestellten Elemente genauer, aufgeschlüsselt, dargestellt.

### 5.1 Einrichtung

Im Folgendem wird die Einrichtung der gewählten Systeme, Plattformen und Server erläutert.

#### 5.1.1 Raspberry Pi

Zunächst muss der Raspberry Pi eingerichtet werden. [W3SCHa] Die Entscheidung für die Geräteauswahl fiel auf den Raspberry Pi 3 Model B. Das Projekt ist aber auch auf alle anderen Raspberry Pi Modellen portierbar.

Für eine einfache Bedienung des Raspberry Pi kommt in diesem Projekt das „Raspbian Stretch with desktop“ [RASPb] zum Einsatz. Solange die Ansteuerung der GPIO Ports gewährleistet ist, kann auch ein beliebiges anderes Betriebssystem auf dem Raspberry Pi eingerichtet werden.

Um ein Betriebssystem für den Raspberry Pi zu installieren, muss das Betriebssystem auf der MicroSD-Karte des Raspberry Pi installiert („geflasht“) werden. Am einfachsten geht dies mit dem kostenlosen Programm Etcher. [ETC17] Etcher ist für Windows, Linux und MacOS erhältlich. In Etcher muss lediglich das Medium (die MicroSD-Karte) sowie ein Image zur Installation ausgewählt und die Installation gestartet werden. Sobald Etcher das Betriebssystem auf dem Raspberry Pi installiert hat, kann die MicroSD-Card in den Raspberry Pi gesteckt werden und von dort direkt gebootet werden.

Der Raspberry Pi kann entweder über direkt angeschlossene Peripheriegeräte bedient werden oder per Remote. Für die Bedienung über eine SSH empfiehlt sich Putty. [PuTTY] In unserem Projekt wurde außerdem RealVNC [RVNC] verwendet, welches auf dem Raspbian Stretch bereits vorinstalliert ist. RealVNC hat gegenüber SSH den Vorteil, dass der gesamte Desktop über Remote bedient werden kann.

#### 5.1.2 Python

[folgt]

### 5.1.3 Node.js

Für das Projekt wird Node.js in Version 9.x verwendet. Um das aktuellste Node.js v9.x zu installieren führt man folgende Befehle aus: [NODEb]

```
curl -sL https://deb.nodesource.com/setup/_9.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

Quellcode 5.1: Node.js Installieren

### 5.1.4 Cloud 9

Für die Entwicklung auf dem Raspberry Pi wurde die Web-IDE Cloud 9 [C9.io] [LUN17] verwendet, welche auf dem Raspberry Pi als Server installiert ist. Die Web-IDE ermöglicht besonders einfaches Entwickeln und Testen über jeden Webbrowser auf Computern im selben Netzwerk wie der Raspberry Pi.

### 5.1.5 Git

Zur Sicherung und Versionierung wurde ein GitHub Repository eingerichtet.

```
git clone https://github.com/RobinWarth/Smart-Home-Solutions.git  
git remote add origin https://github.com/RobinWarth/Smart-Home-Solutions.git
```

Quellcode 5.2: GitHub Repository

### 5.1.6 Amazon Developer Konto

Für das Erstellen von Alexa Skills muss ein Amazon Developer Konto erstellt werden. [AMA18a]

### 5.1.7 Alexa

Zunächst einmal müssen im ersten Schritt unter Skill Information grundlegende Informationen über den Skill festgelegt werden, welche die weiteren Schritte und deren Komplexität beeinflussen.

The screenshot shows the Amazon Developer Console interface for setting up a new skill. The top navigation bar includes 'amazon // DEVELOPER CONSOLE' and user information 'ROBIN WARTH - SHSVSP'. The main menu has tabs for DASHBOARD, APPS & SERVICES, ALEXA (selected), REPORTING, SUPPORT, DOCS, and SETTINGS. The header area shows 'Back to All Skills' and 'Getting Started' links. The skill being configured is 'Raspberry Dude' under the 'Smart Home' category, with ID 'amzn1.ask.skill.920e5e6a-89da-4f0e-9728-69870ca80c88'. The left sidebar lists setup steps: Skill Information (checked), Interaction Model (checked), Configuration (checked), Test (checked), Publishing Information (unchecked), and Privacy & Compliance (unchecked). The main form area includes a language selector set to 'German', a 'Skill Type' dropdown set to 'Smart Home', a 'Language' dropdown set to 'German', an 'Application Id' field with the same ID, a 'Name' field set to 'Raspberry Dude', and a 'Payload Version' section with 'v3 (preferred)' selected. At the bottom right is a yellow 'Next' button.

Abb. 5.1: Setup Skill-Information

Wie schon in [Konzept -> Skilltypen] beschrieben, fiel die Entscheidung auf einen Skill, welcher die Smart Home Skill API unterstützt. Außerdem müssen in diesem Schritt noch die unterstützten Sprachen des Skills sowie ein Skill Name festgelegt werden.

Die Payload Version bestimmt in welcher Form und Stil Nachrichten im JSON-Format zwischen Skill, Alexa Smart Home API und Server ausgetauscht werden sollen. Dies wird im weiteren Verlauf noch einmal erneut aufgegriffen, wenn der Aufbau und Stil der JSON-Nachrichten weiter erläutert wird. [Verweis auf das Kapitel] Auch wenn hier noch zusätzlich die Auswahl einer Payload Version aufgeführt ist, so lässt sich doch nur die Payload Version „v3“ auswählen. Payload Version „v2“ ist zum Zeitpunkt dieser Studienarbeit schon so veraltet, dass man sie nicht einmal auswählen kann. Sie ist nur für ältere schon erstellte Skills noch aufgeführt

The screenshot shows the Alexa Skill Setup interface. At the top, there's a language selector with 'German' selected and a link to 'Add a New Language'. On the left, a sidebar lists the setup steps: 'Skill Information' (checked), 'Interaction Model' (checked and highlighted in orange), 'Configuration' (checked), 'Test' (checked), 'Publishing Information' (unchecked), and 'Privacy & Compliance' (unchecked). The main content area for the 'Interaction Model' step contains an information icon and the text: 'Your skill uses Smart Home Skill API'. Below this, it states: 'The interaction model is already built into the Smart Home Skill API and you can find the supported utterances [here](#). You just have to handle the directives sent by Alexa in your Smart Home adapter (Lambda function)'. At the bottom of the main area are 'Save' and 'Submit for certification' buttons. In the bottom right corner, there is a yellow 'Next' button. A 'Skills Beta Testing' banner at the bottom left indicates the status is 'Not yet eligible'.

Abb. 5.2: Setup Interaction Modell

Da in diesem Skill die Smart Home Skill API zum Einsatz kommt, muss in dem Schritt Interaction Model nichts weiter getan werden. Das Interaktionsmodell ist bereits in die API eingebaut.

Im Schritt Configuration wird zunächst der Service Endpunkt festgelegt, welcher der von Alexa gesendeten Direktiven geeignet verarbeiten werden kann. Da hier die Smart Home Skill API zum Einsatz kommt, wird man von Amazon dazu verpflichtet einen Amazon Web Services (AWS) Lambda Server zu nutzen. Zu dem Aufbau und der Einrichtung des AWS Lambda Server wird in Abschnitt [bla bla keine Ahnung, AWS Lambda] noch gezielt eingegangen.

German ☒ Add a New Language

**Skill Information** ☒  
**Interaction Model** ☒  
**Configuration** ☒  
**Test** ☒  
**Publishing Information** ☐  
**Privacy & Compliance** ☐

**Skills Beta Testing** NEW  
 Status Not yet eligible

**Global Fields**  
 These fields apply to all languages supported by the skill.

**Endpoint**  
 Service Endpoint Type: **AWS Lambda ARN (Amazon Resource Name)**   
*Recommended*  
 Lambda ARN url for Smart Home Adapter.  
[More info about AWS Lambda](#)  
[How to integrate AWS Lambda with Alexa](#)

**Default**  
 arn:aws:lambda:eu-west-1:245784272632:function:myFunctionRaspberry

**Provide geographical region endpoints?**  
 (Optional) ☒ Yes ☐ No

☐ North America ☒ Europe ☐ India ☐ Far East

**Europe**  
 arn:aws:lambda:eu-west-1:245784272632:function:myFunctionRaspberry

**Account Linking**

Abb. 5.3: Setup Configuration Endpoints

Wurde aber statt der Smart Home Skill API das Custom Interaction Model als Skill Typ gewählt, ist es möglich anstatt dem AWS Lambda Server auch jeden beliebigen anderen zertifizierten HTTPS Webserver anzugeben.

Bei Amazon Web Services wie dem Lambda Server (Siehe 5.1.8 Amazon Web Services (AWS) Lambda) wird jedem Server eine eindeutige Adresse zur Ressource und der darin enthaltenen Funktionen zugeteilt. Diese Adresse nennt sich Amazon Resource Name (ARN). Die ARN des Webservices dieser Studienarbeit lautet:

„arn:aws:lambda:eu-west-1:245784272632:function:myFunctionRaspberry“.

In der ARN sind Informationen über den Server Typ (hier: „Lambda“), über den Server Standort (hier: „eu-west“, Irland) und die Funktion (hier: „myFunctionRaspberry“) enthalten, welche den Endpunkt repräsentiert und die Direktiven von Alexa entgegennimmt.

Das Account Linking ist für Skills der Smart Home Skill API zwingend notwendig, aber für Skills mit Custom Interaction Model ist dies optional. Das Account Linking erfolgt mit Hilfe des OAuth2 Authentifizierungsstandard. [PAR18]

## Account Linking

**Authorization URL**  
The url where customers will be redirected in the companion app to enter login credentials.

**Client Id**  
Unique public string used to identify the client requesting for authentication.

**Domain List** (Optional)  
The list of domains that the authorization URL will fetch content from. You can provide up to 30 domains.

**Scope**  
List of permissions to request from the skill user. You can provide up to 15 scopes.

**Redirect URLs** (Optional)  
The list of valid HTTPS redirection endpoints that could be requested during authorization to redirect the user back to after the authorization process.  
[Learn more](#).

**Authorization Grant Type** (Optional)  
Specifies the OAuth authorization grant that Alexa uses to obtain an access token from your provider.  
[Learn more](#).

**Access Token URI**  
This URI will be used for both access token and token refresh requests.

**Client Secret**

**Client Authentication Scheme** (Optional)

**Permissions**

**Request users to access resources and capabilities**  
Please request permissions to resources and capabilities that are absolutely core to the customer experience delivered by the skill.  
[Learn More](#)

☒ **Send Alexa Events** ⓘ

**Alexa Skill Messaging**  
Use clientId and clientSecret for skill messaging, enabling out-of-session (non-customer invoked) events to be sent to skill code.

**Client ID** amzn1.application-oa2-client.bf5d5e600cb941528035489d8d0eabf2  
**Client Secret** \*\*\*\*\* [Show](#)

**Privacy Policy URL**  
Link to the Privacy Policy for this skill. This is mandatory for account linking.

[Save](#) [Submit for certification](#)

[Next](#)

Das Account Linking ist notwendig, um jeden Benutzer des Skills eindeutig identifizieren zu können. Im Allgemeinen wird hier eine Verknüpfung zwischen Alexa-Amazon-Account und dem Account beim Hersteller der Smart Home Geräte, wie z. B. Lampen, geschaffen.

In diesem Project erfolgt das Account Linking mit einem Amazon Account. Die Einstellungen hier sind alle auf Login With Amazon (LWA) getrimmt. [AMA18b] Der Aufwand, um einen guten, sicheren und verlässlichen OAuth2 Server selbst zu erstellen, wäre für diese Studienarbeit zu groß gewesen. Zudem muss der OAuth2 Server auch von Amazon akzeptiert sein und daher wären zusätzliche gültige Zertifikate notwendig. Die Möglichkeiten, welche durch Account Linking geschaffen werden, würden ohnehin nicht ausgereizt werden in dieser Studienarbeit und dem Einsatz Gebiet von dem hier erstellten Alexa Skill. [AMA18c]

...

Um den Skill mit der Alexa App auf dem Smartphone und einem Amazon Echo Gerät zu testen, muss nur der Schalter in Schritt Test umgelegt werden. Der Skill wird nun für die mit dem Konto verbundenen Geräte automatisch zum Testen eingerichtet. Alternativ kann der Skill auch mittels, dem zur Zeit der Studienarbeit in der Beta-Phase befindlichen, Test-Simulator ausgeführt werden. Der Test-Simulator bietet dieselbe Möglichkeit zum Testen wie ein Echo-Gerät und sogar zusätzliche Funktionen. Im Test-Simulator kann direkt mit Alexa über Sprache oder geschriebenem Text interagiert werden. Wurde zu Beginn das Custom Interaction Model statt der Smart Home Skill API gewählt, lassen sich im Test Simulator auch die JSON Direktiven anzeigen, welche von Alexa oder dem Skill-Endpoint gesendet werden.

Abb. 5.5: Setup Test

### 5.1.8 Amazon Web Services (AWS) Lambda

Um einen Alexa Skill mit der Smart Home Skill API zu erstellen, wird ein AWS Lambda Server vorausgesetzt. Hierfür muss ein Konto für AWS erstellt und ein AWS Lambda Server gemietet werden.

Bevor der Lambda Server eingerichtet wird, muss ein Alexa Skill bereits erstellt sein. Bei der Einrichtung des AWS Lambda Server für den Alexa Skill wird die Skill-ID benötigt. Ebenfalls ist ein AWS Account zwingend notwendig.

Nach der Anmeldung bei AWS wird zu Lambda navigiert. Für den Lambda Server ist es wichtig, die richtige Region auszuwählen. Mögliche Regionen für Alexa Skills sind Asia Pacific (Tokyo), EU (Ireland), US East (N. Virginia) und US West (Oregon). [AMA18d]

Als nächstes muss eine Lambda Function werden, um in dieser die von Alexa gesendeten Direktiven zu entgegen zu nehmen und zu verarbeiten.

Bei der Erstellung einer Lambda Function kann man aus verschiedenen Blaupausen oder eine leere Funktion, ohne Codebeispiel, wählen. Für den Einsatz der Lambda Function in dieser Studienarbeit hat zur Zeit der Bearbeitung leider keine passende Blueprint existiert, diese waren veraltet und haben eine alte „Payload-Version“ der Alexa Direktiven unterstützt. Dank Codebeispielen in der Online Dokumentation von Amazon Alexa kann die Lambda Funktion dennoch zügig einsatzbereit gemacht werden. Der Name der hier



erstellten Lambda Funktion lautet „myFunctionRaspberry“.

Danach muss eine Rolle gewählt werden, da zu diesem Zeitpunkt noch keine eigene definierte Rolle existiert. Für dieses Project wurde eine Rolle Namens „myRoleRaspberry“ mithilfe des Templates „Basic Edge Lambda permissions“ erstellt.

Durch das Betätigen des „Create Function“ Buttons wird die Funktion von AWS eingerichtet. Um auf Ereignisse zu reagieren, werden Trigger zur Funktion hinzugefügt. Standardmäßig wurde ein Trigger „Amazon CloudWatch Logs“ bereits der Funktion hinzugefügt. Dieser ist für das Logging zuständig. Jede Log Ausgabe wie z. B. „console.log('test');“ landet somit automatisch in den „Amazon CloudWatch Logs“ mit Datum und eindeutigen Identifikatoren. Um die Lambda Funktion für die Alexa Smart Home Skill API bereit zu machen, wird der Alexa Smart Home Trigger ausgewählt. Dieser Trigger muss nun nur noch mithilfe der Skill-ID bereit gemacht werden.

## 5.2 Entwicklung

Hier wird [folgt]

### 5.2.1 AWS Lambda

[folgt]

#### Export Handler

[insert exports.handler = ... here]

Mit exports.handler wird die Funktion definiert, welche ausgeführt wird, sobald eine Direktive an die Lambda Funktion gesendet wird. In dem Parameter „request“ stecken die Informationen, welche mit der Direktive gesendet werden. Der Parameter „context“ wird von AWS Lambda dazu verwendet Laufzeitinformationen zur ausgeführten Lambda-Funktion bereitzustellen.

Hier im export-handler werden die verschiedenen Typen an Direktiven zunächst einmal anhand von dem „header“ unterschieden und entsprechend weitere Aktionen durchgeführt.

#### Direktiven

Die Direktiven welche von Alexa gesendet werden sind Nachrichten im JSON-Format. Die Direktiven beschreiben Aktionen, welche ausgeführt werden sollen, sowie Informationen zu dem Client welcher die Aktion ausführen möchte.

Beispiele für Direktiven sind:

**Report State** Über die ReportState-Direktive wird der Status der Smart Home Geräte abgefragt. Dies geschieht insbesondere dann, wenn der Benutzer über sein Smartphone in der Alexa App seine Geräte ansieht. In diesem Fall werden in kleinen zeitlichen Abständen kontinuierlich ReportState-Direktiven gesendet.

```
{
  "directive": {
    "header": {
      "namespace": "Alexa",
      "name": "ReportState",
      "payloadVersion": "3",
      "messageId": "e9029fff-5d95-44bb-837b-8032d8c5f07d",
      "correlationToken": "AAAAAAAAAAQCwH+Bi...
        ↪ TAMUZSXeE2sPN6MqRNCVtOlQKG0RY2vxQI="
    },
    "endpoint": {
      "scope": {
        "type": "BearerToken",
        "token": "Atza|IwEBIJqaj14hApPxtKKwPXA...1APp5F-
          ↪ a2cDP_EW7Fgj-A"
      },
      "endpointId": "wirelessSwitch1",
      "cookie": {
        "key1": "arbitrary key/value pairs for skill to reference this endpoint.",
        "key2": "There can be multiple entries",
        "key3": "but they should only be used for reference purposes.",
        "key4": "This is not a suitable place to maintain current endpoint state."
      }
    },
    "payload": {}
  }
}
```

Quellcode 5.3: Request: Report-State

```
{
  "context": {
    "properties": [
      {
        "namespace": "Alexa.PowerController",
        "name": "powerState",
        "value": "OFF",
        "timeOfSample": "2018-03-07T06:58:25.978Z",

```

```

        "uncertaintyInMilliseconds": 500
    }
]
},
"event": {
    "header": {
        "messageId": "e9029fff-5d95-44bb-837b-8032d8c5f07d-R",
        "correlationToken": "AAAAAAAAAAQCwH+Bi...
        ↪ TAMUZSXeE2sPN6MqRNCVtOlQKG0RY2vxQI=",
        "namespace": "Alexa",
        "name": "StateReport",
        "payloadVersion": "3"
    },
    "endpoint": {
        "scope": {
            "type": "BearerToken",
            "token": "Atza|IwEBIJqaj14hApPxtKKwPXA...1APp5F-
            ↪ a2cDP_EW7Fgj-A"
        },
        "endpointId": "wirelessSwitch1",
        "cookie": {}
    },
    "payload": {}
},
"payload": {}
}

```

Quellcode 5.4: Response: Report-State

**Discovery** Über die Discovery-Direktive fragt Alexa nach verfügbaren Smart Home Geräten. Diese Direktive wird gesendet, wenn der Benutzer aktiv in der Alexa App nach Geräten suchen lässt.

```

{
    "directive": {
        "header": {
            "namespace": "Alexa.Discovery",
            "name": "Discover",
            "payloadVersion": "3",
            "messageId": "af4efd28-3d33-4f5e-9d18-ae755a40c8c0"
        },
        "payload": {
            "scope": {

```

```

        "type": "BearerToken",
        "token": "Atza|IwEBIJqaj14hApPxtKKwPXA...1APp5F-
        ↪ a2cDP_EW7Fgj-A"
    }
}
}
}

```

Quellcode 5.5: Request: Discovery

```

{
  "header": {
    "namespace": "Alexa.Discovery",
    "name": "Discover.Response",
    "payloadVersion": "3",
    "messageId": "af4efd28-3d33-4f5e-9d18-ae755a40c8c0"
  },
  "payload": {
    "endpoints": [
      {
        "endpointId": "wirelessSwitch1",
        "manufacturerName": "Raspberry Dude",
        "friendlyName": "Licht",
        "description": "Smart Device Switch",
        "displayCategories": [
          "LIGHT"
        ],
        "cookie": {
          "key1": "arbitrary key/value pairs for skill to reference this endpoint
          ↪ .",
          "key2": "There can be multiple entries",
          "key3": "but they should only be used for reference purposes.",
          "key4": "This is not a suitable place to maintain current endpoint
          ↪ state."
        },
        "capabilities": [
          {
            "type": "AlexaInterface",
            "interface": "Alexa",
            "version": "3"
          },
          {
            "interface": "Alexa.PowerController",

```

```
        "version": "3",
        "type": "AlexaInterface",
        "properties": {
            "supported": [
                {
                    "name": "powerState"
                }
            ],
            "retrievable": true
        }
    }
]
},
{
    "endpointId": "wirelessSwitch2",
    "manufacturerName": "Raspberry Dude",
    "friendlyName": "PC-Switch",
    "description": "Smart Device Switch",
    "displayCategories": [
        "SWITCH"
    ],
    "cookie": {},
    "capabilities": [
        {
            "type": "AlexaInterface",
            "interface": "Alexa",
            "version": "3"
        },
        {
            "interface": "Alexa.PowerController",
            "version": "3",
            "type": "AlexaInterface",
            "properties": {
                "supported": [
                    {
                        "name": "powerState"
                    }
                ],
                "retrievable": true
            }
        }
    ]
}
```

```

    },
    {
      "endpointId": "wirelessSwitch3",
      "manufacturerName": "Raspberry Dude",
      "friendlyName": "TV-Switch",
      "description": "Smart Device Switch",
      "displayCategories": [
        "SWITCH"
      ],
      "cookie": {},
      "capabilities": [
        {
          "type": "AlexaInterface",
          "interface": "Alexa",
          "version": "3"
        },
        {
          "interface": "Alexa.PowerController",
          "version": "3",
          "type": "AlexaInterface",
          "properties": {
            "supported": [
              {
                "name": "powerState"
              }
            ],
            "retrievable": true
          }
        }
      ]
    }
  ]
}

```

Quellcode 5.6: Response: Discovery

**Power Controller** Über den PowerController wird versucht, aktiv einen Gerätestatus zu verändern, wenn z. B. wenn der Benutzer sein Licht einschaltet oder die Heizung aufdrehen möchte.

```

{
  "directive": {

```

```

    "header": {
      "namespace": "Alexa.PowerController",
      "name": "TurnOn",
      "payloadVersion": "3",
      "messageId": "64da8006-d724-4861-a929-d560cbf2b92b",
      "correlationToken": "AAAAAAAAAAQCwH+BiiholKtQip3as2gbcBAIAAAA
        ↪ ...BdP/jF6aJ9bc="
    },
    "endpoint": {
      "scope": {
        "type": "BearerToken",
        "token": "Atza|IwEBIJqaj14hApPxtKKwPXA...1APp5F-
          ↪ a2cDP__EW7Fgj-A"
      },
      "endpointId": "wirelessSwitch1",
      "cookie": {
        "key1": "arbitrary key/value pairs for skill to reference this endpoint.",
        "key2": "There can be multiple entries",
        "key3": "but they should only be used for reference purposes.",
        "key4": "This is not a suitable place to maintain current endpoint state."
      }
    },
    "payload": {}
  }
}

```

Quellcode 5.7: Request: Power-Controller

```

{
  "context": {
    "properties": [
      {
        "namespace": "Alexa.PowerController",
        "name": "powerState",
        "value": "ON",
        "timeOfSample": "2018-03-07T12:34:45.360Z",
        "uncertaintyInMilliseconds": 500
      }
    ]
  },
  "event": {
    "header": {
      "messageId": "64da8006-d724-4861-a929-d560cbf2b92b-R",

```

```
    "correlationToken": "AAAAAAAAAAQCwH+BiiholKtQip3as2gbcBAIAAAA
        ↪ ...BdP/jF6aJ9bc=",
    "namespace": "Alexa",
    "name": "Response",
    "payloadVersion": "3"
  },
  "endpoint": {
    "scope": {
      "type": "BearerToken",
      "token": "Atza|IwEBIJqaj14hApPxtKKwPXA...1APp5F–
        ↪ a2cDP_EW7Fgj–A"
    },
    "endpointId": "wirelessSwitch1",
    "cookie": {}
  },
  "payload": {}
},
"payload": {}
}
```

Quellcode 5.8: Response: Power-Controller



## 6 Ergebnis, Fazit und Ausblick

[folgt]

# Literatur

- [ALEXA] AMAZON.COM. *Amazon Alexa Logo*. abgerufen am 01.12.2017 15:23 Uhr. 2017. URL: [https://www.wink.com/img/vendor/logo\\_amazon.svg](https://www.wink.com/img/vendor/logo_amazon.svg) [siehe S. 17].
- [AMA18a] AMAZON.COM. *Amazon Developer*. abgerufen am 27.02.2018 12:23 Uhr. 2018. URL: <https://developer.amazon.com/de/> [siehe S. 19].
- [AMA18b] AMAZON.COM. *Login with Amazon for Websites*. abgerufen am 27.02.2018 12:33 Uhr. 2018. URL: <https://developer.amazon.com/de/docs/login-with-amazon/web-docs.html> [siehe S. 24].
- [AMA18c] AMAZON.COM. *Authenticate an Alexa User to a User in Your System with Account Linking*. abgerufen am 27.02.2018 12:33 Uhr. 2018. URL: <https://developer.amazon.com/de/docs/smarthome/authenticate-an-alexa-user-account-linking.html> [siehe S. 24].
- [AMA18d] AMAZON.COM. *Host a Custom Skill as an AWS Lambda Function*. abgerufen am 27.02.2018 12:40 Uhr. 2018. URL: <https://developer.amazon.com/de/docs/custom-skills/host-a-custom-skill-as-an-aws-lambda-function.html> [siehe S. 25].
- [BAG17] J. BAGER. „Assistent allgegenwärtig - Digitale Assistenten: vom Spielzeug für Nerds zur Bedienoberfläche für alles“. In: *c't 22/2017* [2017]. Online unter <https://www.heise.de/ct/ausgabe/2017-22-Digitale-Assistenten-vom-Spielzeug-fuer-Nerds-zur-Bedienoberflaeche-fuer-alles-3854034.html> - abgerufen am 06.02.201 12:14 Uhr, S. 64–69 [siehe S. 5–7, 9].
- [BLE17] Reche M. BLEICH H. „Freundlich, hölzern, clever und arrogant - Vier Sprachassistenten in ihrem natürlichen Umfeld“. In: *c't 22/2017* [2017]. Online unter <https://www.heise.de/ct/ausgabe/2017-22-Vier-Sprachassistenten-in-ihrem-natuerlichen-Umfeld-3853597.html> - abgerufen am 13.11.2017 10:58 Uhr, S. 80–85 [siehe S. 5–9].
- [C9.io] C9.IO. *C9.io Github Repository*. abgerufen am 27.02.2018 11:57 Uhr. 2018. URL: <https://github.com/c9> [siehe S. 19].
- [CLA18] G. CLAUSER. *Amazon Echo vs. Google Home: Which Voice Controlled Speaker Is Best for You?* abgerufen am 19.02.2018 11:12 Uhr. 2018. URL: <https://thewirecutter.com/reviews/amazon-echo-vs-google-home/> [siehe S. 6, 8].

- [DEV17] DEVOLO AG. *Definition Smart Home*. abgerufen am 17.11.2017 14:22 Uhr. 2017. URL: <https://www.devollo.de/smart-home/> [siehe S. 3].
- [DIS17] F. DISSELHOFF. *Google Home oder Amazon Echo? Der Vergleichstest [mit Video]*. abgerufen am 09.11.2017 12:14 Uhr. 2017. URL: <https://curved.de/reviews/google-home-oder-amazon-echo-der-vergleichstest-mit-video-516362>.
- [DON14] W. DONAT. *Learn Raspberry Pi - Programming with Python. learn to program on the world's most popular tiny computer*. USA: apress, 2014. ISBN: 978-1-4302-6425-5.
- [ECHO] Amazon ECHO. *Amazon Echo Dot 2*. abgerufen am 01.12.2017 15:24 Uhr. 2017. URL: [https://images-eu.ssl-images-amazon.com/images/I/41iz5Tw82IL.\\_SY300\\_QL70\\_.jpg](https://images-eu.ssl-images-amazon.com/images/I/41iz5Tw82IL._SY300_QL70_.jpg) [siehe S. 16].
- [ETC17] RESIN.IO. *Etcher Download*. abgerufen am 27.10.2018 11:35 Uhr. 2018. URL: <https://etcher.io/> [siehe S. 18].
- [HAR17] D. HARDAWAR. *One week with Samsung Bixby - It has potential, but can't yet compete with Apple, Google or Amazon*. abgerufen am 09.02.2018 13:17 Uhr. 2017. URL: <https://www.engadget.com/2017/08/23/samsung-bixby/> [siehe S. 10].
- [HEI12] H. HEIKO. *elropi.py - Remote switch Elro using Python on Raspberry PI*. abgerufen am 10.11.2017 14:01 Uhr. 2012. URL: <https://pastebin.com/aRipYrZ6>.
- [HET17] M. L. HETLAND. *Beginning Python. From Novice to Professional*. 3. Aufl. Trondheim, Norwegen: apress, 2017. ISBN: 978-1-4842-0028-5.
- [ISO06] *ISO 9241: Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*. Standard. 2006 [siehe S. 12].
- [LUN17] J. LUNSFORD. *Setup JavaScript IDE on Pi 3*. abgerufen am 27.02.2018 12:03 Uhr. 2017. URL: <https://dev.to/jtlunsford/setup-javascript-ide-on-pi-3> [siehe S. 19].
- [MER16] N. MERCIER. *Run Python scripts from Node.js with simple (but efficient) inter-process communication through stdio*. abgerufen am 14.11.2017 10:54 Uhr. 2016. URL: <https://github.com/extrabacon/python-shell>.
- [MOZ17] H. BRAUN. *Mozilla Common Voice: Sprachsteuerung für alle und ohne Rückgriff auf die Cloud*. abgerufen am 06.12.2017 14:01 Uhr. 2017. URL: <https://www.heise.de/newsticker/meldung/Mozilla-Common-Voice->

- Sprachsteuerung-fuer-alle-und-ohne-Rueckgriff-auf-die-Cloud-3904454.html [siehe S. 4].
- [NJS0J] NODE.JS *Foundation*. *Node.js Tutorial*. abgerufen am 14.11.2017 10:02 Uhr. o.J. URL: <https://nodejs.org/en/docs/>.
- [NODEa] NODE.JS *Foundation*. *Node.js Logo*. abgerufen am 01.12.2017 15:25 Uhr. 2017. URL: <https://nodejs.org/static/images/logos/nodejs-new-pantone-black.png> [siehe S. 17].
- [NODEb] NODE.JS *Foundation*. *Installing Node.js via package manager*. abgerufen am 27.02.2018 11:50 Uhr. 2018. URL: <https://nodejs.org/en/download/package-manager/> [siehe S. 19].
- [PAR18] NODE.JS *Foundation*. *OAuth 2.0*. abgerufen am 27.02.2018 12:27 Uhr. 2018. URL: <https://oauth.net/2/> [siehe S. 22].
- [PFI17] Kaufmann T. PFISTER B. *Sprachverarbeitung. Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. 2. Aufl. Zürich, Schweiz: Springer Vieweg, 2017. ISBN: 978-3-662-52838-9 [siehe S. 4].
- [PSF17] Python Software FOUNDATION. *Python 3.6.3 documentation*. abgerufen am 14.11.2017 18:07 Uhr. 2017. URL: <https://docs.python.org/3/>.
- [PuTTY] S. TATHAM. *Download PuTTY: latest release (0.70)*. abgerufen am 27.02.2018 11:40 Uhr. 2017. URL: <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> [siehe S. 18].
- [RAO17] Manjunath K.E. RAO S. K. *Speech Recognition Using Articulatory and Excitation Source Feature. Speech Technology*. Indien: Springer, 2017. ISBN: 978-1-4842-2922-4 [siehe S. 3].
- [RASPa] Raspberry Pi FOUNDATION. *Raspberry Pi Logo*. abgerufen am 01.12.2017 15:26 Uhr. 2017. URL: <https://www.raspberrypi.org/app/uploads/2011/10/Raspi-PGB001.png> [siehe S. 17].
- [RASPb] Raspberry Pi FOUNDATION. *Download Raspbian for Raspberry Pi*. abgerufen am 26.02.2018 11:29 Uhr. 2018. URL: <https://www.raspberrypi.org/downloads/raspbian/> [siehe S. 18].
- [REI17] A. REINHARDT. *Samsung Bixby Voice im Alltags-Check*. abgerufen am 09.02.2018 13:18 Uhr. 2017. URL: <https://www.teltarif.de/samsung-bixby-sprachassistent-erfahrungsbericht/news/70083.html> [siehe S. 10].
- [ROT17] K. ROTHER. *Pro Python - Best Practices. Debugging, Testing and Maintenance*. Berlin, Deutschland: apress, 2017. ISBN: 978-1-4842-2241-6.

- [ROU17a] M. ROUSE. *Amazon Developer Services*. abgerufen am 09.01.2018 09:57 Uhr. 2017. URL: <http://searchaws.techtarget.com/definition/Amazon-Developer-Services> [siehe S. 12].
- [ROU17b] M. ROUSE. *Alexa Skills Kit*. abgerufen am 09.01.2018 10:31 Uhr. 2017. URL: <http://searchaws.techtarget.com/definition/Alexa-Skills-Kit> [siehe S. 12].
- [RVNC] RealVNC LIMITED. *Remotezugriffssoftware für Desktop-Computer und Mobilgeräte / RealVNC*. abgerufen am 27.02.2018 11:46 Uhr. 2018. URL: <https://www.realvnc.com/de/> [siehe S. 18].
- [SCH17] K. SCHILLER. *Was ist ein Smart Home? Geräte und Systeme*. abgerufen am 10.11.2017 12:20 Uhr. 2017. URL: <https://www.homeandsmart.de/was-ist-ein-smart-home> [siehe S. 3].
- [SLIoJ] Inc STRONGLOOP. *Express - Schnelles, offenes, unkompliziertes Web-Framework für Node.js*. abgerufen am 14.11.2017 10:05 Uhr. o.J. URL: <http://expressjs.com/de/>.
- [SOF16] o.V. *How to call Python function from NodeJS*. abgerufen am 14.11.2017 19:27 Uhr. 2016. URL: <https://stackoverflow.com/questions/23450534/how-to-call-python-function-from-nodejs>.
- [SOP17] M. E. SOPER. *Expanding Your Raspberry Pi. Storage, printing, peripherals, and network connections for your Raspberry Pi*. Indianapolis, USA: apress, 2017. ISBN: 978-1-4842-2922-4.
- [VOE16] F. VÖLKELE. *Die Historie des Smart Home von 1963 - Heute*. abgerufen am 02.02.2018 10:53 Uhr. 2016. URL: [https://www.smartest-home.com/blog/smart\\_home\\_historie\\_1939\\_bis\\_2017\\_frank\\_voelkel/](https://www.smartest-home.com/blog/smart_home_historie_1939_bis_2017_frank_voelkel/) [siehe S. 1].
- [W3SCHa] o.V. *Node.js and Raspberry Pi*. abgerufen am 14.11.2017 10:00 Uhr. o.J. a. URL: [https://www.w3schools.com/nodejs/nodejs\\_raspberrypi.asp](https://www.w3schools.com/nodejs/nodejs_raspberrypi.asp) [siehe S. 18].
- [W3SCHb] o.V. *Node.js Tutorial*. abgerufen am 14.11.2017 10:01 Uhr. o.J. b. URL: <https://www.w3schools.com/nodejs/>.
- [WIG17] K. WIGGERS. *How to use the brand-new Samsung Bixby voice assistant, and everything it can do - Bixby 2.0 brings improved speech recognition and third-party app support*. abgerufen am 15.02.2017 15:23 Uhr. 2017. URL: <https://www.digitaltrends.com/mobile/samsung-bixby-how-to-use/> [siehe S. 10].

# Anhang

dsfdgfs