



FOM HOCHSCHULE FÜR OEKONOMIE & MANAGEMENT

UNIVERSITY OF APPLIED SCIENCES

HOCHSCHULZENTRUM KÖLN

BACHELOR-THESIS

IM STUDIENGANG WIRTSCHAFTSINFORMATIK

ZUR ERLANGUNG DES GRADES EINES

BACHELOR OF SCIENCE (B.Sc.)

---

**Evaluierung des Einsatzes von ChatOps zur  
Arbeitserleichterung der Mitarbeiter im  
Rechenzentrum beim Umgang mit einer  
Configuration-Management-Database**

---

*Autor:*

Kevin WEISS

Matrikel-Nr. 415171

*Betreuer:*

Stiven RASO M.A.

Köln, 4. Juli 2019

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>Abbildungsverzeichnis</b>	<b>VI</b>
<b>Tabellenverzeichnis</b>	<b>VII</b>
<b>Listings</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Umfeld . . . . .	2
1.2 Ziel der Arbeit . . . . .	2
1.3 Abgrenzung . . . . .	2
1.4 Vorgehensweise . . . . .	2
<b>2 Grundlagen</b>	<b>4</b>
2.1 Chatbots . . . . .	4
2.1.1 Komponenten . . . . .	4
2.1.2 Arten . . . . .	5
2.1.3 Sonderfall Sprachassistenten . . . . .	5
2.1.4 Charakter . . . . .	5
2.2 ChatOps . . . . .	6
2.2.1 Grundgedanke . . . . .	6
2.2.2 Komponenten . . . . .	6
2.3 CMDB . . . . .	9
2.4 Python . . . . .	12
2.5 Application Programming Interface . . . . .	12
<b>3 Anforderungsanalyse anhand von Experteninterviews</b>	<b>13</b>
3.1 Aktuelle Situation . . . . .	14
3.2 Expertenauswahl . . . . .	16

3.2.1	Kriterien . . . . .	16
3.2.2	Vorstellung . . . . .	18
3.3	Ermittlung der Anforderungen an einen CMDB Chatbot . . . . .	18
3.3.1	Leitfadenerstellung . . . . .	19
3.3.2	Tonaufnahme . . . . .	19
3.4	Dokumentation der Anforderungen . . . . .	19
3.4.1	Transkription . . . . .	20
3.4.2	Paraphrase . . . . .	20
3.4.3	(Bildung von) Überschriften . . . . .	27
3.5	Validierung der Anforderungen . . . . .	32
3.5.1	Qualitätsanforderungen . . . . .	33
3.5.2	Funktionale Anforderungen . . . . .	35
<b>4</b>	<b>Entwurf eines Chatbots für eine CMDB</b>	<b>36</b>
4.1	Qualitätsanforderungen . . . . .	36
4.1.1	Ansatz, Bot-Framework, Plattform-Agnostik, Schnittstellenanbindung . . . . .	36
4.1.2	Berechtigungskonzept . . . . .	37
4.1.3	Reaktivität . . . . .	38
4.1.4	API-Einsatz . . . . .	38
4.1.5	Formatierung des Outputs . . . . .	39
4.1.6	Charakter . . . . .	39
4.1.7	Avatar . . . . .	40
4.1.8	Nachvollziehbarkeit . . . . .	40
4.1.9	Vermeidung Fehlbedienung . . . . .	40
4.1.10	Herstellerkonsolidierung . . . . .	40
4.1.11	Arbeitsüberwachung . . . . .	41
4.1.12	Webinterface-Eingaben . . . . .	41
4.1.13	Hilfe . . . . .	41
4.1.14	Vorhersage . . . . .	41
4.1.15	Standardanfragen . . . . .	41
4.2	Funktionale Anforderungen . . . . .	41
4.2.1	Suchfunktion . . . . .	41
4.2.2	Nächste freie IP . . . . .	42
4.2.3	Serverplanung . . . . .	42
4.2.4	Standortabfrage, IP-Adresse, Statusabfrage, Softwarestand, Netzinformationen, Kontaktperson . . . . .	43

4.2.5	Bestandsprüfung, Standortreport, Ressourcenübersicht . . . . .	43
4.2.6	Ressourcen . . . . .	43
4.2.7	Statuswechsel . . . . .	44
4.2.8	Logbuch . . . . .	44
<b>5</b>	<b>Fazit</b>	<b>45</b>
5.1	Zusammenfassung und Schlussfolgerung . . . . .	45
5.2	Ausblick . . . . .	46
<b>6</b>	<b>Quellenverzeichnis</b>	<b>X</b>
	<b>Anhang</b>	<b>XV</b>
<b>A</b>	<b>Leitfaden für die Experteninterviews</b>	<b>XV</b>
<b>B</b>	<b>Experteninterview Hemmerich</b>	<b>XVII</b>
<b>C</b>	<b>Experteninterview Middendorf</b>	<b>XXI</b>
<b>D</b>	<b>Experteninterview Kirsten</b>	<b>XXVI</b>
<b>E</b>	<b>Experteninterview Buck</b>	<b>XXX</b>

# Abkürzungsverzeichnis

<b>API</b>	Application Programming Interface
<b>ARD</b>	Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland
<b>CAMS</b>	Culture of Automation, Measurement and Sharing
<b>CI</b>	Configuration Item
<b>CMDB</b>	Configuration Management DataBase
<b>CMS</b>	Configuration Management System
<b>DNS</b>	Domain Name System
<b>DW</b>	Deutsche Welle
<b>IaC</b>	Infrastructure as Code
<b>ITIL</b>	IT Infrastructure Library
<b>IVZ</b>	Informations-Verarbeitungs-Zentrum
<b>KI</b>	Künstliche Intelligenz
<b>LAN</b>	Local Area Network
<b>MIT</b>	Massachusetts Institute of Technology
<b>MPC</b>	Music Player Client
<b>MPD</b>	Music Player Daemon
<b>rbb</b>	Rundfunk Berlin-Brandenburg
<b>SAN</b>	Storage Area Network
<b>SLA</b>	Service Level Agreement
<b>Slack</b>	Searchable Log of All Conversation and Knowledge
<b>WDR</b>	Westdeutscher Rundfunk

# Abbildungsverzeichnis

1.1	Bewertung von Chatbots . . . . .	1
2.1	ChatOps Prozess . . . . .	7
2.2	Configuration Item Beziehungen . . . . .	9
3.1	i-doit IP Management . . . . .	15
3.2	Experteneinordnung . . . . .	17

# Tabellenverzeichnis

2.1	Verschiedene ChatOps Bots . . . . .	8
2.2	ChatOps Systemintegrationen . . . . .	8
2.3	Beziehungen zwischen Configuration Items . . . . .	10
2.4	Status von Configuration Items . . . . .	11
3.1	Arten von Experteninterviews . . . . .	14
3.2	Experteninterview und Anforderungsmanagement im Vergleich . . . . .	14
3.3	Anforderungen Experteninterview Hemmerich . . . . .	29
3.4	Anforderungen Experteninterview Middendorf . . . . .	30
3.5	Anforderungen Experteninterview Kirsten . . . . .	31
3.6	Anforderungen Experteninterview Buck . . . . .	31

# Listings

2.1	Beispiel requirements.txt . . . . .	12
4.1	Virtualenv Einrichtung und Errbot Installation . . . . .	36
4.2	Struktur eines Errbot Projektes . . . . .	37
4.3	Errbot Access controls . . . . .	38
4.4	Funktionen zur Formatierung . . . . .	39
4.5	Hinzufügen eines Logbuch Eintrages . . . . .	40
4.6	Nachfrage bei der Archivierung von Objekten . . . . .	40
4.7	Suchfunktion . . . . .	42
4.8	Nächste freie IP . . . . .	42
4.9	Serverplanung mit i-doit-cli . . . . .	42
4.10	Detailansicht eines Objekts mit i-doit-cli . . . . .	43
4.11	Ressourcenauslastung eines Geräts . . . . .	43
4.12	Statuswechsel eines Objekts . . . . .	44
4.13	Logbucheintag . . . . .	44

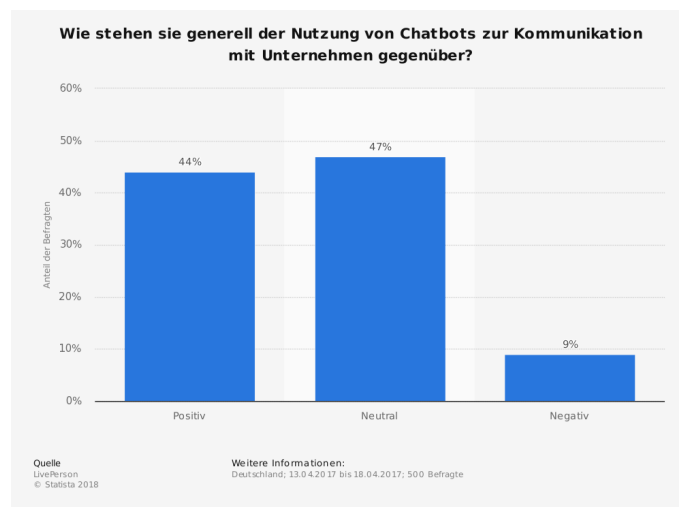


# Kapitel 1

## Einleitung

Bereits 1966 hat Joseph Weizenbaum mit ELIZA am Massachusetts Institute of Technology (MIT) ein Programm geschaffen, das Strukturen in menschlicher Sprache erkennen, verstehen und auf diese antworten kann, den ersten Chatbot sozusagen.<sup>1</sup>

Auch in der heutigen Gesellschaft ist diese Funktionalität gefragt denn je. Besonders auf Unternehmenswebsites oder Online-Shops sind immer häufiger Chatbots anzutreffen und die Kunden stehen dem, wie in Abbildung 1.1 zu sehen ist vermehrt neutral bis positiv gegenüber.<sup>2</sup>



Quelle: LivePerson (2019), o. S.

Abbildung 1.1: Bewertung von Chatbots

Auch in Privathaushalten halten Chatbots in Form von Sprachassistenten wie *Amazon Alexa* oder *Google Home* Einzug und an jedem aktuellen Windows-Computer kann mittlerweile *Microsoft Cortana* genutzt werden. Im Rahmen der Heimautomatisierung können so Heizung oder Licht per Sprachinteraktion mit Sprachassistenten bzw. Chatbots gesteuert werden.<sup>3</sup>

Da ist es naheliegend, dass man diese Technologie auch im Bereich der Softwareentwicklung oder des Rechenzentrumsbetriebs zur Automatisierung verwenden möchte. GitHub hat dies 2013 erkannt und mit

<sup>1</sup>Vgl. Weizenbaum (1966), o. S.

<sup>2</sup>Vgl. Grötz (2018), S. 50.

<sup>3</sup>Vgl. Kuhn (2015), o. S.

der Vorstellung des Chatbots *Hubot* den Begriff ChatOps, also die Integration eines Chatbots in eine Toolchain oder einen Prozess, geprägt.<sup>4</sup>

Auch das bekannte, an Unternehmenskunden gerichtete Chat-Tool Searchable Log of All Conversation and Knowledge (Slack) bietet die Möglichkeit, Chatbots zu entwickeln und somit an die eigene Arbeitsumgebung anzupassen.<sup>5</sup>

## 1.1 Umfeld

Der Autor ist beim Informations-Verarbeitungs-Zentrum (IVZ) beschäftigt, einer Kooperation aller Rundfunkanstalten der Arbeitsgemeinschaft der öffentlich-rechtlichen Rundfunkanstalten der Bundesrepublik Deutschland (ARD), sowie Deutschlandradio und Deutsche Welle (DW) mit Hauptsitz im Rundfunk Berlin-Brandenburg (rbb). Es wurde 1993 als öffentlich-rechtliche Verwaltungsgemeinschaft gegründet und beschäftigt etwa 200 Mitarbeiter an mehreren Standorten.<sup>6</sup>

Das IVZ bietet seinen Kunden zentral, ARD-weite Rechenzentrumsdienstleistungen, wie z. B. SAP oder Archive (Video, Audio, Text) an. Unternehmensweit wird ein Chat-Tool zur Teamkommunikation eingesetzt.

## 1.2 Ziel der Arbeit

Ziel der vorliegenden Arbeit ist es, zu evaluieren, ob die Einführung einer ChatOps Schnittstelle zur Configuration Management DataBase (CMDB) die Arbeit der Mitarbeiter im Rechenzentrumsbetrieb beim Umgang mit dieser erleichtert und angenehmer macht. Bestimmte Aktionen der CMDB können in diesem Szenario aus einem Chat-Tool heraus aufgerufen werden und bieten eine Alternative zum Standardweg per Website oder Client. Durch die Befragung von Experten in diesem Gebiet soll beantwortet werden, ob der Einsatz von ChatOps in diesem Zusammenhang nutzenbringend ist.

## 1.3 Abgrenzung

Nicht Bestandteil dieser Arbeit ist die Evaluierung von ChatOps im Allgemeinen, sowie der Nutzen von ChatOps im gesamten Tätigkeitsspektrum des Rechenzentrumsbetriebs. Der Fokus liegt lediglich auf der Verwendung der CMDB. Außerdem soll nicht der gesamte Funktionsumfang einer CMDB betrachtet werden, sondern nur die Komponenten und Funktionen, die im Tagesgeschäft des Rechenzentrumsbetriebs häufig verwendet werden.

## 1.4 Vorgehensweise

In Kapitel 2 werden die für das Verständnis der Thematik notwendigen Fachbegriffe eingeführt, erläutert und in einen Zusammenhang gesetzt. In Kapitel 3 wird eine Anforderungsanalyse durchgeführt, um die im Arbeitsalltag am meisten genutzten CMDB Funktionen zu ermitteln und herauszufinden, in welchen Fällen eine ChatOps Umsetzung möglich und sinnvoll ist. Die Anforderungen werden mithilfe von Experteninterviews ermittelt, wobei die Analyse der Transkripte nach der Methode von Meuser und Nagel

---

<sup>4</sup>Vgl. Newland (2013), o. S.

<sup>5</sup>Vgl. Költzsch (2019), o. S.

<sup>6</sup>Vgl. ARD.de (2018), o. S.

erfolgt.

In Kapitel 4 wird ein Chatbot entworfen und die Interaktion mit der bestehenden CMDB beschrieben. Abschließend wird in Kapitel 5 ein Fazit gezogen, es wird aufgrund der in dieser Arbeit gewonnenen Erkenntnisse geschlussfolgert, welchen Nutzen und Vorteil ChatOps im betrachteten Kontext bietet. Darüber hinaus wird ein Ausblick gegeben, wie ChatOps sich in Zukunft entwickeln könnte und in welchen Gebieten (im Rechenzentrum und darüber hinaus) es ebenfalls Fuß fassen könnte.

Diese Arbeit ist in  $\LaTeX$  gesetzt. Die Versionierung wurde mit Git vorgenommen.

## Kapitel 2

# Grundlagen

### 2.1 Chatbots

Ein Chatbot ist ein Programm, das für den Dialog mit Menschen erdacht ist und natürliche Sprache als Text oder Audio entgegennehmen und verarbeiten kann. Chatbots werden hauptsächlich auf Websites und Instant-Messaging Systemen eingesetzt. Benutzereingaben werden angenommen und passende Antworten dazu ausgegeben. Außerdem sollen Chatbots Gäste begrüßen und die Unterhaltung in Gang halten.

Chatbots können rein regelbasiert oder Künstliche Intelligenz (KI)-gesteuert reagieren.

Durch die inherente Automatisierung sind Chatbots relevant bezüglich der „Unterstützung und Ersetzung von Arbeitskräften“<sup>7</sup>

Chatbots können mit realen Personen oder anderen Chatbots kommunizieren. Ein Chatbot ist jeweils für einen bestimmten Aufgabenbereich vorgesehen und deckt nicht das gesamte Spektrum der menschlichen Kommunikation ab. Ähnlich wie bei Avataren können Chatbots eine „körperliche Erscheinung“<sup>8</sup> und „virtuelle Identität“<sup>9</sup> bzw. Charakter haben.

Im Gegensatz zu Agenten arbeiten Chatbots nicht autonom, sondern reaktiv. Sie sind also auf die Befehle von Menschen angewiesen.<sup>10</sup>

Entscheidend für die Gesprächskompetenz von Chatbots ist das zugrundeliegende Wissen. Die eingesetzten Algorithmen spielen nur eine nachgelagerte Rolle.

Für komplexere Problemstellungen kann es nötig werden, den Kontext genauer zu erfassen. Dazu kann die Gesprächshistorie berücksichtigt und Laufzeitvariablen pro Nutzer gespeichert werden.<sup>11</sup>

#### 2.1.1 Komponenten

Die wesentlichen Komponenten eines Chatbots sind Utterances (Äußerungen), Intents (Absichten) und Entities (Entitäten).

Als Äußerungen gelten alle Eingaben des Nutzers in ihrer Basisausprägung, z. B. *Wie wird das Wetter?*. Aus

---

<sup>7</sup>Bendel (2018), o. S.; Vgl. Bendel (2018), o. S.

<sup>8</sup>Vries und Möbus (2006), S. 71 f.

<sup>9</sup>Ebd., S. 71 f.

<sup>10</sup>Vgl. ebd., S. 69 ff.

<sup>11</sup>Vgl. Feindt (2006), S. 82 f.

der Äußerung wird eine Absicht interpretiert, in diesem Beispiel eine Funktion, die das aktuelle Wetter anzeigt. Entitäten sind Zusatzparameter für Äußerungen und werden der, mit der Absicht verknüpften Funktion übergeben. Z. B. *Wie wird das Wetter morgen?*, wobei *morgen* die Entität ist.<sup>12</sup>

## 2.1.2 Arten

Chatbots werden grundsätzlich als regelbasiert oder selbstlernend kategorisiert.

### 2.1.2.1 Regelbasiert

Regelbasierte Chatbots verfügen über einen festen Regelbaum und können nur vordefinierte Fragen beantworten und auf bestimmte Schlüsselwörter reagieren.

Wenn der Nutzer sich vertippt, die Fragen leicht anders formuliert oder nicht die richtigen Schlüsselwörter verwendet, kann der Bot daraus keine Aktion ableiten.<sup>13</sup>

### 2.1.2.2 Selbstlernend

Durch den Einsatz von KI und maschinellem Lernen können Chatbots menschliche Sprache besser verstehen und es muss nicht jede Formulierung eines Satzes angelernt werden. Die Komplexität ist bei dieser Variante höher.<sup>14</sup>

Allerdings können dann Richtigkeit und Qualität der Antworten nicht mehr sichergestellt werden, bzw. die Absicht des Benutzers wird möglicherweise falsch interpretiert.<sup>15</sup>

## 2.1.3 Sonderfall Sprachassistenten

Als Sonderform der Chatbots können Sprachassistenten betrachtet werden, die sich spätestens seit der Einführung von Amazon Alexa, Google Assistant und Apple Siri großer Beliebtheit erfreuen.<sup>16</sup>

Die Steuerbefehle werden dabei nicht textuell, sondern verbal übergeben und müssen in der Regel durch ein Schlüsselwort wie *Okay Google* oder *Alexa* eingeleitet werden.

Dank der Auslagerung der Spracherkennung in die Cloud hat sich die Qualität dieser erheblich verbessert. Das Anlernen von Klangfarbe und Fachvokabular muss nicht mehr individuell manuell vorgenommen werden, sondern geschieht aggregiert in der Cloud.

Als zusätzliche Herausforderungen im Vergleich zur textuellen Variante existieren bei Sprachassistenten „Dialekte, Sprachfehler und Umgebungsgeräusche“<sup>17</sup>

## 2.1.4 Charakter

Wichtig für die Akzeptanz eines Chatbots ist, ihm einen Charakter zu verleihen. Es sollte früh entschieden werden, ob er sachlich, empathisch oder frech sein soll.

Zu den Eigenschaften zählen Alter, Geschlecht, Erscheinung (Mensch, Tier, Maschine), Zweck und Geschichte bzw. Herkunft.

---

<sup>12</sup>Vgl. Grötz (2018), S. 51.

<sup>13</sup>Vgl. ebd., S. 51 f.

<sup>14</sup>Vgl. ebd., S. 51 f.

<sup>15</sup>Vgl. Feindt, Janßen und Sölbrandt (2006), S. 151 f.

<sup>16</sup>Vgl. Grötz (2018), S. 50.

<sup>17</sup>Puscher (2018), S. 64; Vgl. Puscher (2018), S. 64.

Merkmale wie Humor, Offenheit, Redseligkeit, Auftreten (laut, leise, frech) und Ähnlichkeit mit bekannten Figuren (z. B. Comicfiguren) sollten auch geplant werden.

Eine Möglichkeit ist, das Firmenmaskottchen als Chatbot umzusetzen, allerdings sollte abgewägt werden, ob es die nötige Kompetenz verkörpert.<sup>18</sup>

## 2.2 ChatOps

Der Begriff ChatOps stammt aus dem DevOps-Umfeld und wurde von GitHub durch die Einführung ihres Chatbots *Hubot* im Jahr 2013 geprägt. Jesse Newland, der damals den Vortrag hielt spricht von „placing tools directly in the middle of the conversation“<sup>19</sup>, also davon, die im Unternehmen genutzten Tools direkt im Chatroom verfügbar zu machen.<sup>20</sup>

### 2.2.1 Grundgedanke

Unter ChatOps wird die Verbindung von Chatbots mit operativen Prozessen verstanden. Dies geschieht durch die transparente Verbindung von Mitarbeitern, Tools, Prozessen und Automatisierung.<sup>21</sup>

Dabei entsteht eine kollaborative Arbeitsumgebung, bei der für alle Mitarbeiter ersichtlich ist, woran gerade wie gearbeitet wird und ermöglicht eine ausführliche Historie der Tätigkeiten und der beteiligten Personen.<sup>22</sup>

Als Grundsatz von ChatOps gilt die Culture of Automation, Measurement and Sharing (CAMS), also eine Kultur, die Automatisierung, Messbarkeit und Transparenz mit sich bringt. Dieser Gedanke wurde dem DevOps Umfeld entliehen.<sup>23</sup>

Wenn Fehler auftreten, kann direkt geprüft werden, ob und welche Änderungen am System vorgenommen wurden und nicht alle Mitarbeiter müssen einzeln befragt werden.

Auch bei der Einarbeitung von neuen Mitarbeitern ist die Transparenz hilfreich, da sie sämtliche Arbeitsabläufe ihres Teams mitverfolgen können und dazu nicht einmal am gleichen Standort sein müssen. Sie können einfach den Räumen beitreten, die für ihre Rollen und Verantwortlichkeiten vorgesehen sind.<sup>24</sup>

Durch anpassbare Skripte und Plugins können individuelle Prozesse aus dem Chatroom heraus angestoßen werden. So kann z. B. in einem Softwareprojekt das Code Deployment gestartet werden, oder in der IT-Security auf gemeldete Sicherheitsvorfälle reagiert werden. Außerdem können Benachrichtigungen an die Teammitglieder versendet werden, die gerade zuständig sind.<sup>25</sup>

### 2.2.2 Komponenten

Um ChatOps einführen zu können, müssen als Kernkomponenten ein Kollaborationswerkzeug, ein Bot und eine Systemintegration gegeben sein.

Ein Kollaborationswerkzeug ist das Chat-Tool, in dem Stakeholder miteinander kommunizieren können und in Teams organisiert sind.

---

<sup>18</sup>Vgl. Puscher (2018), S. 68 f.

<sup>19</sup>Vgl. Newland (2013), S. 62.

<sup>20</sup>Vgl. Sigler (2014), o. S.

<sup>21</sup>Vgl. Zyane (2017), o. S.

<sup>22</sup>Vgl. Betz (2016), S. 190.

<sup>23</sup>Vgl. Zyane (2017), o. S.

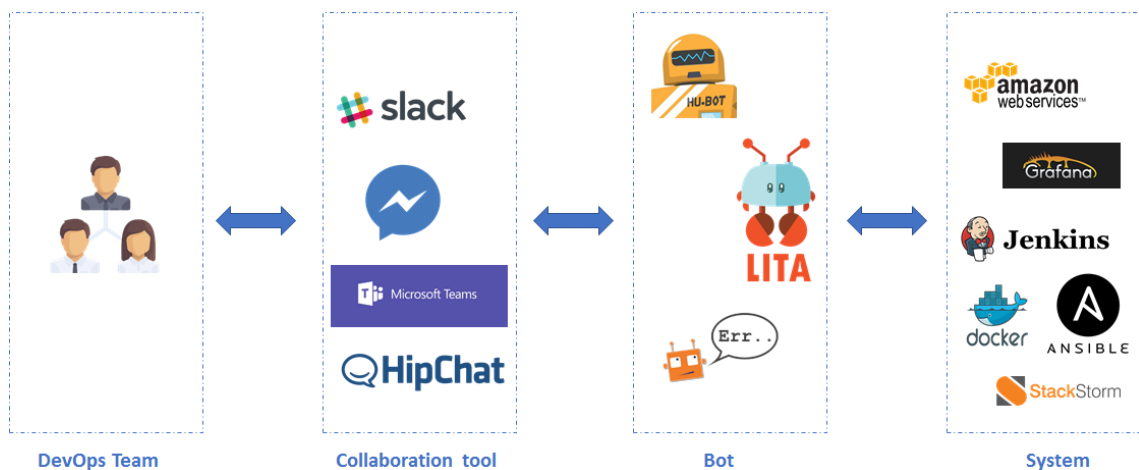
<sup>24</sup>Vgl. Sigler (2014), o. S.; Vgl. Hand (2016), S. 66.

<sup>25</sup>Vgl. Sigler (2014), o. S.

Ein Bot im ChatOps-Kontext bildet die Brücke zwischen Kollaborationswerkzeug und Systemwerkzeugen. Er erhält von den Teammitgliedern natürlichsprachige Befehle in Textform, interpretiert diese und fragt dann die angeforderte Information vom verbundenen System ab bzw. führt die angeforderte Aktion aus.

Als Systemintegration kommt jedes Tool in Frage, das von einem Unternehmen genutzt wird und eine Schnittstelle nach außen bietet, die vom Bot genutzt werden kann.<sup>26</sup>

Die beschriebenen Zusammenhänge sind in Abbildung 2.1 ersichtlich. Das Team arbeitet mit einem Kollaborationswerkzeug und die Automatisierung und Anbindung an die Systeme wird von einem Bot sichergestellt.



Quelle: Zyane (2017), o. S.

Abbildung 2.1: ChatOps Prozess

### 2.2.2.1 Kollaborationswerkzeug

Bekannte Beispiele für Kollaborationswerkzeuge sind Slack, Atlassian HipChat oder Cisco WebEx Teams, die allesamt die Integration von Bots ermöglichen.<sup>27</sup>

Ziel solcher Werkzeuge ist es, Kollaboration zu ermöglichen, die Kommunikation von Arbeitsteams effektiv zu gestalten und klassische Kanäle wie E-Mail oder SMS abzulösen. Dabei ist auch die Anbindung von anderen Diensten, wie z. B. Cloud Speicher zur kollaborativen Bearbeitung von Dokumenten möglich. So müssen Unternehmen nicht zwangsläufig die Software von einem bestimmten Hersteller beziehen, sondern können das Kollaborationswerkzeug ihrer Wahl an verschiedene Dienste anbinden.<sup>28</sup>

Slack zum Beispiel gilt als erfolgreichstes Kollaborationswerkzeug. Das Unternehmen wurde 2013 gegründet und hat heute 8 Millionen Nutzer, davon 3 Millionen zahlende und der Marktwert wird auf 7,1 Milliarden Dollar geschätzt. In der Basisversion ist es kostenlos, für Features wie längeres Vorhalten der Chatverläufe muss eine Lizenz erworben werden.<sup>29</sup>

Zum Februar 2019 hat Slack den Konkurrenten HipChat abgeschaltet und die bisherigen Kunden nach Slack migriert, nachdem dieser Ende 2018 übernommen wurde.<sup>30</sup>

<sup>26</sup>Vgl. Zyane (2017), o. S.

<sup>27</sup>Vgl. ebd., o. S.

<sup>28</sup>Vgl. Költzsch (2019), o. S.

<sup>29</sup>Vgl. ebd., o. S.

<sup>30</sup>Vgl. Donath (2018), o. S.

### 2.2.2.2 Bot

Als ChatOps Bots können z. B. Hubot, Lita, oder Errbot verwendet werden, wobei alle in etwa die gleiche Funktionalität besitzen, da sie sich am ursprünglichen ChatOps Bot Hubot orientieren. Allerdings sind sie in verschiedenen Sprachen geschrieben, was für die Weiterentwicklung und Anbindung an Systeme wichtig sein kann, und weisen einige Besonderheiten auf. Eine Übersicht ist in Tabelle 2.1 zu sehen.<sup>31</sup>

Cog hingegen ist ein Bot Framework, das möglichst plattformunabhängig arbeitet und ähnlich der Unix Funktion *Pipe* funktioniert, also eine Verknüpfung verschiedener Befehle in beliebiger Komplexität ermöglicht.<sup>32</sup>

Name	Sprache	Besonderheiten
Hubot	CoffeeScript	große Community
Lita	Ruby	viele Plugins
Errbot	Python	einfache Integration von APIs
Cog	(Framework)	Unix-ähnliches Piping

Quelle: Eigene Abbildung nach: Zyane (2017), o. S.

Tabelle 2.1: Verschiedene ChatOps Bots

### 2.2.2.3 Systemintegration

Ein Chatbot kann z. B. an das Ticketsystem, Versionskontrollsystem, Konfigurationsmanagementsystem oder Monitoringsystem angebunden werden, um dort automatisiert Aufgaben auszuführen. Eine Auflistung mit Beispielen ist in Tabelle 2.2 zu finden.<sup>33</sup>

System	Beispiele
Ticket	JIRA, OTRS, TeamForge
Versionskontrolle	GitHub, Gitlab, Bitbucket
Konfigurationsmanagement	Ansible, Salt, Chef, Puppet
Monitoring	Nagios, CheckMK, Grafana, Prometheus, Kibana
Continuous Integration	Jenkins, Travis CI, Bamboo
Infrastructure as Code (IaC)	Docker, Swarm, Kubernetes, Terraform, Vagrant

Quelle: Eigene Abbildung nach: Zyane (2017), o. S.

Tabelle 2.2: ChatOps Systemintegrationen

<sup>31</sup>Vgl. Zyane (2017), o. S.

<sup>32</sup>Vgl. ebd., o. S.

<sup>33</sup>Vgl. ebd., o. S.



## 2.3 CMDB

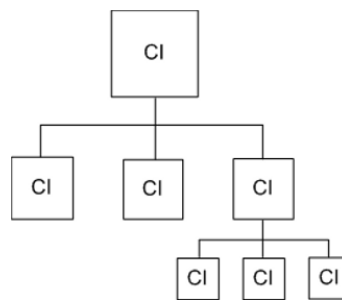
Eine Configuration Management DataBase (CMDB) ist ein aus dem IT Infrastructure Library (ITIL) Umfeld stammendes Modell, um die IT-Infrastruktur eines Unternehmens abzubilden. Mit ihr kann auf Informationen zu Diensten und Konfigurationen zugegriffen werden.<sup>34</sup>

Neben Hardware und Infrastruktur werden auch Dokumente wie Verträge, Service Level Agreements (SLA's) oder Notfallpläne, sowie Personen und Prozesse dort dokumentiert.<sup>35</sup>

Die Komponenten einer CMDB werden als Configuration Items (CI's) bezeichnet. Ein Configuration Item (CI) ist eine Einheit, die aus einem oder mehreren Objekten bestehen kann. Also z. B. eine CPU, ein Kabel, ein kompletter Server, ein Netzwerksegment oder ein Cluster.<sup>36</sup>

Zu den CI's wird geplant, welchen Zweck diese erfüllen und in welchem Kontext sie eingesetzt werden (technisch und organisatorisch). Es kann identifiziert werden, wer für das CI zuständig ist und in welcher Version es vorliegt. Sämtliche Änderungen an CI's werden dokumentiert, wodurch eine vollständige Änderungshistorie über den gesamten Lebenszyklus entsteht.<sup>37</sup>

Ein CI kann verschiedene Eigenschaften bzw. Attribute besitzen und es können Verlinkungen zu anderen CI's hergestellt werden, wodurch die Bildung einer Objekthierarchie möglich ist. Eine beispielhafte Anordnung ist in Abbildung 2.2 zu sehen. CI's werden außerdem in eine Kategorie eingeordnet und es kann ein Status zugewiesen werden.<sup>38</sup>



Quelle: Olbrich (2008), S. 74

Abbildung 2.2: Configuration Item Beziehungen

Neben rein hierarchischer Untergliederung sind auch andere Beziehungen zwischen CI's möglich. Eine Übersicht mit Beispielen ist in Tabelle 2.3 zu sehen.

Durch die Verknüpfungen entsteht eine Übersicht, welche Systeme bei kritischen Ereignissen betroffen sind. Z. B. welche Auswirkungen der Ausfall eines Switches hat.<sup>39</sup>

<sup>34</sup>Vgl. Olbrich (2008), S. 70 f.

<sup>35</sup>Vgl. Sturm, Pollard und Craig (2017), S. 253.

<sup>36</sup>Vgl. Olbrich (2008), S. 70 f.

<sup>37</sup>Vgl. ebd., S. 70 f.

<sup>38</sup>Vgl. ebd., S. 72 ff.

<sup>39</sup>Vgl. ebd., S. 74 f.

Beziehung	Beispiel
X ist Teil von Y	eine CPU ist Teil eines Servers
X ist verbunden mit Y	ein Server ist mit einem Storagesystem verbunden
X benutzt Y	zwei Netzwerke benutzen eine VPN Verbindung
X ist eine Ausprägung von Y	gleicher Server, andere RAM Bestückung

Quelle: Eigene Abbildung nach: Olbrich (2008), S. 74

Tabelle 2.3: Beziehungen zwischen Configuration Items

Jedes CI durchläuft einen Lebenszyklus, der aus Beschaffung (Procurement), Betrieb (Operating) und Entsorgung (Disposal) besteht. Die Beschaffung beinhaltet Bestellung und Lieferung. Sobald das CI betriebsbereit ist, geht es in die Betriebsphase und wenn es ausgemustert wird, geht es in die Entsorgungsphase.

Je nach Phase können die CI's verschiedene Zustände einnehmen. Diese können einen veränderbaren (V) oder finalen (F) Status haben. Eine Übersicht der Zustände pro Phase ist in Tabelle 2.4 zu sehen.<sup>40</sup>

---

<sup>40</sup>Vgl. Olbrich (2008), S. 76 f.

<i>Beschaffung</i>		
Status	Art	Beschreibung
ordered	V	Bestellung wurde ausgelöst
in progress	V	Bestellvorgang wird bearbeitet
cancelled	F	Bestellung wurde storniert
delivered	V	Ware wurde geliefert
returned	F	Ware wurde zurückgesendet

<i>Betrieb</i>		
Status	Art	Beschreibung
active	V	CI ist im produktiven Einsatz
inactive	V	CI ist temporär nicht im Einsatz (Wartung)
in store	V	CI ist im Lager
planning	V	CI wird geplant
test	V	CI wird getestet

<i>Entsorgung</i>		
Status	Art	Beschreibung
stolen	V	CI wurde gestohlen
sold	F	CI wurde verkauft
discarded	F	CI wurde ausgemustert

Quelle: Eigene Abbildung nach: Olbrich (2008), S. 77

Tabelle 2.4: Status von Configuration Items

## 2.4 Python

Python ist eine universelle Programmiersprache, die auf vielen Betriebssystemen, wie Windows, Linux oder macOS läuft. Sie gilt als leicht zu erlernen und bietet objektorientierte Features. Sie wurde 1989 von Guido van Rossum als Nachfolger der Sprache *ABC* entwickelt. Der Name ist angelehnt an die Fernseh-sendung *Monty Python's Flying Circus*.

Durch die *batteries included* Philosophie sind viele Features schon in Python integriert.<sup>41</sup>

Um zusätzliche Module zu verwenden, wird empfohlen eine virtuelle Umgebung (*virtualenv*) zu verwenden. So können Konflikte vermieden werden, wenn Tools Module in unterschiedlichen Versionen benötigen. Braucht z. B. ein Tool *A* das Modul *M* in der Version *1.0* und ein Tool *B* selbiges in der Version *2.0*, so kann für jedes Tool ein eigenes *virtualenv* angelegt werden. Auch Updates können in den Umgebungen getrennt von einander erfolgen.

Für gewöhnlich stellen Tools eine *requirements.txt* Datei bereit, in der benötigte Module und deren Versionen vermerkt sind. Ein Beispiel ist in Listing 2.1 zu finden.<sup>42</sup>

Listing 2.1: Beispiel requirements.txt

```
1 novas==3.1.1.3
2 numpy==1.9.2
3 requests==2.7.0
```

## 2.5 Application Programming Interface

Eine API ist eine Programmierschnittstelle, um Informationen auf standardisiertem Weg auszutauschen. Daten und Befehle werden strukturiert und in einer festgelegten Syntax übergeben.

Der Einsatz einer API erlaubt die Modularisierung von Programmen, lediglich die Schnittstellen der Programmmodule müssen spezifiziert sein. Außerdem bietet eine unveränderte API eine hohe Stabilität auf lange Sicht, während der Quellcode darunter beliebig verändert werden kann.

Häufig verwendet werden APIs bei Online-Shops oder Content-Management-Systemen, z. B. um Bezahl-dienstleister, Bewertungssysteme oder Versanddienstleister anzubinden.<sup>43</sup>

<sup>41</sup>Vgl. Gowrishankar und Veena (2019), S. 1 ff.

<sup>42</sup>Vgl. Python Software Foundation (2019), o. S.

<sup>43</sup>Vgl. Luber (2017), o. S.

## Kapitel 3

# Anforderungsanalyse anhand von Experteninterviews

Um die Anforderungen an einen CMDB Chatbot zu erfassen, wird in diesem Kapitel eine Anforderungsanalyse nach Pohl und Rupp durchgeführt, die als Werkzeug zur Anforderungsermittlung das Experteninterview nach Meuser und Nagel verwendet.

Ziel des Anforderungsmanagements bzw. Requirements Engineerings nach Pohl und Rupp ist die Erfassung und Dokumentation von Anforderungen.<sup>44</sup>

Wesentliche Funktionen des Experteninterviews sind Informationsgewinnung und Theorieentwicklung, weswegen es sich als Methode eignet, um Anforderungen zu erheben.<sup>45</sup>

Wenn von Experteninterviews die Rede ist, kann man davon ausgehen, dass ein leitfadengestütztes, qualitatives Interview gemeint ist. Da das Instrument Experteninterview dem Forschungsvorhaben angepasst werden muss, können jedoch verschiedene Dinge unter dem Begriff verstanden werden und es ist nötig klarzustellen, welche Ausprägung gemeint ist.<sup>46</sup> Auch der Expertenbegriff muss der Forschungsfrage angepasst werden.<sup>47</sup>

In dieser Arbeit wird der qualitativ ausgerichtete Ansatz von Meuser und Nagel angewendet. Quantitative Ansätze oder qualitative Ansätze anderer Autoren, die von dem behandelten abweichen werden hier nicht weiter ausgeführt.<sup>48</sup>

Meuser und Nagel unterscheiden Experteninterviews grundlegend in systematisierende und explorativ-felderschließende Interviews. Bei ersteren wird der Experte zu seinem eigenen Fachgebiet befragt (Betriebswissen), bei letzterem liefert er „Informationen über die Kontextbedingungen des Handelns der Zielgruppe“<sup>49</sup> (Kontextwissen).<sup>50</sup>

Bogner, Littig und Menz sprechen in diesem Zusammenhang von fundierenden und explorativen Interviews und fassen die vormaligen genannten Kategorien unter dem Begriff *informativ* zusammen. Außerdem ergänzen sie zwei deutungswissenorientierte Varianten. Eine genaue Aufschlüsselung ist in Tabel-

---

<sup>44</sup>Vgl. Pohl und Rupp (2015), S. 3.

<sup>45</sup>Vgl. Bogner, Littig und Menz (2014), S. 9; Vgl. Gläser und Laudel (2010), S. 187 f.

<sup>46</sup>Vgl. Bogner, Littig und Menz (2014), S. 3.

<sup>47</sup>Vgl. Meuser und Nagel (1994), S. 180.

<sup>48</sup>Vgl. Meuser und Nagel (2010), o. S.

<sup>49</sup>Meuser und Nagel (1991), S. 445.

<sup>50</sup>Vgl. ebd., S. 445.

le 3.1 zu sehen.<sup>51</sup>

	Explorativ (Kontextwissen)	Fundierend (Betriebswissen)
Informatorisch	Explorative Datensammlung	Systematisierendes Interview
Deutungswissenorientiert	Exploration von Deutungen	Theoriegenerierendes Interview

Quelle: Eigene Abbildung nach: Bogner, Littig und Menz (2014), S. 22 ff.; Meuser und Nagel (1991), S. 445 ff.

Tabelle 3.1: Arten von Experteninterviews

Als Interviewart wird in dieser Arbeit das *informatorische Interview*, genauer gesagt die *explorative Datensammlung* genutzt, um von den Experten *Kontextwissen* abzufragen.

Anforderungen werden dabei als Aussagen über Leistungen oder Fähigkeiten, die von einem System zu erbringen sind und die entsprechenden zur Zielerreichung nötigen Eigenschaften verstanden.<sup>52</sup>

Die Haupttätigkeiten der Anforderungsanalyse sind die Ermittlung, Dokumentation, Prüfung und Verwaltung von Anforderungen, wobei die Verwaltung mit den drei vorigen Tätigkeiten einher geht.<sup>53</sup>

Ein Experteninterview kann aus bis zu acht Phasen bestehen. Der Leitfadenerstellung, Tonaufnahme, Transkription, Paraphrase, (Bildung von) Überschriften, dem thematischen Vergleich, der soziologischen Konzeptualisierung und theoretischen Generalisierung. Da Kontextwissen und nicht Betriebswissen abgefragt wird, empfehlen Meuser und Nagel, das Experteninterview mit dem thematischen Vergleich abzuschließen und die letzten beiden Phasen entfallen zu lassen. Es wird also in diesem Fall keine soziologische Konzeptualisierung und theoretische Generalisierung durchgeführt.<sup>54</sup>

Die Phasen und Parallelen von Anforderungsanalyse und Experteninterview sind in einer Übersicht in Tabelle 3.2 dargestellt.

	Anforderungsanalyse	Experteninterview
4. Verwaltung	1. Ermittlung	1. Leitfadenerstellung 2. Tonaufnahme
	2. Dokumentation	3. Transkription 4. Paraphrase 5. Überschriften
	3. Prüfung und Abstimmung	6. Thematischer Vergleich <del>7. Soziologische Konzeptualisierung</del> <del>8. Theoretische Generalisierung</del>

Tabelle 3.2: Experteninterview und Anforderungsmanagement im Vergleich

### 3.1 Aktuelle Situation

Das IVZ verwendet die *i-doit* CMDB des Herstellers synthetics in dreistufiger Ausführung, also Entwicklung, Integration und Produktion. Sämtliche Hardware wie Server, Infrastruktur (Local Area Network

<sup>51</sup>Vgl. Bogner, Littig und Menz (2014), S. 22 ff.; Vgl. Meuser und Nagel (1991), S. 445 f.

<sup>52</sup>Vgl. Pohl und Rupp (2015), S. 3.

<sup>53</sup>Vgl. ebd., S. 4 f.

<sup>54</sup>Vgl. Meuser und Nagel (1991), S. 466 f.

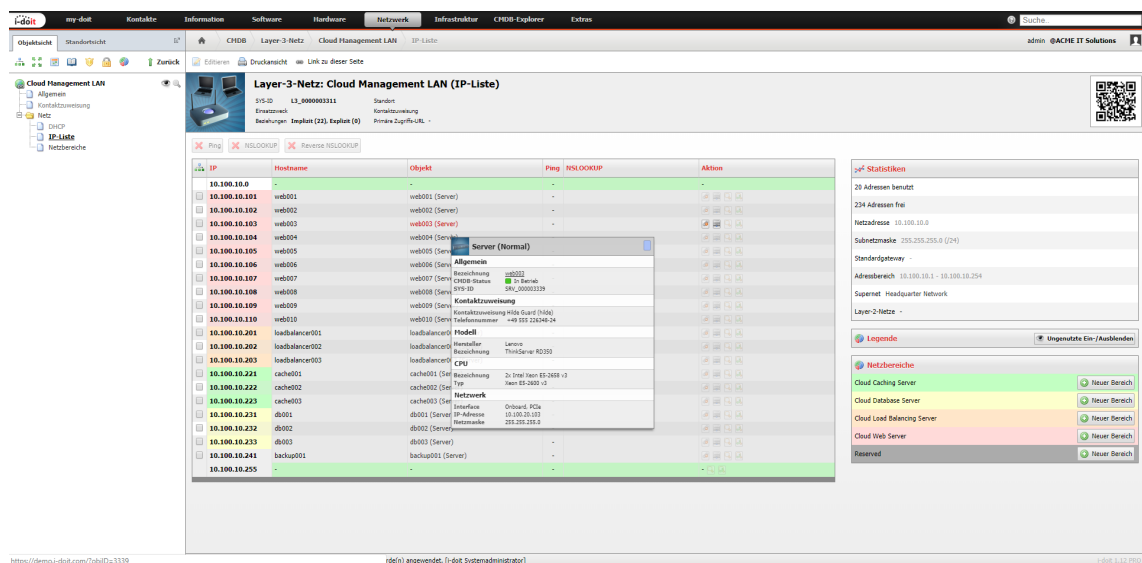
(LAN), Storage Area Network (SAN) Switches usw.) und Verkabelung sind in dieser CMDB erfasst. Auch Clients wie Computer und Laptops sowie mobile Endgeräte wie Tablets und Smartphones werden in die CMDB eingetragen.

Virtuelle Maschinen und Cluster sind ebenfalls in der CMDB hinterlegt.

Neben Hardware werden auch Softwarelizenzen, Wartungsverträge, Service Level Agreements (SLA's) und der IT Infrastructure Library (ITIL) Servicekatalog abgebildet. Dadurch können auslaufende Lizenzen und Verträge rechtzeitig erkannt und verlängert oder erneuert werden.

Beim Anlegen von virtuellen Maschinen oder Hardware Servern kann direkt der entsprechende Service oder Wartungsvertrag verknüpft werden. So entsteht auch ein Überblick darüber, welche Hardware wann aus der Wartung läuft und zu welchem Grad das Lizenzkontingent erschöpft ist. Gleiches gilt für Clients und deren Betriebssystem und etwaige installierte, lizenzpflichtige Software.

Auch Domain Name System (DNS) Einträge und Layer-2 und 3 Netze sind in der CMDB dokumentiert. Beim Anlegen oder Planen eines neuen Servers (physikalisch oder virtuell) können so direkt ein Hostname und eine IP-Adresse reserviert werden, bevor die Eintragung im DNS-Server erfolgt. Das entsprechende User Interface ist in Abbildung 3.1 zu sehen. Etwaige Konflikte fallen so frühzeitig auf, z. B. durch Ping, Lookup oder Reverse Lookup der gewünschten Adressen aus der IP Liste heraus kann die Verfügbarkeit der Adressen überprüft werden, wobei bestimmte Bereiche pro Netz der Infrastruktur vorbehalten sind.



Quelle: synetics (2019b), o. S.

Abbildung 3.1: i-doit IP Management

Die CMDB wird überwiegend vom Team *Rechenzentrum* über das Webinterface verwendet. Wenn z. B. ein neuer Server angefordert wird, erfolgt die Planung von Hostname und IP-Adresse in der CMDB. Auch beim Einbau von Hardware, z. B. neuen Servern, wird die CMDB prozessvorbereitend, -begleitend und -unterstützend eingesetzt: Es kann vorab geprüft werden, in welchen Racks noch genug Platz ist und wo entsprechende Möglichkeiten zur Verkabelung von Strom und Netzwerk bestehen. Wenn geprüft wurde, ob Kabel in der benötigten Länge vorrätig sind, kann der Einbau erfolgen und der Status der Hardware wird von *geplant* auf *in Betrieb* geändert. Die verwendeten Kabel werden ebenfalls dokumentiert und die entsprechenden Knotenpunkte (z. B. Netzwerkartenport zu Switchport oder Netzteil zu Steckerleiste) verknüpft.

Um die CMDB von der Kommandozeile aus verwenden zu können, besteht die Möglichkeit, *i-doit-cli* zu nutzen, einen Application Programming Interface (API)-Client für die i-doit CMDB.

Mit dem Client lassen sich z. B. Objekte suchen, Details dazu anzeigen und freie IP-Adressen in einem bestimmten Netz anzeigen. Um diese Operationen zu beschleunigen wird ein lokales Caching verwendet. Außerdem können Dummy Einträge erzeugt werden, um die System-Performance zu testen. Der Client wird bisher nur in einigen Skripten verwendet und hat für die manuellen Tätigkeiten der Mitarbeiter keine Relevanz.<sup>55</sup>

Als Chat-Tool wird derzeit Slack eingesetzt, allerdings ist die Migration nach Cisco WebEx Teams geplant.

## 3.2 Expertenauswahl

Um das Experteninterview anwenden zu können, muss geklärt werden, wer als Experte gilt, was Expertenwissen bedeutet und wie man es abrufen kann.<sup>56</sup>

### 3.2.1 Kriterien

Das Wort *Experte* leitet sich aus dem lateinischen *expertus* (erprobt, bewährt) ab, welches wiederum aus dem passiven Verb *experiri* (prüfen, ausprobieren) gebildet wird. Experten sind also Fachleute und Kenner, bzw. allgemein gesprochen Personen, die über ein gewisses Spezialwissen verfügen, das auf der Erfahrung des ausgiebigen Prüfens und Ausprobierens fußt.<sup>57</sup>

Gläser und Laudel definieren Experten als Menschen, die ein besonderes Wissen über soziale Sachverhalte besitzen.<sup>58</sup>

Man könnte nun davon ausgehen, dass jeder ein Experte ist, da er sich in seinem Fachgebiet gut auskennt. Genauer betrachtet gibt es jedoch große Unterschiede zwischen Laien und Experten. Ein solcher Expertenbegriff ist also zu weit gefasst.<sup>59</sup>

Ein Experte zeichnet sich durch eine bestimmte Kombination von Macht und Wissen aus, welche ihn von Eliten, Spezialisten und spezialisierten Laien unterscheidet. Dieser Zusammenhang ist in Abbildung 3.2 erkennbar.<sup>60</sup>

---

<sup>55</sup>Vgl. Heisig (2019), o. S.

<sup>56</sup>Vgl. Bogner, Littig und Menz (2014), S. 6 f.

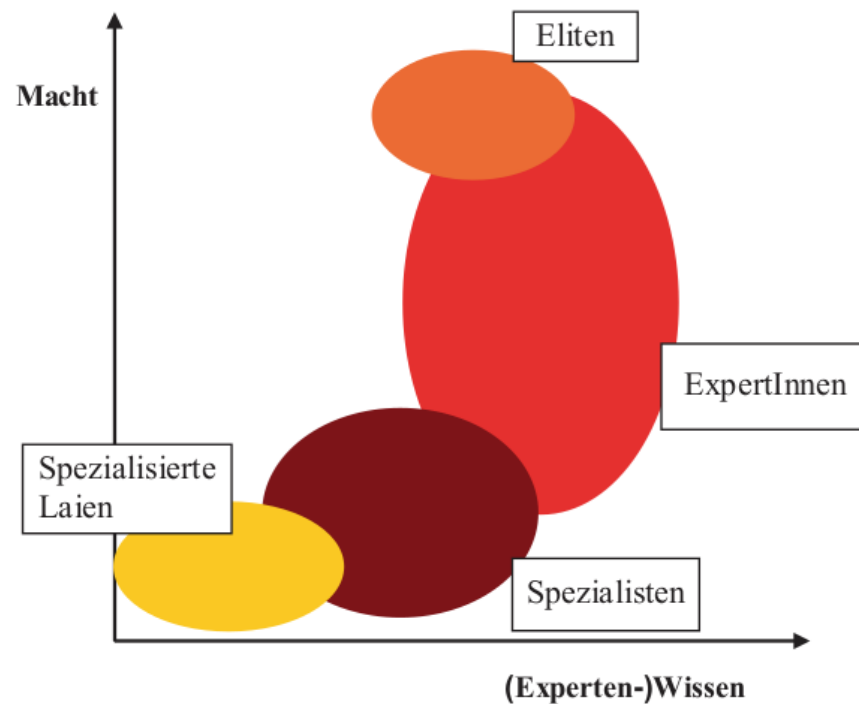
<sup>57</sup>Vgl. ebd., S. 9.

<sup>58</sup>Vgl. Gläser und Laudel (2010), S. 12.

<sup>59</sup>Vgl. Bogner, Littig und Menz (2014), S. 10 f.

<sup>60</sup>Vgl. Littig (2008), o. S.





Quelle: Littig (2008), o. S.

Abbildung 3.2: Experteneinordnung

„Der Experte ist – im Gegensatz zum Spezialisten – nicht allein durch Sonderwissen in Form fachspezifischer Kompetenzen charakterisiert, sondern durch seine Fähigkeit, Verbindungen zu anderen Wissensbeständen und Wissensformen herzustellen und die Relevanz des eigenen Wissens zu reflektieren“.<sup>61</sup>

Das Interessante am Expertenwissen ist nicht nur das Wissen an sich, sondern die Wirkung in der Praxis. Experten werden gefragt, weil ihre Entscheidungen und Ratschläge das Handeln anderer Akteure mitstrukturieren und -gestalten.<sup>62</sup>

Bogner spricht davon, dass Expertenwissen „seine Bedeutung über [...] soziale Wirkmächtigkeit“<sup>63</sup> erhält. Ferner folgert er daraus die Definition:

„Experten lassen sich als Personen verstehen, die sich – ausgehend von einem spezifischen Praxis- oder Erfahrungswissen, das sich auf einen klar begrenzbaren Problemkreis bezieht – die Möglichkeit geschaffen haben, mit ihren Deutungen das konkrete Handlungsfeld sinnhaft und handlungsleitend für Andere zu strukturieren.“<sup>64</sup>

Meuser und Nagel betrachten den Experten als Konstrukt des Forschungsinteresses. Die Expertise ist also keine allgemeingültige Eigenschaft oder Fähigkeit einer Person, sondern eine Zuschreibung des Forschenden an diese, im speziellen Kontext der Forschungsfrage.<sup>65</sup>

Zusammenfassend lässt sich also sagen, dass Experten über besonderes Wissen auf einem Gebiet verfügen und eine machtvollen Positionen innehaben können, die zwischen Spezialisten und Eliten liegt. Die Expertise muss nicht durch einen Titel oder Ähnliches verifiziert werden, sie kann auch durch jahrelange berufliche oder private Auseinandersetzung mit einem Thema erlangt werden.

<sup>61</sup>Bogner, Littig und Menz (2014), S. 14; Vgl. Hitzler (1994), S. 21 f.

<sup>62</sup>Vgl. Bogner, Littig und Menz (2014), S. 13.

<sup>63</sup>Ebd., S. 13.

<sup>64</sup>Ebd., S. 13.

<sup>65</sup>Vgl. Meuser und Nagel (1994), S. 181.

Die Aufgabe des Forschers ist es demzufolge, die Experten für seine spezielle Forschungsfrage ausfindig zu machen und zu befragen.

Im Falle dieser Arbeit stammen die Experten aus dem beruflichen Umfeld des Autors und haben sich durch wichtige Positionen oder herausragende Leistungen auf dem Gebiet des Forschungsthemas hervor getan.

### 3.2.2 Vorstellung

Als Experten zur Befragung wurden die IVZ Mitarbeiter Alexander Hemmerich und Jörg Middendorf, sowie die synetics Mitarbeiter Konrad Buck und Daniel Kirsten ausgewählt.

Hemmerich und Middendorf sind seit 2011 Teil der IVZ Belegschaft und können daher aus einiger Erfahrung schöpfen. Hemmerich ist aktuell Leiter des Rechenzentrums am Standort Köln mit den Tätigkeitsschwerpunkten Storage, Backup und Loadbalancer. Somit hat er einen guten Gesamtüberblick über die Tätigkeiten des Rechenzentrums.

Middendorf ist neben der Unix- und Storage Administration auch für die CMDB des IVZ zuständig und hat somit ein umfangreiches Wissen über Infrastruktur, Aufbau, Funktionen und Schnittstellen der CMDB.

Buck ist PR- und Community-Manager sowie Kommunikationsberater der synetics GmbH für das Produkt i-doit. Er hat Publizistik und Germanistik studiert und sich als IT-Fachjournalist fortgebildet. Für synetics übernimmt er die Moderation und Kundenkommunikation, um für dauerhafte Kundenbindung zu sorgen. Durch seine Tätigkeiten hat er ein gutes Gesamtbild der Unternehmen, die eine i-doit CMDB des Herstellers synetics einsetzen und kann den Nutzen von ChatOps für diese beurteilen.<sup>66</sup>

Kirsten arbeitet seit 1998 bei der synetics GmbH und ist Produktmanager für die i-doit CMDB. Zuvor war er als IT-Berater im Bereich Infrastruktur und Server tätig und konnte Erfahrung zu den Themen Routing/Switching, virtuelle Umgebungen und Speichernetze sammeln.<sup>67</sup>

Die Experten selbst haben nicht direkt mit Chatbots zu tun, sondern sollen sie in ihrem beruflichen Kontext, auf dem sie Experten sind, also Rechenzentrumsbetrieb und CMDB, bewerten. Durch die genannten Experten werden wichtige Schlüsselfunktionen betrachtet und deren Perspektiven berücksichtigt. Auf normale *Verwender* der CMDB kann daher als Interviewpartner verzichtet werden.

## 3.3 Ermittlung der Anforderungen an einen CMDB Chatbot

Die Ermittlung der Anforderungen ist der erste Schritt des Anforderungsmanagements. Im Experteninterview werden darunter die ersten beiden Schritte *Leitfadenerstellung* und *Tonaufnahme* verstanden.

Bei der Ermittlung der Anforderungen können verschiedene Quellen, wie z. B. Stakeholder, Dokumente oder aktive Systeme genutzt werden, um Anforderungen zu detektieren und zu verfeinern.<sup>68</sup>

Das Interview ist eine mögliche Befragungstechnik, bei der ein Stakeholder (Interviewpartner) vom Requirements Engineer (Interviewer) vorgegebene Fragen gestellt bekommt. Die Antworten werden protokolliert. Etwaige Rückfragen können sofort im Gesprächsverlauf geklärt werden. Durch geschickt gestellte Fragen können auch unterbewusste Anforderungen ermittelt werden. Der Verlauf des Gesprächs kann individuell angepasst werden, es wird gezielt nachgefragt und auf einzelne Themen eingegangen, um das

---

<sup>66</sup>Vgl. Buck (2018b), o. S.; Vgl. Buck (2018a), o. S.

<sup>67</sup>Vgl. Kirsten (2017), o. S.

<sup>68</sup>Vgl. Pohl und Rupp (2015), S. 21.

behandelte Thema möglichst komplett abzubilden.<sup>69</sup>

### 3.3.1 Leitfadenerstellung

Der Einsatz eines Leitfadens dient dazu, das Interview thematisch zu begrenzen und würdigt den Expertenstatus durch die damit verbundene Auseinandersetzung mit dem Thema. Der Leitfaden ist dabei nicht strikt zu befolgen, sondern bietet lediglich das Grundgerüst des Interviews. Reihenfolge und Tiefe der Fragen bzw. Themenkomplexe können zwischen den Interviews variieren. Die Fragen sollten dabei nicht abgelesen werden, sondern in der „Sprache des Experten“<sup>70</sup> gestellt werden.<sup>71</sup>

Leitfäden in der qualitativen Forschung liegen in Art und Umfang in verschiedensten Ausprägungen vor. Je nach Forschungs- und Interviewstil reichen sie von groben Themensammlungen bis hin zu ausführlichen Leitfäden mit ausformulierten Fragen.

Ein Leitfaden besteht aus verschiedenen Themenblöcken, mit darin enthaltenen Hauptfragen, die sich weiter in Teilaspekte aufgliedern können. Die Hauptfragen sollten bei dem Interview auf jeden Fall zur Sprache kommen und die Teilaspekte können angesprochen werden, um die Frage zu vertiefen.<sup>72</sup>

Der allgemeingültige Leitfaden für alle in dieser Untersuchung zu Wort kommenden Experten ist in Anhang A zu finden. Er gliedert das Thema in die drei Themenkomplexe *CMDB im Rechenzentrum*, *Chatbots allgemein* und *Chatbots zur Bedienung der CMDB*.

### 3.3.2 Tonaufnahme

Die Interviews mit Alexander Hemmerich und Jörg Middendorf wurden persönlich durchgeführt und mit der Audiorekorderfunktion eines Handys aufgezeichnet. Die Interviews mit Daniel Kirsten und Konrad Buck erfolgten in dem Konferenztool Cisco WebEx Meeting und wurden mit dessen Aufnahmefunktion aufgezeichnet.

## 3.4 Dokumentation der Anforderungen

Die Dokumentation der Anforderungen ist der zweite Schritt des Anforderungsmanagements. Im Experteninterview werden darunter die Schritte drei bis fünf, *Transkription*, *Paraphrase* und *(Bildung von) Überschriften* verstanden und es ist das einzelne Interview im Fokus.

Die erarbeiteten Anforderungen werden in natürlicher Sprache oder in dafür vorgesehenen Modellen dokumentiert, wobei entsprechende Techniken zum Einsatz kommen.<sup>73</sup>

Die im Requirements Engineering anfallenden Tätigkeiten müssen geeignet dokumentiert werden. Dazu zählt z. B. auch die Erstellung von Interviewprotokollen. Als Dokumentation gilt jede Form der formalen Darstellung, von Beschreibung in natürlicher Sprache über strukturierten Text bis hin zu formalen Techniken wie Diagrammen. Die Dokumentationsform sollte der zugrundeliegenden Aktivität angemessen sein und diese sinnvoll wiedergeben.<sup>74</sup>

---

<sup>69</sup>Vgl. Pohl und Rupp (2015), S. 28.

<sup>70</sup>Meuser und Nagel (1991), S. 449.

<sup>71</sup>Vgl. ebd., S. 448 f.

<sup>72</sup>Vgl. Bogner, Littig und Menz (2014), S. 27 f.

<sup>73</sup>Vgl. Pohl und Rupp (2015), S. 4.

<sup>74</sup>Vgl. ebd., S. 35 ff.

### 3.4.1 Transkription

Das Hauptaugenmerk von Experteninterviews liegt auf der Gewinnung von Wissen. Daher ist eine mühselige Notation, wie sie bei narrativen Interviews vorgenommen wird nicht nötig. Elemente wie Pausen, Betonung oder Stimmlage werden nicht interpretiert. Eine Transkription wird in der Regel nicht für die gesamte Tonaufnahme durchgeführt, sondern kann selektiv erfolgen. Bei der Abfrage von Betriebswissen fällt die Transkription umfangreicher aus als bei Kontextwissen.<sup>75</sup>

Zur Transkription wurden die Audioaufnahmen unter Linux mit Music Player Daemon (MPD) wiedergegeben und mit dem Music Player Client (MPC) gesteuert. Dieser erlaubt die Bedienung des Backends mit Kommandozeilenbefehlen. So lassen sich globale Tastaturkürzel festlegen, um die Wiedergabe zu pausieren oder einige Sekunden zurückzuspringen und diese können genutzt werden, um die Wiedergabe zu steuern, ohne den Editor zu verlassen oder die Hände von der Tastatur zu heben.

Die Transkripte der Interviews sind in Anhang B, Anhang C, Anhang D und Anhang E zu finden.

### 3.4.2 Paraphrase

Die Paraphrase erfolgt unter Berücksichtigung der Forschungsfrage und gibt die Aussagen der Experten chronologisch geordnet und unverfälscht wieder, wobei diese zu thematischen Verbänden zusammengefasst werden. Ob die Passage zusammenfassend oder detailliert wiedergegeben wird hängt von der Priorität des Themas und nicht zwangsläufig von der ihm gewidmeten Zeit ab.<sup>76</sup>

In den folgenden Abschnitten werden die einzelnen Interviews paraphrasiert. Dabei werden die in Unterabschnitt 3.4.3 vorgesehenen Überschriften der besseren Lesbarkeit halber bereits der Passage vorangestellt.

#### 3.4.2.1 Interview Hemmerich

**CMDB allgemein:** Es soll zunächst das Interview mit Alexander Hemmerich betrachtet werden. Er beschreibt die CMDB als Tool, um „Configuration Items im Rechenzentrum, vom Kabel über Server über Ausstattung innerhalb eines Rechnerraums“<sup>77</sup> zu verwalten und zählt auch die vermerkten Lizenzen und Supportverträge dazu.

Seiner Meinung nach lässt sich die Vielzahl an Komponenten „gar nicht mehr anders handhaben“<sup>78</sup>. Außerdem führt er an, dass die CMDB gefordert sei, um eine ISO 27001 Zertifizierung zu erlangen.<sup>79</sup>

**Bestandsprüfung, Statusabfrage, Ressourcenübersicht:** Bei der täglichen Arbeit nutzt Hemmerich die CMDB „[u]m Gegenstände und Bestände abzufragen“<sup>80</sup>, also beispielsweise zur Prüfung des aktuellen Kabelbestands im Lager oder um den Status eines Servers abzufragen (eingelagert, verschrottet, etc.). Außerdem nutzt er die Report Funktion, um zu ermitteln welche Ressourcen ein Kunde oder eine Anwendung belegt (RAM, CPU, etc.). Dies sei einerseits bei der Abrechnung ausschlaggebend und andererseits bei der Beantragung neuer Dienstleistungen, wie „zusätzliche[n] virtuelle[n] Maschinen“<sup>81</sup>, nötig, um festzustellen, ob die Anforderungen durch den bestehenden Vertrag abgebildet sind oder ob eine Erweiterung erfolgen muss. „[D]as ... [geht] ... zu unserem Vertrieb und daraus entsteht ... eine Preisinformation

---

<sup>75</sup>Vgl. Meuser und Nagel (1991), S. 455 f.

<sup>76</sup>Vgl. ebd., S. 456 f.

<sup>77</sup>Hemmerich (2019), o. S.

<sup>78</sup>Ebd., o. S.

<sup>79</sup>Vgl. ebd., o. S.

<sup>80</sup>Ebd., o. S.

<sup>81</sup>Ebd., o. S.

für den Kunden“<sup>82</sup>

**CMDB Schnittstellen:** Hemmerich hält die CMDB für besonders wichtig, da sie „das führende System ist“<sup>83</sup>. Das Monitoring System und ein Scan System liefern Daten zu und diese werden in der CMDB „aggregiert, um eine Gesamtsicht auf das Rechenzentrum, auf die Ausstattung und sogar auf die Standorte des Unternehmens zu bieten.“<sup>84</sup>.

Außerdem ergänzt er, dass neben Configuration Items des Rechenzentrums auch Gebäude, Büros und Mitarbeiter in der CMDB vermerkt sind.<sup>85</sup>

**Chatbots allgemein:** Als Chatbots betrachtet Hemmerich Bots, „mit denen man kommunizieren kann“<sup>86</sup> und die auf Anfrage „Informationen liefern“<sup>87</sup> oder „Aktion[en] ausführen“<sup>88</sup>. Persönlich hat er noch keine Erfahrungen mit Chatbots oder ChatOps gemacht, er blickt jedoch gespannt auf deren großes Einsatzspektrum.<sup>89</sup>

**Chatbots zur Automatisierung:** Als Ersatz für menschliche Arbeitskräfte betrachtet Hemmerich Chatbots nicht, sondern vielmehr als Unterstützung bei der täglichen Arbeit. „Ein Chatbot könnte einfache Tätigkeiten übernehmen und zur Automatisierung des täglichen Rechenzentrumsbetriebs beitragen.“<sup>90</sup> Aus seiner Sicht können dadurch alltägliche Arbeitsschritte vereinfacht werden.<sup>91</sup>

**Suchfunktion, Nächste freie IP, Serverplanung, Schnittstellenanbindung:** Als mögliche Funktionen für einen Chatbot, der die CMDB bedient kann Hemmerich sich die Suche nach Objekten und die Anzeige der nächsten freien IP-Adresse in einem bestimmten Netz vorstellen. Dabei hat er einen Workflow vor Augen, bei dem ein Server über den Bot geplant und mit einem Namen versehen wird. Daraufhin soll für diesen eine freie IP-Adresse in einem bestimmten Netz gesucht und reserviert werden.

Als sinnvolle Ergänzung würde er hier begrüßen, dass in diesem Workflow direkt die virtuelle Maschine bereitgestellt, die Konfiguration per Puppet vorgenommen und die Netzkonfiguration hergestellt wird. Davon verspricht er sich eine Vermeidung von Inkonsistenzen zwischen CMDB und Realität.<sup>92</sup>

Dies übersteigt den CMDB-Fokus, kann jedoch im Ausblick und in der Möglichkeit einer Anbindung von weiteren Tools und Schnittstellen berücksichtigt werden.

**Vermeidung Fehlbedienung, Berechtigungskonzept, Nachvollziehbarkeit:** Die Integrität der CMDB ist für Hemmerich ein hohes Gut und eine versehentliche Löschung oder Manipulation von Daten durch „Fehlfunktion oder Fehlbedienung [des] Bots“<sup>93</sup> sollte vermieden werden. Bei der Entwicklung sollte darauf geachtet werden, die Betriebssicherheit zu gewährleisten.

Außerdem legt Hemmerich Wert darauf, dass nur ein bestimmter Personenkreis berechtigt ist, Änderungen oder Löschungen vorzunehmen und es soll nachvollziehbar sein, wer Änderungen vorgenommen hat.<sup>94</sup>

**Akzeptanz, Webinterface Eingaben:** Probleme erwartet Hemmerich bei der Einführung eines CMDB Chatbots nicht, er könnte sich jedoch vorstellen, dass ältere Kollegen etwas Zeit brauchen, um sich daran zu gewöhnen und lieber weiterhin die Eingaben über das Webinterface vornehmen. Da es sich aber größtenteils um junge, motivierte Menschen handelt, „die auch in der Freizeit viel mit neuen Technologien

<sup>82</sup>Hemmerich (2019), o. S.; Vgl. Hemmerich (2019), o. S.

<sup>83</sup>Hemmerich (2019), o. S.

<sup>84</sup>Ebd., o. S.

<sup>85</sup>Vgl. ebd., o. S.

<sup>86</sup>Ebd., o. S.

<sup>87</sup>Ebd., o. S.

<sup>88</sup>Ebd., o. S.

<sup>89</sup>Vgl. ebd., o. S.

<sup>90</sup>Ebd., o. S.

<sup>91</sup>Vgl. ebd., o. S.

<sup>92</sup>Vgl. ebd., o. S.

<sup>93</sup>Ebd., o. S.

<sup>94</sup>Vgl. ebd., o. S.

arbeiten“<sup>95</sup>, erwartet Hemmerich, dass ein CMDB Chatbot „in dem Bereich sehr großen Anklang finden wird“<sup>96, 97</sup>.

**Transparenz, Arbeitsüberwachung:** Die ChatOps-inherente Transparenz befürwortet Hemmerich grundsätzlich. Er gibt jedoch zu bedenken, dass man darin eine Form von Arbeitsüberwachung sehen kann. In einem transparenten Chat Raum sei es denkbar, dass man nachvollziehen kann, woran und wie intensiv andere Kollegen arbeiten, deshalb solle man diesbezüglich Rücksprache mit dem Betriebsrat halten.

Seitens der Kollegen rechnet er mit Zustimmung und Offenheit für diese Arbeitsweise.<sup>98</sup>

**Framework Python, Weiterentwicklung:** Bei der Wahl des Bot-Frameworks legt Hemmerich Wert auf die Wartbarkeit und Möglichkeit der Weiterentwicklung. Aufgrund des fundierten Python Know-Hows im Rechenzentrum würde er einen Bot in dieser Sprache bevorzugen. Er empfiehlt, „Entwicklungen in den Sprachen zu betreiben, wo das Know-How da ist und auch in Zukunft da sein wird. Denn es gibt nichts Schlimmeres als eine Anwendung, die ein Praktikant oder Azubi entwickelt hat in einer Sprache, mit der sich hier keiner auskennt und die nach seinem Weggang eben nicht mehr wartbar ist.“<sup>99</sup>

**Charakter, Avatar:** Bei der Modellierung der Persönlichkeit des CMDB Chatbots verweist Hemmerich auf die Professionalität des Rechenzentrumsbetriebs und schlägt einen dementsprechenden Charakter vor. Da das Unternehmen im Medienbereich tätig ist, könnte er sich Maus oder Elefant aus der Sendung mit der Maus als Avatar vorstellen. Da im Alltag eine humorvolle Stimmung herrscht, würde er einen solchen Charakterzug befürworten.<sup>100</sup>

**Betriebssicherheit:** „[I]m Hinblick auf die Betriebssicherheit“<sup>101</sup> zieht Hemmerich einen regelbasierten Chatbot einem KI-gestützten vor, da das Regelwerk klar definiert werden kann und keine ungeplanten oder unbeabsichtigten Aktionen ausgeführt werden, die im schlimmsten Fall zu „einer Katastrophe oder Störung“<sup>102</sup> führen können.<sup>103</sup>

**Potential:** Auch für andere Tätigkeiten des Rechenzentrums sieht Hemmerich „großes Potential“<sup>104</sup> in der Verwendung von ChatOps. Er warnt davor, sich vor einer solchen Technologie zu fürchten und sie als Bedrohung des eigenen Arbeitsplatzes zu sehen. Stattdessen sei sie eine „sinnvolle Ergänzung der täglichen Arbeit“<sup>105</sup>. Auch für andere Unternehmen kann er sich eine ChatOps Lösung vorstellen. „Jeder der eine CMDB einsetzt, kann von so etwas nur profitieren.“<sup>106</sup>

#### 3.4.2.2 Interview Middendorf

**CMDB allgemein:** Als nächstes soll das Interview mit Jörg Middendorf betrachtet werden. Unter einer CMDB versteht er eine Datenbank, in der „alle Gerätschaften eines Unternehmens dokumentiert sind und in Relation gesetzt werden“.<sup>107</sup> Z. B. werden „Server, Lizenzen und Aufstellungsorte“<sup>108</sup> dokumentiert, um sie für Supportfälle oder Planungstätigkeiten zu nutzen.<sup>109</sup>

---

<sup>95</sup>Hemmerich (2019), o. S.

<sup>96</sup>Ebd., o. S.

<sup>97</sup>Vgl. ebd., o. S.

<sup>98</sup>Vgl. ebd., o. S.

<sup>99</sup>Hemmerich (2019), o. S.; Vgl. Hemmerich (2019), o. S.

<sup>100</sup>Vgl. Hemmerich (2019), o. S.

<sup>101</sup>Ebd., o. S.

<sup>102</sup>Ebd., o. S.

<sup>103</sup>Vgl. ebd., o. S.

<sup>104</sup>Ebd., o. S.

<sup>105</sup>Ebd., o. S.

<sup>106</sup>Hemmerich (2019), o. S.; Vgl. Hemmerich (2019), o. S.

<sup>107</sup>Middendorf (2019), o. S.

<sup>108</sup>Ebd., o. S.

<sup>109</sup>Vgl. ebd., o. S.

**Suchfunktion, Standortabfrage, Standortreport:** Bei der täglichen Arbeit nutzt Middendorf die CMDB vor Allem zur „Dokumentation neu erstellter Maschinen“<sup>110</sup> und um Objekte wie Server zu suchen und deren Standort zu überprüfen. Außerdem nutzt er die Reporting Funktion zur Unterstützung bei Planungen, z. B. um festzustellen, „wie viele Server an einem Standort ...sind“<sup>111</sup>. Durch ein aktuelles „Projekt, bei dem Rechenzentren umgezogen werden“<sup>112</sup> wird diese Funktionalität häufig benötigt. Abseits davon nutzt er die CMDB um IP-Netze zu dokumentieren und neue Adressen zu vergeben, bzw. nachzuprüfen, wofür bestehende genutzt werden.<sup>113</sup>

**CMDB Schnittstellen:** Besonders wichtig findet Middendorf die Schnittstellen der CMDB. Es gibt einerseits die Möglichkeit, aus dem Monitoring Daten abzufragen und andererseits die Möglichkeit, in „der CMDB geplante Daten ins Monitoring zu transferieren“<sup>114</sup>. Außerdem findet er die Reporting-Funktion für die tägliche Arbeit sehr wichtig.<sup>115</sup>

**ChatOps allgemein:** ChatOps versteht Middendorf im DevOps-Gedanken, also als Zusammenarbeit von Entwicklern und Administratoren. Mit Chatbots hatte er vor Allem schon bei Supportanbietern zu tun, die diese einsetzen, um „Anliegen aufzunehmen und zu klassifizieren“<sup>116</sup>. Auch mit Sprachassistenten, wie Siri oder Alexa hatte er bereits zu tun.<sup>117</sup>

**Softwarestand, IP-Adresse, Automatisierung:** Im Rechenzentrumsbereich sieht Middendorf für Chatbots vor allem Potential, Standardaufgaben zu übernehmen. Dazu zählt er die Suche nach dem Standort von Geräten und die Ermittlung von Softwareständen und IP-Adressen. Für diese Aufgaben werden bisher die RZ-Mitarbeiter persönlich oder per E-Mail gefragt. Von einer Automatisierung per Chatbot verspricht er sich für die Mitarbeiter eine Konzentration auf die eigentlichen Aufgaben des RZ-Betriebs.<sup>118</sup>

**Mobilität, IP-Adresse, Netzinformationen, Ressourcen:** Als Vorteil eines Chatbots zur Bedienung der CMDB sieht Middendorf vor Allem die Mobilität. Als konkretes Beispiel könne er sich hier vorstellen, bereits auf dem Weg zum Rechenzentrum nach einem Server zu suchen, um herausfinden, in welchem Rack und an welcher Position sich dieser befindet. Außerdem könne er vor Ort prüfen, welche Software der Maschine zugewiesen ist, welche IP sie hat, welchem Subnetz und VLAN sie angehört und wie viel Festplattenplatz und RAM belegt sind. Außerdem ergänzt er, dass die Abfrage per Webinterface auf Mobilgeräten möglicherweise gar nicht möglich sei.<sup>119</sup>

**Servereinbau, Serverausbau:** Als Möglichkeit zur Arbeitsvereinfachung sieht Middendorf einen CMDB Chatbot auch beim Einbau von neuer Hardware im Rechenzentrum. So könne er direkt den Einbauort vermerken, den Server benennen und ihm eine IP-Adresse zuweisen. Beim Ausbau eines Geräts könne er hingegen den Status des Geräts auf *außer Betrieb* oder *verschrottet* setzen.<sup>120</sup>

**Berechtigungskonzept, API Einsatz:** Nach Middendorfs Ansicht sollte nur ein bestimmter Personenkreis Zugriff auf die CMDB Features haben. Dies gelte vor Allem für die schreibenden Funktionen, für die Lesenden sei eine andere Regelung denkbar (z. B. Öffnung der Suche für alle). Als weitere Anforderung gibt er vor, die APIs oder Command Line Tools des CMDB Herstellers zu nutzen und nicht direkt auf die Datenbank zuzugreifen.<sup>121</sup>

---

<sup>110</sup> Middendorf (2019), o. S.

<sup>111</sup> Ebd., o. S.

<sup>112</sup> Ebd., o. S.

<sup>113</sup> Vgl. ebd., o. S.

<sup>114</sup> Ebd., o. S.

<sup>115</sup> Vgl. ebd., o. S.

<sup>116</sup> Ebd., o. S.

<sup>117</sup> Vgl. ebd., o. S.

<sup>118</sup> Vgl. ebd., o. S.

<sup>119</sup> Vgl. ebd., o. S.

<sup>120</sup> Vgl. ebd., o. S.

<sup>121</sup> Vgl. ebd., o. S.



**Nachvollziehbarkeit:** Middendorf ergänzt weiterhin, dass nachvollziehbar sein sollte, wer welche Änderung vorgenommen hat. Dies könne beispielsweise im Chatverlauf oder in einem Log File festgehalten werden und solle dauerhaft abrufbar sein. Im Idealfall werde diese Information direkt bei der Manipulation eines CIs als Kommentar vermerkt.<sup>122</sup>

**Arbeitsüberwachung:** Mit Problemen beim Einsatz eines CMDB Chatbots rechnet Middendorf grundsätzlich nicht, er gibt jedoch zu bedenken, dass eine Chat Software, die außerhalb eines Unternehmens gehostet wird und Unternehmensinterna darin kommuniziert werden, besondere Regularien erfüllen müsse. Die Transparenz, die mit ChatOps einhergeht, sieht Middendorf als Vorteil. Man könne direkt mitverfolgen, was sich in der Umgebung ändert oder sich nach Abwesenheit einen Überblick über die erfolgten Änderungen verschaffen. Allerdings solle diese Übersicht nicht zur Auswertung der Arbeitsleistung herangezogen werden.<sup>123</sup>

**Framework:** Bei der Wahl des Bot-Frameworks sieht Middendorf Errbot in Betrachtung der Weiterentwicklung als klaren Favoriten, da im Rechenzentrum bereits großes Python Know-How vorhanden und die Sprache auch für Laien recht leicht lesbar ist.<sup>124</sup>

**Charakter:** Als Persönlichkeit für einen Chatbot kann sich Middendorf keine konkrete Figur oder Person vorstellen. Er würde jedoch einen humorvollen Charakter bevorzugen. Es müsse aber darauf geachtet werden, dass die Aussagen des Bots nicht verletzend wirken. Bei mehrfachen Anfragen nach dem gleichen Inhalt oder Falscheingaben könne durchaus eine freche Antwort folgen.<sup>125</sup>

**Ansatz, Hilfe:** Bei der Wahl der Art des Chatbots spricht sich Middendorf klar für einen regelbasierten Ansatz aus. Der Bot solle nur genau das machen, was angefordert ist und nichts, das er durch künstliche Intelligenz dazu gelernt hat. Eine Interpretation solle nicht erfolgen, das Ergebnis könne dann zu stark von der Erwartung abweichen. Ein KI-gestützter Chatbot sei deshalb höchstens für andere Anwendungen, wie die Begrüßung neuer Mitarbeiter oder die Beantwortung der FAQ zu bevorzugen, nicht aber für Rechenzentrumstätigkeiten.

Da die RZ Mitarbeiter täglich auf Kommandozeilenebene arbeiten geht Middendorf davon aus, dass sie mit den Befehlen des Bots schnell zurecht kommen. Außerdem müsse dieser über eine Hilfe Funktion verfügen, welche die gültigen Kommandos mit Beispielen ausgibt. Eine Einführungsveranstaltung solle vor Inbetriebnahme des Bots erfolgen.<sup>126</sup>

**Nachfrage, Herstellerkonsolidierung:** Bei kritischen Aktionen hält Middendorf eine zusätzliche Abfrage für sinnvoll. Außerdem sei es bisher der Fall, dass der gleiche Hersteller unter unterschiedlichen Namen im System vermerkt ist (z. B. HP, Hewlett-Packard). Daher sei es sinnvoll, bei der Eintragung neuer Hardware, die bestehenden Hersteller zu berücksichtigen und bei Ähnlichkeit vorzuschlagen, um Dopplungen zu vermeiden. Dies solle interaktiv erfolgen.<sup>127</sup>

**Potential:** Für andere Rechenzentrumstätigkeiten sieht Middendorf ebenfalls Potential. Alle Dienste mit einer API seien zur Automatisierung gut geeignet. Er könne sich z. B. die Provisionierung von virtuellen Maschinen, die Anbindung an Puppet und die Bearbeitung von DNS Einträgen vorstellen. Auch für andere Unternehmen hält er ChatOps für sinnvoll. Vor Allem die Knappheit an qualifiziertem Personal spreche für mehr Automatisierung und ein Chatbot sei dafür eine sinnvolle Möglichkeit, da er bei der täglichen Arbeit die Komplexität von Workflows reduziert.<sup>128</sup>

---

<sup>122</sup>Vgl. Middendorf (2019), o. S.

<sup>123</sup>Vgl. ebd., o. S.

<sup>124</sup>Vgl. ebd., o. S.

<sup>125</sup>Vgl. ebd., o. S.

<sup>126</sup>Vgl. ebd., o. S.

<sup>127</sup>Vgl. ebd., o. S.

<sup>128</sup>Vgl. ebd., o. S.



### 3.4.2.3 Interview Kirsten

**CMDB allgemein:** Als nächstes soll das Interview mit Daniel Kirsten betrachtet werden. Er betrachtet in seiner Funktion als Produktmanager die CMDB streng nach ITIL und versteht darunter die zentrale Erfassung von IT-Infrastruktur Elementen und deren Konfiguration. Das Verständnis unterscheide sich dabei stark und die gespeicherten Daten reichten von Asset Management Informationen bis hin zu technischen Konfigurationen. Außerdem spreche man nach ITIL nicht mehr von einer CMDB, sondern eher von einem Configuration Management System (CMS), welches eine Vereinigung mehrerer CMDBs darstellt. Der Großteil der Anwender betrachte die CMDB als Datenbank, die „alle Komponenten einer IT-Infrastruktur vereint, dokumentiert und im Idealfall mindestens die technischen Grundkonfigurationen dieser Elemente aufnimmt.“<sup>129</sup>

**Interner Einsatz:** Bei der täglichen Arbeit nutzt Kirsten selbst die i-doit CMDB, um die internen technischen Komponenten zu dokumentieren. Sie diene als Nachschlagewerk und werde manuell, ohne Zuhilfenahme automatisierter Prozesse befüllt. Neben den internen Komponenten seien auch noch Cloud Ressourcen dort dokumentiert.<sup>130</sup>

**IP-Adressen, Passwörter, Cloud:** Die Erfassung von IP-Adressen und Passwörtern in der CMDB findet Kirsten besonders wichtig. Da vermehrt Cloud Lösungen eingesetzt werden, sei es zudem wichtig Zugang und verknüpfte Dienste zu diesen Ressourcen zu dokumentieren.<sup>131</sup>

**ChatOps, Chatbots, Output:** Mit ChatOps hatte Kirsten noch keine Berührungspunkte, im Grunde verstehe er es aber als DevOps Prozess, der durch Chatbots unterstützt wird. Beim Thema Chatbots kann er aus einiger Erfahrung schöpfen, da er fast zehn Jahre lang TCL Skripte für Eggdrop Bots programmierte. Einen Chatbot definiert er als Instanz, die User Input entgegennimmt, verschiedene Datenquellen nutzt und einen menschenlesbaren Output liefert.<sup>132</sup>

**Vorteile Chatbots:** Als Vorteil von Chatbots sieht Kirsten den schnellen Abruf von Informationen. Damit spare man sich z. B. das Öffnen des Browsers und die Anmeldung. Er warnt jedoch auch davor, zu viele Informationen von einem Chatbot ausgeben zu lassen. Für die gängigsten Funktionen einer CMDB kann Kirsten sich einen Chatbot gut vorstellen.<sup>133</sup>

**IP-Adressen, Standortdaten, Kontaktpersonen, Suche, Logbuch:** Da der Output von Chatbots auf Text limitiert ist, empfiehlt Kirsten, die Prozesse eines Chatbots nicht zu komplex zu gestalten. Als konkrete Anwendungsfälle eines CMDB Chatbots könne er sich die Abfrage von IP-Adressen, Standortdaten oder Kontaktpersonen vorstellen. Außerdem sei eine Suche sinnvoll, wenn man den kompletten Objektnamen nicht weiß. Zudem könne er sich vorstellen, freie IP-Adressen zu reservieren oder als Administrator Änderungen am Logbuch zu erfassen. Vor Allem die Ausgabe von Informationen hält Kirsten für leicht umsetzbar, die Eingabe stehe eher im Hintergrund, da die Normalisierung fehleranfällig ist.<sup>134</sup>

**Rechtesystem, Plattform-Agnostik:** Ein CMDB Chatbot soll laut Kirsten ein Rechtesystem abbilden, das nur den Nutzern Informationen darstellt, die berechtigt sind, diese zu sehen. Er sehe sonst ein Missbrauchspotential. Außerdem rät er dazu, einen Chatbot nur im internen Netz zu betreiben, um die CMDB nicht für das Internet zugänglich zu machen. Weiterhin rät er dazu, den Chatbot möglichst plattformagnostisch zu gestalten, also die Möglichkeit zu bieten, „verschiedene Chatsysteme daran anzubinden“<sup>135</sup>

---

<sup>129</sup>Kirsten (2019), o. S.; Vgl. Kirsten (2019), o. S.

<sup>130</sup>Vgl. Kirsten (2019), o. S.

<sup>131</sup>Vgl. ebd., o. S.

<sup>132</sup>Vgl. ebd., o. S.

<sup>133</sup>Vgl. ebd., o. S.

<sup>134</sup>Vgl. ebd., o. S.

<sup>135</sup>Ebd., o. S.

. Als Standard sehe er jedoch aktuell Slack.<sup>136</sup>

**Akzeptanz, Feedback:** Mit Problemen bei der Akzeptanz eines CMDB Chatbots rechnet Kirsten nicht. Es müsse einer die Funktionen in einem öffentlichen Raum nutzen, damit andere das Potential erkennen und es ihm gleich tun. So könne Viralität entstehen. Man erhalte als Entwickler jedoch kaum Feedback. Wenn der Bot nicht angenommen wird, spiegele es sich lediglich darin, dass er nicht genutzt wird. Generell ist Kirsten jedoch zuversichtlich bezüglich der Akzeptanz eines CMDB Chatbots.<sup>137</sup>

**Programmiersprache:** Wegen der großen Verbreitung, der Mächtigkeit und dem langen Bestehen, empfiehlt Kirsten als Programmiersprache Python. Bei neueren Technologien wie Ruby oder Coffeescript rate er eher zur Vorsicht.<sup>138</sup>

**Persönlichkeit, Avatar:** Dem CMDB Chatbot eine Persönlichkeit zu verleihen hält Kirsten nicht für notwendig. Da vorwiegend Techniker damit arbeiten sollen, seien Formalitäten wie Begrüßungen überflüssig. Als Avatar könne er sich jedoch einen Roboter vorstellen.<sup>139</sup>

**Ansatz:** Kirsten spricht sich für einen regelbasierten Chatbot aus, ein KI-gestützter Bot sei eher schwierig und führe zu Frustrationen beim Anlernen. Er empfiehlt die Konzentration auf „kleine, regelbasierte Aufgaben“<sup>140</sup>

**Potential:** Für andere Unternehmen sieht Kirsten großes Potential für die Verwendung eines CMDB Chatbots. Vor Allem durch die Verwendung von Slack und WhatsApp seien die Anwender mit dem Medium vertraut und können damit professionell umgehen. Schnittstellen dazu zu haben, halte er daher für sinnvoll.<sup>141</sup>

**Reaktivität:** Ergänzend warnt Kirsten davor, Funktionen zu nutzen, die automatisch Inhalte darstellen. Dadurch gingen Informationen verloren und niemand reagiere darauf. Daher solle ein CMDB Chatbot rein reaktiv arbeiten.<sup>142</sup>

#### 3.4.2.4 Interview Buck

**CMDB allgemein:** Als Letztes soll das Interview mit Konrad Buck betrachtet werden. Unter einer CMDB versteht er eine Datenbank, die dynamisch alle für den IT-Betrieb notwendigen Items aufnimmt und verwaltet, um einen effektiven Betrieb zu gewährleisten, Prozesse zu planen und Changes durchzuführen.<sup>143</sup>

**CMDB Funktionen:** Besonders wichtig findet Buck dabei die UI, also die grafische Darstellung von Netzwerken oder Räumen mit der Möglichkeit in einzelne Bereiche hereinzuzoomen. Heutzutage seien alle an Videos und „einfache bildliche Darstellung[en]“<sup>144</sup> gewöhnt, weshalb Listenanzeigen unpassend sind. Außerdem müsse die Benutzerführung gut sein und es solle die Möglichkeit bestehen, Wartungspläne, Notfallpläne, Arbeitsabläufe und eine Wissensdatenbank zu organisieren. Schnittstellen solle eine CMDB auch haben, um z. B. Ticketing und Inventarisierung anzubinden und einen Gesamtüberblick zu erlangen.<sup>145</sup>

**Chatbots, ChatOps:** Unter einem Chatbot versteht Buck ein Programm, das zur Automatisierung genutzt wird. Damit sei es möglich, Tätigkeiten wie die Kommunikation mit dem Herstellersupport oder

---

<sup>136</sup>Vgl. Kirsten (2019), o. S.

<sup>137</sup>Vgl. ebd., o. S.

<sup>138</sup>Vgl. ebd., o. S.

<sup>139</sup>Vgl. ebd., o. S.

<sup>140</sup>Kirsten (2019), o. S.; Vgl. Kirsten (2019), o. S.

<sup>141</sup>Vgl. Kirsten (2019), o. S.

<sup>142</sup>Vgl. ebd., o. S.

<sup>143</sup>Vgl. Buck (2019), o. S.

<sup>144</sup>Ebd., o. S.

<sup>145</sup>Vgl. ebd., o. S.

Service-Management abzulösen und vorgefasste Antworten auf häufige Fragen zu entgegnen. Mit Chat-Ops hat Buck bisher noch keine Erfahrungen.<sup>146</sup>

**Kontakt Chatbots:** Persönlich hatte Buck bisher nur in Warteschleifen von Hotlines Kontakt mit Chatbots, die das Anliegen erfassen, kategorisieren und an einen echten Mitarbeiter weiterleiten.<sup>147</sup>

**Standardanfragen, Vorhersage, Sprachassistenten:** Laut Buck könnte ein CMDB Chatbot sämtliche Standardanfragen, die ein Administrator sonst übernimmt bearbeiten. Der Bot müsste dafür an eine Wissensdatenbank mit entsprechenden Fragen angebunden werden. Als weiteres Feature betrachtet Buck die Vorhersage von Zuständen, also z. B. der Ressourcenbelegung von Servern, um Aktionen zu planen. Außerdem sehe er Sprachassistenten als mögliches Interface, da diese in der Lage seien Emotionalität in der Stimme zu erkennen. In der Verbindung mit dem Profil des Nutzers könne so die Intention besser verstanden werden. Er sieht in einem CMDB Chatbot die Chance, Workflows zu vereinfachen, eine Zeitersparnis zu erzielen und die Akzeptanz bei Endanwendern zu verbessern.<sup>148</sup>

**Vertrauen, Gemütszustand:** Buck vermutet, dass die Benutzer der CMDB anderen Menschen mehr vertrauen als einer Maschine, was zu einer Skepsis bei der Verwendung von Chatbots führen könnte. Beim Thema Sicherheit hingegen hat er keine Bedenken, es handele sich lediglich um eine Verschiebung „der Schnittstelle zwischen Mensch und Maschine“<sup>149</sup>. Er verspricht sich sogar einen Gewinn an Sicherheit, da ein sprachbasierter Chatbots anhand der Stimme Müdigkeit oder Desinteresse erkennen und eine Fehlbedienung vermeiden könne.<sup>150</sup>

**Akzeptanz:** Buck rechnet damit, dass Chatbots einen schweren Einstieg haben werden, da sie oftmals nur aus Telefonwarteschleifen bekannt sind. Davon müssten sich professionelle Chatbots abheben.<sup>151</sup>

**Avatar, Persönlichkeit:** Als Avatar für einen CMDB Chatbot kann Buck sich *Dschinni* vorstellen und als Persönlichkeit eine Mischung aus Professionalität und Lockerheit. Ein Chatbot müsse die notwendige Professionalität ausstrahlen, damit ein Administrator seine Arbeit erledigen kann.<sup>152</sup>

**Ansatz:** Bei der Wahl des Chatbots spricht sich Buck klar für eine KI-gestützte Variante aus. Diese sei insbesondere bei der Verwendung von Stimmerkennung vorteilhaft. Allerdings merkt er auch an, dass die KI-Unterstützung recht kostspielig sein kann und deshalb für kleinere Unternehmen eher nicht in Frage kommt. Die Gefahr von Fehlinterpretationen durch KI-Systeme schätzt Buck eher gering ein. Er setze voraus, dass die Anbieter solcher Lösungen dies im Griff haben.<sup>153</sup>

**Potential:** Auch für andere Unternehmen sieht Buck Potential für die Verwendung eines CMDB Chatbots. Vor Allem die Neuheit des Themas und die mögliche Arbeitserleichterung sei dabei ausschlaggebend. Über das Thema KI werde häufig gesprochen, allerdings gebe es kaum konkrete Anwendungen dafür. Dann müsse aber auch eine entsprechende Plausibilitätsprüfung durchgeführt und Sicherheit gewährleistet werden.<sup>154</sup>

### 3.4.3 (Bildung von) Überschriften

Die paraphrasierten Abschnitte werden möglichst textnah mit einer oder mehreren Überschriften versehen, je nach dem, wie viele Themen angesprochen werden. Dabei ist darauf zu achten, dass das Exper-

---

<sup>146</sup>Vgl. Buck (2019), o. S.

<sup>147</sup>Vgl. ebd., o. S.

<sup>148</sup>Vgl. ebd., o. S.

<sup>149</sup>Ebd., o. S.

<sup>150</sup>Vgl. ebd., o. S.

<sup>151</sup>Vgl. ebd., o. S.

<sup>152</sup>Vgl. ebd., o. S.

<sup>153</sup>Vgl. ebd., o. S.

<sup>154</sup>Vgl. ebd., o. S.

tenwissen im Fokus steht, der Experte als Person ist nicht relevant.<sup>155</sup>

Die Anforderungen aus den paraphrasierten Textpassagen werden im Folgenden in Tabelle 3.4, Tabelle 3.3, Tabelle 3.5 und Tabelle 3.6 pro Interview mit Überschriften versehen und in der Reihenfolge der Nennung tabellarisch dargestellt. Außerdem werden sie in funktionale Anforderungen und Qualitätsanforderungen eingeteilt.

Funktionale Anforderungen legen fest, welche Funktionalität ein System bieten soll. Sie können weiter in Funktions-, Verhaltens- und Strukturanforderungen unterteilt werden. In den Anforderungen werden diese mit einem *F* im Bezeichner vermerkt.<sup>156</sup>

Qualitätsanforderungen beschreiben die Qualitäten, die ein System erfüllen muss und beeinflussen die Systemarchitektur. Typische Qualitätsanforderungen sind z. B. Performance, Verfügbarkeit, Skalierbarkeit oder Portabilität. In den Anforderungen werden diese mit einem *Q* im Bezeichner vermerkt.<sup>157</sup>

Dabei werden die laut Pohl und Rupp vorgesehenen Spalten Autor und Quelle ausgelassen und nur Identifikator und Name aufgeführt, da sich die Anforderungen auf das jeweilige Experteninterview beziehen und die Quelle das Interviewprotokoll und der Autor der Interviewpartner ist.<sup>158</sup>

---

<sup>155</sup>Vgl. Meuser und Nagel (1991), S. 458 f.

<sup>156</sup>Vgl. Pohl und Rupp (2015), S. 8.

<sup>157</sup>Vgl. ebd., S. 9.

<sup>158</sup>Vgl. ebd., S. 126.

Bezeichner	Name	Anforderung
1-1-F	Bestandsprüfung	Bestand von Gegenständen im Lager (z. B. Kabel) soll geprüft werden.
1-2-F	Statusabfrage	Der Status von Servern (eingelagert, verschrottet) soll abgefragt werden.
1-3-F	Ressourcenübersicht	Es soll ein Report über die Ressourcen eines Kunden / einer Anwendung möglich sein.
1-4-F	Suchfunktion	Es soll eine Suche nach Objekten möglich sein.
1-5-F	Nächste freie IP	Es soll die nächste freie IP in einem Netz abgefragt werden können.
1-6-F	Serverplanung	Ein neuer Server soll mit Name und IP angelegt werden können.
1-7-Q	Schnittstellenanbindung	Bei der Entwicklung sollen zukünftige Schnittstellen zu Puppet und vCenter berücksichtigt werden.
1-8-Q	Vermeidung Fehlbedienung	Versehentliche Löschung soll vermieden werden.
1-9-Q	Berechtigungskonzept	Nur ein bestimmter Personenkreis soll Änderungen durchführen dürfen.
1-10-Q	Nachvollziehbarkeit	Es soll nachvollziehbar sein, wer Änderungen vorgenommen hat.
1-11-Q	Webinterface Eingaben	Eingaben über das Webinterface sollen weiterhin möglich sein.
1-12-Q	Arbeitsüberwachung	Eine Auswertung der Leistung Einzelner soll nicht erfolgen.
1-13-Q	Framework Python	Es soll ein Bot-Framework in der Sprache Python gewählt werden.
1-14-Q	Charakter	Der Charakter soll Professionalität ausdrücken, aber humorvoll sein.
1-15-Q	Avatar	Als Avatar soll Maus oder Elefant gewählt werden.
1-16-Q	Betriebssicherheit	Es soll ein regelbasierter Bot gewählt werden.

Quelle: Eigene Darstellung

Tabelle 3.3: Anforderungen Experteninterview Hemmerich

Bezeichner	Name	Anforderung
2-1-F	Suchfunktion	Eine Suche nach Objekten soll möglich sein.
2-2-F	Standortabfrage	Es soll der Standort von Objekten abrufbar sein.
2-3-F	Standortreport	Die Anzahl der Server an einem Standort soll abgefragt werden.
2-4-F	Softwarestand	Der Softwarestand für ein Gerät soll ausgegeben werden.
2-5-F	IP-Adresse	Die IP-Adresse eines Geräts soll ausgegeben werden.
2-6-F	Netzinformationen	VLAN und Subnetz eines Geräts sollen ausgegeben werden.
2-7-F	Ressourcen	HDD und RAM Belegung eines Geräts sollen angezeigt werden.
2-8-F	Serveereinbau	Beim Einbau eines Servers soll dieser mit Name, IP und Ort dokumentiert werden.
2-9-F	Serveerausbau	Beim Ausbau eines Servers soll dieser auf <i>außer Betrieb</i> oder <i>verschrottet</i> gesetzt werden.
2-10-Q	Berechtigungskonzept	Nur ein bestimmter Personenkreis soll Zugriff auf die schreiben- den Funktionen haben.
2-11-Q	API Einsatz	Es soll per API oder Client zugegriffen werden, kein direkter Datenbank-Zugriff.
2-12-Q	Nachvollziehbarkeit	Es soll nachvollziehbar sein, wer Änderungen vorgenommen hat.
2-13-Q	Arbeitsüberwachung	Die Arbeitsleistung soll nicht ausgewertet werden.
2-14-Q	Framework	Errbot soll als Bot Framework zum Einsatz kommen.
2-15-Q	Charakter	Der Charakter des Bots soll humorvoll sein.
2-16-Q	Ansatz	Der Chatbot soll regelbasiert sein.
2-17-Q	Hilfe	Der Bot soll über eine Hilfe Funktion mit Beispielen verfügen.
2-18-Q	Nachfrage	Bei kritischen Aktionen soll eine zusätzliche Abfrage zur Bestä- tigung erfolgen.
2-19-Q	Herstellerkonsolidierung	Bestehende Hersteller sollen berücksichtigt werden, um Dopp- lungen zu vermeiden.

Quelle: Eigene Darstellung

Tabelle 3.4: Anforderungen Experteninterview Middendorf

Bezeichner	Name	Anforderung
3-1-Q	Output	Die Ausgabe soll einfach menschenlesbar sein.
3-2-F	IP-Adresse	Es soll die IP-Adresse eines Objekts angezeigt werden.
3-3-F	Standortdaten	Es soll der Standort eines Objekts ausgegeben werden.
3-4-F	Kontaktperson	Es soll die Kontaktperson eines Objekts ausgegeben werden.
3-5-F	Suche	Es soll die Suche nach einem Objekt über den Namen erfolgen können.
3-6-F	Freie IP	Es soll die nächste freie IP in einem Netz abgefragt werden können.
3-7-F	Logbuch	Es sollen Logbucheinträge vorgenommen werden können.
3-8-Q	Rechtesystem	Nur Berechtigten sollen Informationen angezeigt werden.
3-9-Q	Plattform-Agnostik	Der Chatbot soll an verschiedene Chat Plattformen angebunden werden können.
3-10-Q	Programmiersprache	Als Programmiersprache soll Python verwendet werden.
3-11-Q	Persönlichkeit	Der Chatbot soll auf Formalitäten wie Begrüßungen verzichten.
3-12-Q	Avatar	Als Avatar soll ein Roboter verwendet werden.
3-13-Q	Ansatz	Es soll ein regelbasierter Einsatz gewählt werden.
3-14-Q	Reaktivität	Der Chatbot soll rein reaktiv arbeiten.

Quelle: Eigene Darstellung

Tabelle 3.5: Anforderungen Experteninterview Kirsten

Bezeichner	Name	Anforderung
4-1-Q	Standardanfragen	Chatbots sollen Standardanfragen bearbeiten, die sonst von Administratoren erledigt werden.
4-2-Q	Vorhersage	Die Ressourcenauslastung soll zur Planung vorhergesagt werden.
4-3-Q	Sprachassistenten	Anhand von Stimmlage und Benutzerprofil soll die Intention erkannt werden.
4-4-Q	Avatar	Dschinni soll als Avatar dienen.
4-5-Q	Persönlichkeit	Der Chatbot soll Professionalität und Lockerheit ausstrahlen.
4-6-Q	Ansatz	Es soll ein KI-basierter Ansatz gewählt werden.

Quelle: Eigene Darstellung

Tabelle 3.6: Anforderungen Experteninterview Buck

### 3.5 Validierung der Anforderungen

Die Validierung der Anforderungen ist der dritte und letzte Schritt des Anforderungsmanagements. Im Experteninterview wird darunter Schritt sechs, *thematischer Vergleich*, verstanden. Ab hier werden alle Interviews gemeinsam betrachtet und die identifizierten Anforderungen können unabhängig von den Interviewpartnern betrachtet werden. Es sollen Gemeinsamkeiten und Unterschiede der Expertenaussagen aufgezeigt werden.

Um die Qualität der Anforderungen gewährleisten zu können, muss diese frühzeitig geprüft werden.<sup>159</sup> Zu den Qualitätskriterien zählen u.A. Korrektheit und Abgestimmtheit. Die Methoden der Qualitätskontrolle können dabei sowohl für einzelne Anforderungen, als auch für Anforderungsdokumente eingesetzt werden. Bei der Überprüfung der Anforderungen sollen Fehler wie „Mehrdeutigkeit, Unvollständigkeit und Widersprüche“<sup>160</sup> aufgedeckt werden.<sup>161</sup>

Ab dem thematischen Vergleich werden nicht mehr die Interviews im Einzelnen betrachtet, es erfolgt ein Vergleich der thematisch ähnlichen Textpassagen aller Interviews, wobei die Überschriften nach Möglichkeit vereinheitlicht werden. Es wird geprüft, ob sich die Äußerungen der Experten decken oder ob sie unterschiedliche Ansichten haben. Außerdem wird betrachtet, zu welchen Themen sich alle Experten äußern und zu welchen nur einzelne.<sup>162</sup>

Die Interpretation erfolgt dabei auch nach Meuser und Nagel, die kein striktes Vorgehen beschreiben, sondern lediglich ein Modell vorgeben, das hinsichtlich der jeweiligen Untersuchung angepasst werden muss. Dabei wird die Bedeutung der Expertenaussagen kontextabhängig interpretiert und es werden thematische Blöcke aus den Textpassagen gebildet, die inhaltlich zusammengehören. Der berufliche Kontext, in dem die Experten agieren sollte bei der Interpretation auch beachtet werden.<sup>163</sup>

Im Gegensatz zu der inhaltlichen Strukturierung nach Mayring ist diese Interpretationsmethode zwar weniger umfangreich, jedoch auf das Experteninterview zugeschnitten.<sup>164</sup>

Zunächst werden die Anforderungen aufgeführt, die von mehreren Experten genannt wurden. Bei Übereinstimmung können diese zusammengeführt werden, bei Kontroversen müssen sie diskutiert werden. Auffällig bei der Dokumentation der Anforderungen war, dass die funktionalen Anforderungen nur von Hemmerich, Middendorf und Kirsten stammen, die technisch mit der CMDB vertraut sind. Buck hingegen, der eher organisatorisches Wissen besitzt, nannte nur Qualitätsanforderungen. Die meisten funktionalen Anforderungen nannten Middendorf und Hemmerich, vermutlich da sie täglich mit der CMDB arbeiten.

---

<sup>159</sup>Vgl. Pohl und Rupp (2015), S. 4.

<sup>160</sup>Ebd., S. 95.

<sup>161</sup>Vgl. ebd., S. 95.

<sup>162</sup>Vgl. Meuser und Nagel (1991), S. 459 f.; Vgl. Matthes-Nagel (1986), S. 37.

<sup>163</sup>Vgl. Meuser und Nagel (1991), S. 452 f.

<sup>164</sup>Vgl. Mayring (2015), S. 103 f.



### 3.5.1 Qualitätsanforderungen

**Charakter:** Der Charakter des Bots soll Professionalität ausdrücken, aber humorvoll und locker sein. Auf Formalitäten, z. B. bei der Begrüßung soll verzichtet werden (1-14-Q, 2-15-Q, 3-11-Q, 4-5-Q).

**Ansatz:** Als Ansatz für einen Chatbot haben sich drei Experten für einen regelbasierten Bot ausgesprochen (1-16-Q, 2-16-Q, 3-13-Q) und einer für KI-Unterstützung (4-6-Q).

Hemmerich, Middendorf und Kirsten, die sich für den regelbasierten Ansatz aussprachen, haben eher betriebliches Wissen und arbeiten selbst mit der CMDB, weswegen sie vor Allem ein Augenmerk auf den Erhalt der Betriebssicherheit legen, die besser mit einem regelbasierten Chatbot gewährleistet werden kann. Bei KI-Unterstützung befürchten sie hingegen Spielraum für Fehlinterpretationen und lange Anlernprozesse. Zudem gehen sie davon aus, dass Mitarbeiter, die mit der Kommandozeile vertraut sind schnell mit der Befehlssyntax eines regelbasierten Bots zurechtkommen werden. Als PR-Manager versteht Buck zwar die Konzepte der CMDB, hat jedoch eher eine organisatorische Perspektive und sieht in der neuen Technologie KI vor Allem Chancen.

Aus betrieblicher Sicht kommt also nur der regelbasierte Ansatz in Frage und die Zahl der Experten, die sich für diesen aussprechen überwiegt.

**Berechtigungskonzept:** Drei der vier Experten nannten die Anforderung eines Berechtigungskonzepts. Dabei sollen nur berechtigte Personen Informationen anzeigen und Änderungen vornehmen dürfen (1-9-Q, 2-10-Q, 3-8-Q).

**Bot Framework:** 3 Experten forderten, dass die Programmiersprache Python sein soll, bzw. das konkrete Bot Framework Errbot (1-13-Q, 2-14-Q, 3-10-Q).

**Avatar:** Einen Avatar für den Chatbot konnten sich drei der vier Experten vorstellen: Die Maus (aus der Sendung mit der Maus), Dschinni<sup>165</sup> und einen Roboter (1-15-Q, 3-12-Q, 4-4-Q). Die von Hemmerich genannte Maus entstammt vermutlich dem Gedanken, dass der Bot im IVZ entwickelt wird, das am Standort Köln dem WDR nahe steht, der die Sendung mit der Maus produziert. Da der Bot aber nicht für ein spezielles Unternehmen entwickelt wird, sondern ein generelles Interface zur i-doit CMDB geschaffen werden soll, macht ein allgemeingültiger Avatar mehr Sinn.

Die Idee, einen Dschinni zu benutzen ist nachvollziehbar, da er seinem Besitzer Wünsche erfüllt, ähnlich wie ein Chatbot Befehle ausführt, die ein Benutzer *wünscht*. Ein Roboter hingegen unterstreicht, dass es sich um eine Maschine, bzw. ein Programm handelt und keinen Menschen mit dem man sich unterhalten kann, daher ist die Verwendung eines Roboters als Avatar treffender für einen CMDB Chatbot.<sup>166</sup>

**Vermeidung Fehlbedienung:** Eine versehentliche Fehlbedienung soll bei kritischen Aktionen, wie der Löschung von Objekten vermieden werden. Dazu soll eine Abfrage erfolgen (1-8-Q, 2-18-Q).

**Nachvollziehbarkeit:** Es soll nachvollziehbar sein, wer eine Änderung per Bot vorgenommen hat (1-10-Q, 2-12-Q).

**Arbeitsüberwachung:** Die Arbeitsleistung Einzelner soll nicht über den Bot ausgewertet werden (1-12-Q, 2-13-Q). Der Arbeitgeber ist grundsätzlich befugt, die Einhaltung der von ihm erteilten Weisungen zu überprüfen und somit Leistung und Verhalten des Arbeitnehmers zu kontrollieren. Eingeschränkt wird er allerdings durch rechtliche Beschränkungen, wie Betriebsvereinbarungen, die Compliance Regeln aufstellen oder das Bundesdatenschutzgesetz, das die Verarbeitung personenbezogener Daten nur bei Einwilligung gestattet.<sup>167</sup>

Im Kontext des IT-Nutzungsverhaltens wird unter Arbeitsüberwachung bzw. Arbeitnehmerüberwa-

<sup>165</sup>Dschinni ist eine magische Figur in Aladin und die Wunderlampe aus den Märchen aus 1001 Nacht

<sup>166</sup>Vgl. Westdeutscher Rundfunk Köln (2019), o. S.

<sup>167</sup>Vgl. Rudkowski und Schreiber (2015), S. 1 ff.

chung die Kontrolle des Verhaltens verstanden, also welche Websites besucht werden, wie lange die Verweildauer ist, an welchen Dokumenten in welchem Umfang gearbeitet wird und welche Fenster geöffnet sind (z. B. durch regelmäßige Screenshots).<sup>168</sup>

**Schnittstellenanbindung:** Bei der Entwicklung sollen zukünftige Schnittstellen zu Puppet und vCenter berücksichtigt werden (1-7-Q).

**Webinterface-Eingaben:** Eingaben über das Webinterface der CMDB sollen weiterhin möglich sein (1-11-Q).

**API Einsatz:** Es soll per API oder Client auf die CMDB zugegriffen werden, nicht direkt über die Datenbank (2-11-Q).

**Hilfe:** Zu den Befehlen des Bots soll eine Hilfe abrufbar sein und es sollen Beispiele angeführt werden (2-17-Q).

**Herstellerkonsolidierung:** Bestehende Hersteller sollen beim Anlegen von Objekten berücksichtigt werden, um Dopplungen zu vermeiden (2-19-Q).

**Formatierung des Outputs:** Die Ausgaben des Bots sollen einfach menschenlesbar sein (3-1-Q). Im Sinne der Software-Ergonomie nach Stary beeinflussen Abstand des Nutzers vom Bildschirm, Helligkeit, Gestalt und Zwischenräume der Zeichen, Überschriften und Struktur, sowie Farbe die Lesbarkeit von Text. Je nach Framework können bei der Entwicklung eines Chatbots Überschriften, Struktur und Farbe für eine bessere Lesbarkeit der Ausgabe verwendet werden. Die restlichen Faktoren werden von Nutzer, Hardware oder Chat Software vorgegeben.<sup>169</sup>

**Plattform-Agnostik:** Der Bot soll an verschiedene Chat Plattformen angebunden werden können (3-9-Q).

**Reaktivität:** Der Bot soll rein reaktiv arbeiten, also nicht ohne Aufforderung Nachrichten posten oder Aktionen ausführen (3-14-Q).

**Standardanfragen:** Der Bot soll Standardanfragen beantworten, die sonst von Administratoren erledigt werden (4-1-Q).

**Vorhersage:** Die Ressourcenauslastung soll zur Planung vorhergesagt werden (4-2-Q).

**Sprachassistenten:** Anhand von Stimmlage und Benutzerprofil soll die Intention des Nutzers erkannt werden (4-3-Q). Diese Anforderung steht in Konflikt zu den Anforderungen *Python* und *Ansatz*. Die Verwendung von Errbot sieht eine rein textbasierte Eingabe vor, während die Erkennung der Stimmlage eine Spracheingabe und künstliche Intelligenz voraussetzt. Diese Anforderung kann daher nicht umgesetzt werden.

<sup>168</sup>Vgl. Rudkowski und Schreiber (2015), S. 55.

<sup>169</sup>Vgl. Stary (1996), S. 66 f.

### 3.5.2 Funktionale Anforderungen

**Suchfunktion:** In drei der vier Interviews wurde die Anforderung genannt, eine Suchfunktion zu bieten. Diese soll mit dem Objektnamen oder dem Teil eines Objektnamens erfolgen können (1-4-F, 2-1-F, 3-5-F).

**Nächste freie IP:** Die nächste freie IP-Adresse für ein Netz soll ausgegeben werden (1-5-F, 3-6-F).

**Serverplanung:** Bei der Serverplanung sollen Name, IP und Ort dokumentiert werden. (1-6-F, 2-8-F).

**Standortabfrage:** Der Standort eines Objekts soll dargestellt werden (2-2-F, 3-3-F).

**IP-Adresse:** Die IP-Adresse eines Objekts soll abgebildet werden (2-5-F, 3-2-F).

**Bestandsprüfung:** Der Bestand von Objekten im Lager (z. B. Kabel) soll geprüft werden können (1-1-F).

**Statusabfrage:** Der Status eines Objekts (eingelagert, verschrottet, etc.) soll abrufbar sein (1-2-F).

**Ressourcenübersicht:** Eine Übersicht über die Ressourcen eines Kunden oder einer Anwendung soll angezeigt werden (1-3-F).

**Standortreport:** Ein Report über die Server an einem Standort soll generiert werden können (2-3-F).

**Softwarestand:** Der Softwarestand eines Geräts soll ausgegeben werden (2-4-F).

**Netzinformationen:** VLAN und Subnetz eines Geräts sollen ausgegeben werden (2-6-F).

**Ressourcen:** Die Ressourcenauslastung (HDD, RAM) eines Geräts soll zurückgeliefert werden (2-7-F).

**Statuswechsel:** Beim Ausbau eines Servers soll der Status auf *außer Betrieb* oder *verschrottet* gesetzt werden können (2-9-F).

**Kontaktperson:** Zu einem Objekt soll die Kontaktperson ausgegeben werden (3-4-F).

**Logbuch:** Es sollen Logbucheinträge vorgenommen werden können (3-7-F). Im Logbuch werden Änderungen für Objekte und Services festgehalten. Dies geschieht automatisch bei Aktionen, es können aber auch eigene Einträge für ein Objekt ergänzt werden.<sup>170</sup>

---

<sup>170</sup>Vgl. synetics (2019c), o. S.

## Kapitel 4

# Entwurf eines Chatbots für eine CMDB

Dieses Kapitel hat zum Ziel, die in Abschnitt 3.5 erarbeiteten Anforderungen anhand eines rudimentären Chatbots, der an eine *i-doit* CMDB des Herstellers *synetics* angebunden ist, umzusetzen oder deren mögliche Umsetzung zu beschreiben. Dabei wird zunächst die Umsetzung der Qualitätsanforderungen erläutert, da diese sich stärker auf die Systemarchitektur auswirken und die Grundlage für funktionale Anforderungen bilden.<sup>171</sup>

Das Resultat dieses Kapitels, ein Errbot Plugin, ist auf GitHub verfügbar.<sup>172</sup>

### 4.1 Qualitätsanforderungen

Auch unter den Qualitätsanforderungen gibt es Abhängigkeiten, weshalb diese aufeinander aufbauend erläutert werden. Angefangen mit der Wahl des Ansatzes und Bot-Frameworks, sowie Gewährleistung von Plattform-Agnostik, Reaktivität und der Möglichkeit, weitere Schnittstellen anzubinden, hin zum Berechtigungskonzept, API-Einsatz und Formatierung des Outputs und schlussendlich zu Anforderungen ohne Abhängigkeiten.

#### 4.1.1 Ansatz, Bot-Framework, Plattform-Agnostik, Schnittstellenanbindung

Da sich das Bot-Framework *Errbot* als Anforderung herausgebildet hat und es ein grundlegendes Element ist, wird es zuerst betrachtet.

Für die Installation von Errbot werden verschiedene Methoden angeboten: Als System-Paket, systemweit mit dem Python Paketmanager *pip* oder in einer virtuellen Python Umgebung, einem sogenannten *virtualenv*. Da letztere Variante die präferierte ist, wird diese durchgeführt. Die Installation erfolgt wie in Listing 4.1 beschrieben.<sup>173</sup>

Listing 4.1: Virtualenv Einrichtung und Errbot Installation

```
1 mkvirtualenv -p 'which python3' errbot-ve
2 pip install errbot
```

<sup>171</sup>Vgl. Pohl und Rupp (2015), S. 9.

<sup>172</sup>Vgl. Weiss (2019), o. S.

<sup>173</sup>Vgl. errbotio (2018f), o. S.

Mit *errbot -init* wird ein initiales Projekt angelegt, das bereits ein Beispiel Plugin enthält. Die Struktur ist in Listing 4.2 ersichtlich. In der Datei *config.py* werden Konfigurationen, wie das verwendete Backend, das Log File, die Bot Administratoren und die Bot Identität vorgenommen.<sup>174</sup>

Listing 4.2: Struktur eines Errbot Projektes

```
1 config.py
2 data
3 plugins
4     err-example
5         example.plug
6         example.py
```

Mit der Grundeinrichtung von Errbot kann die Anforderung *Errbot* als erfüllt betrachtet werden. Für die CMDB Anbindung wird ein eigenes Plugin anhand des Beispiels *err-example* angelegt. In der *.plug* Datei werden Beschreibung und Kompatibilität (Python 2, 3, 2+) eingetragen und in der *.py* die Kommandos, die vom Bot aufrufbar sind. Weitere Schnittstellen, z. B. zu vCenter oder Monitoring können zukünftig per eigenem Plugin realisiert und vom selben Bot bedient werden. Somit gilt die Anforderung der Schnittstellenanbindung als erfüllt.<sup>175</sup>

Errbot kann standardmäßig an die Backends XMPP, IRC, HipChat, Slack und Telegram angebunden werden. Außerdem können Backends von Drittanbietern, wie z. B. Cisco WebEx Teams angebunden werden. Diese wurden von Usern erstellt und nicht von errbotio selbst und können in der Konfiguration mit *BOT\_EXTRA\_BACKEND\_DIR* unter Angabe des entsprechenden Ordners verwendet werden. In dieser Umsetzung wird Slack als Backend verwendet, dazu wird in der Konfiguration unter *BOT\_IDENTITY* der Bot Token hinterlegt und die Proxy Konfiguration für die Verwendung im internen Firmennetz gesetzt. Durch die verschiedenen Backends kann ein Errbot Plugin an eine Vielzahl von Chat Plattformen angebunden werden und die Anforderung der Plattform-Agnostik ist somit erfüllt.<sup>176</sup>

Ob das Plugin regelbasiert oder KI-gestützt umgesetzt wird, ist allein dessen Autor überlassen. Das CMDB Plugin wird wie gefordert regelbasiert umgesetzt und die Anforderung kann als erfüllt angesehen werden. Im Detail schlägt sich dies in den funktionalen Anforderungen nieder.

#### 4.1.2 Berechtigungskonzept

Errbot bietet die Möglichkeit, mit sogenannten *Access controls* den Zugriff auf bestimmte Plugins oder Befehle zu beschränken. Dabei können Benutzer oder Räume explizit erlaubt oder verboten werden. Außerdem können Befehle in Räumen generell verboten werden (allowmuc), d. h. der Befehl kann dem Bot nur per Privatnachricht geschickt werden, was bei Befehlen mit langer Ausgabe sinnvoll sein kann. Wie in Listing 4.3 zu erkennen ist, wurde das CMDB Modul nur im Channel *cbottest* erlaubt und das *render\_test* Kommando ist nur per privater Nachricht aufrufbar.<sup>177</sup>

<sup>174</sup>Vgl. errbotio (2018f), o. S.

<sup>175</sup>Vgl. errbotio (2018e), o. S.

<sup>176</sup>Vgl. errbotio (2018f), o. S.; Vgl. Sullivan (2018), o. S.

<sup>177</sup>Vgl. errbotio (2018a), o. S.

Listing 4.3: Errbot Access controls

```
1 ACCESS_CONTROLS = {'CMDB:*': {'allowrooms': ('#cbottest',)},  
2                       'render_test': {'allowmuc': False},  
3                       }
```

### 4.1.3 Reaktivität

Durch die Architektur ist in Errbot *by-design* sichergestellt, dass er nur auf Nennungen und Befehle reagiert. Selbstständiges Erzeugen von Nachrichten ist nicht vorgesehen.<sup>178</sup>

### 4.1.4 API-Einsatz

Mit dem Tool `i-doit-cli` können von der Kommandozeile aus verschiedene CMDB Funktionen, wie lesen, schreiben oder suchen ausgeführt werden. Außerdem können individuelle API Aufrufe damit abgesetzt werden. Mit dem `subprocess` Modul kann das Tool einfach in Python ausgeführt und die Ausgabe ausgewertet werden, dabei werden die Parameter `-no-colors` und `-y` genutzt, um nicht interaktiv zu arbeiten und keine Farben auszugeben. So werden mögliche Kompatibilitätsprobleme vermieden.<sup>179</sup>

---

<sup>178</sup>Vgl. errbotio (2018c), o. S.

<sup>179</sup>Vgl. Heisig (2019), o. S.

### 4.1.5 Formatierung des Outputs

Die Ausgabe des i-doit-cli Tools besteht aus reinem Text. Zur besseren Lesbarkeit werden für verschiedene Ausgaben, wie die Liste der Suchergebnisse oder die Detailansicht Funktionen zur Formatierung angelegt (Listing 4.4). Für die Suchergebnisse (searchformatter) wird die Ausgabe am doppelten Zeilenumbruch getrennt, um die Suchergebnisse zu separieren und dann die Bestandteile des Ergebnisses zu formatieren. Dem Namen des Objekts wird eine URL zu den Details in der CMDB hinterlegt und der Ort wird in Klammern dahinter dargestellt.

Für die Detailanzeige eines Objekts (nonemptyformatter) werden nur Einträge angezeigt, die einen Inhalt haben. Ansonsten wird die Ausgabe nicht verändert.

Listing 4.4: Funktionen zur Formatierung

```
1 def searchformatter(objectliststring):
2     """ format search result with links """
3     if objectliststring.split('\n\n')[0] == '':
4         return 'Keine_Ergebnisse'
5     itemlist = [i.split('\n') for i in objectliststring.split('\n\n
    ↳ ')] # get single lines for every search result item
6     items = [{ 'name': i[0], 'place': i[1].replace('Source:_', ''), '
    ↳ url': i[2].replace('Link:_', '') } for i in itemlist] #
    ↳ save information to dict and remove description in text
7     return '\n'.join(["'<{url}>|{name}>_({place})'\n'".format
    ↳ (**item) for item in items])
8
9
10 def nonemptyformatter(infostring):
11     """ return only fields with content for show output """
12     lines = infostring.split('\n')
13     result = []
14     for line in lines:
15         if not line.endswith(':_-'):
16             result.append(line)
17     return '\n'.join(result)
```

### 4.1.6 Charakter

Um die Professionalität sicherzustellen, werden nur die angeforderten Informationen ohne Umschweife ausgegeben. Wenn eine Ausgabe von Text nötig ist, geschieht dies jedoch mit Humor und Lockerheit. Wird z. B. ein benötigter Parameter vergessen, antwortet der Bot mit *Was soll ich denn damit anfangen? :man-facepalming: Da musst du schon mehr eingeben ....* Bei einem Fehler entgegnet der Bot *Da ist wohl was schief gelaufen:* und gibt anschließend die technische Fehlermeldung aus. Wenn eine Suche nichts zurückliefert, wird hingegen recht sachlich mit *Keine Ergebnisse* geantwortet.

#### 4.1.7 Avatar

Avatar und Name eines Bots werden beim Slack Backend bei der Erstellung des Bot Users vergeben und können nicht für ein Errbot Plugin vorgegeben werden. Der erhaltene Bot Token wird in der Errbot Konfiguration hinterlegt. Jedes Unternehmen, das eine Errbot Instanz installiert um das CMDDB Plugin zu nutzen, kann also selbst entscheiden, welcher Avatar genutzt werden soll. Für das IVZ wird daher die Maus aus der Sendung mit der Maus genutzt.<sup>180</sup>

#### 4.1.8 Nachvollziehbarkeit

Da die schreibenden Funktionen, also Serverplanung, Statuswechsel und Logbuch als Funktionsuser vorgenommen werden, muss zusätzlich vermerkt werden, wer diese initiiert hat. Dazu bietet sich die Funktion *idoitcli log* an, mit der ein Eintrag für ein Objekt im Logbuch hinzugefügt werden kann (Listing 4.5).

Listing 4.5: Hinzufügen eines Logbuch Eintrages

```
1  idoitcli log mylittleserver -m "created_by_username_via_Errbot"
```

#### 4.1.9 Vermeidung Fehlbedienung

Um Fehlbedienung zu vermeiden, sollte für kritische Kommandos wie Löschung oder Archivierung eine Bestätigung erfolgen, nachdem der Nutzer gefragt wurde, ob seine Daten korrekt sind. Dazu ist es nötig, die Parameter des initialen Aufrufs zwischenspeichern, um sie im Falle der Bestätigung weiter verarbeiten zu können und bei einem Abbruch zu verwerfen. Errbot bietet die Möglichkeit, solche Variablen persistent, auch über einen Neustart hinaus, zu speichern. So können die offenen Änderungen pro User vermerkt werden.

Nach der Abfrage müsste dann vom gleichen User der Befehl *!yes* zur Bestätigung oder *!no* zum Abbruch folgen. Im Rahmen dieser Arbeit ist keine Funktion vorgesehen, die eine solche Prüfung benötigt, ein möglicher Dialog bei der Archivierung eines Objekts könnte aber so aussehen, wie in Listing 4.6.<sup>181</sup>

Listing 4.6: Nachfrage bei der Archivierung von Objekten

```
1  Benutzer: !archive mylittleserver
2  Bot: Soll "mylittleserver" wirklich archiviert werden?
3  Benutzer: !yes
```

#### 4.1.10 Herstellerkonsolidierung

Um doppelte Herstellernamen zu vermeiden kann entweder vor dem Anlegen eines neuen Objekts eine Suche nach dem Namen erfolgen oder es kann nach der Eingabe des Namens automatisch auf Ähnlichkeit zu anderen Namen geprüft werden. In der ersten Version des CMDDB Chatbots wird auf die Ähnlichkeitsprüfung verzichtet, da dies auch mit einer zuvor getätigten Suche sichergestellt werden kann.

<sup>180</sup>Vgl. Slack (2019), o. S.

<sup>181</sup>Vgl. errbotio (2018d), o. S.



Es muss zudem definiert werden, welche Ähnlichkeiten geprüft werden sollen (Beginn, Ende) und welche Hersteller trotz fehlender Ähnlichkeit im Namen zusammengefasst werden sollen (Umbenennung, Zusammenschluss).

#### 4.1.11 Arbeitsüberwachung

Bei der Entwicklung des Plugins werden keine Funktionen umgesetzt, mit denen sich Aktivität und Leistung automatisiert personenbezogen messen lassen.

#### 4.1.12 Webinterface-Eingaben

Die Bedienung der CMDB per Chatbot bietet lediglich eine Ergänzung der Eingabemethoden. Das Webinterface der CMDB kann weiterhin parallel genutzt werden.

#### 4.1.13 Hilfe

Als Hilfe zu den verschiedenen Funktionen von Errbot wird standardmäßig die Dokumentation der hinterlegten Funktion (sogenannte *docstrings*) ausgegeben. Die Hilfe lässt sich mit *!help* aufrufen. Zur Erfüllung dieser Anforderung müssen also die Funktionen entsprechend dokumentiert werden.<sup>182</sup>

#### 4.1.14 Vorhersage

Da es in der CMDB keine entsprechende Funktion zur Vorhersage der Ressourcenauslastung gibt, kann auch kein ChatOps Interface dazu geschaffen werden.

#### 4.1.15 Standardanfragen

Da im Rahmen der Experteninterviews Administratoren der CMDB befragt wurden, welche Aufgaben bzw. funktionale Anforderungen ein Chatbot erledigen können soll, kann diese Anforderung als erfüllt betrachtet werden.

### 4.2 Funktionale Anforderungen

Die Umsetzung oder geplante Umsetzung der funktionalen Anforderungen wird im Folgenden einzeln beschrieben. Wenn mehrere Anforderungen durch eine Funktion abgedeckt werden können, sind diese in einem Punkt zusammengefasst. Dabei wird darauf eingegangen, aus welchem Tool oder welcher Quelle die Informationen gewonnen werden, wie die Funktionen im Bot Plugin umgesetzt sind und wie die Ausgabe erfolgt. Befehle sind dabei kursiv dargestellt und Parameter in spitzen Klammern. In den Code Listings wird das Beispielobjekt *mylittleserver* verwendet.

#### 4.2.1 Suchfunktion

Die Suche nach Objekten wird über die *idoitcli search <Suchbegriff>* Funktion realisiert. Analog dazu wurde das Bot Kommando *!cmdb search <Suchbegriff>* implementiert. Die Ausgabe wird dabei zur besseren

<sup>182</sup>Vgl. errbotio (2018b), o. S.; Vgl. Goodger (2001), o. S.

Lesbarkeit formatiert (Listing 4.7). Bei leerem Suchstring erfolgt *None* als Ausgabe. Die Funktion *clirun* ist der Wrapper für das *idoitcli* binary.

Listing 4.7: Suchfunktion

```
1 def search(term):
2     """ search for term """
3     if term == '':
4         return None
5     search_cmd = 'search_{}_{}'.format(term)
6     out = clirun(search_cmd)[1]
7     return out
```

#### 4.2.2 Nächste freie IP

Die Suche nach der nächsten freien IP wird mit der Funktion *idoitcli nextip <Netz>* realisiert und ist mit *!cldb nextip <Netz>* abrufbar. Wenn ähnlich wie beim IVZ die ersten Adressen eines Netzes für Infrastruktur reserviert sein sollen, muss darauf geachtet werden, dass diese auch in der CMDB entsprechend blockiert sind. Die *nextip* Funktion gibt die erste Adresse aus, der kein Objekt zugewiesen ist.

Listing 4.8: Nächste freie IP

```
1 @botcmd
2 def cldb_nextip(self, msg, args):
3     """ next free IP address """
4     return cldb.clirun('nextip', str(args))[1]
```

#### 4.2.3 Serverplanung

Um Ort, Name und IP eines neuen Servers zu speichern, kann dieser mit *i-doit-cli* initial angelegt und mit den entsprechenden Informationen versehen werden (Listing 4.9). Als Bot Kommando muss dafür entsprechend *!cldb server <Servername>, <Rack>, <IP>* verwendet werden. Nach der Erstellung folgt automatisch eine Suche nach dem neuen Objekt und Ausgabe auf dem Bildschirm, um den Erfolg überprüfen zu können.

Für virtuelle Maschinen wird entsprechend der Befehl *!cldb vm <Servername>, <IP>* verwendet, der keinen Standort Parameter benötigt und das Objekt in der Kategorie *Virtual Server*, statt *server* speichert.

Listing 4.9: Serverplanung mit i-doit-cli

```
1 idoitcli save server/mylittleserver
2 idoitcli save server/mylittleserver/location -a location="myrack"
3 idoitcli save "server/mylittleserver/Host_address" -a "IPv4_
  ↪ address="192.168.0.2"
```

#### 4.2.4 Standortabfrage, IP-Adresse, Statusabfrage, Softwarestand, Netzinformationen, Kontaktperson

Um Details zu einem Objekt anzuzeigen, können die jeweiligen Kategorien abgefragt werden. IP-Adresse und Netzinformationen aus *Host address*, Standort in *Location*, der Status in *General*, die Kontaktperson in *Contact assignment* und die Software in *Applications* (Listing 4.10). Die Ausgabe der Details zu einem Objekt erfolgt nach Eingabe von `!cldb detail <Servername>`. Die Informationen aus den Kategorien werden per `formatter` Funktion gefiltert und konsolidiert.

Listing 4.10: Detailansicht eines Objekts mit i-doit-cli

```
1 idoitcli read "mylittleserver/Host_address"
2 idoitcli read "mylittleserver/Location"
3 idoitcli read "mylittleserver/General"
4 idoitcli read "mylittleserver/Contact_assignment"
5 idoitcli read "mylittleserver/Applications"
```

#### 4.2.5 Bestandsprüfung, Standortreport, Ressourcenübersicht

Um den Bestand von Objekten im Lager oder an einem Standort prüfen zu können werden in der i-doit C MDB Reports verwendet, die vom Betreiber individuell gestaltet und jederzeit ausgeführt werden können. Die verwendeten Ressourcen eines Kunden oder einer Anwendung werden ebenfalls über Reports abgerufen. Im i-doit-cli gibt es derzeit keine Funktion, um Reports abzurufen. Um diese Features umzusetzen müsste entsprechend eine Erweiterung des i-doit-cli bei synetics beauftragt werden. Alternativ könnte auch betrachtet werden, ob es sich per individueller API Abfrage umsetzen lässt. Eine andere Möglichkeit ist der Einsatz der i-doit *Console*, um per Cronjob die Reports täglich zu generieren und per Bot unter einer vordefinierten URL bei Anfrage herunterzuladen und in den Chat zu Posten. Im Rahmen dieser Arbeit kann dieses Feature jedoch nicht umgesetzt werden.

Eine mögliche Syntax für diese Funktion könnte `!cldb report <Reportname>` sein.<sup>183</sup>

#### 4.2.6 Ressourcen

Um die Ressourcenauslastung eines Geräts zu ermitteln, können die Kategorien *Memory* und *CPU* eines Objekts per `idoitcli read <Servername>` abgefragt werden (Listing 4.11). Im Bot Plugin ist diese Funktion per `!cldb load <Servername>` abrufbar.

Listing 4.11: Ressourcenauslastung eines Geräts

```
1 idoitcli read mylittleserver/Memory
2 idoitcli read mylittleserver/CPU
```

<sup>183</sup>Vgl. synetics (2019d), o. S.; Vgl. synetics (2019a), o. S.

### 4.2.7 Statuswechsel

Der Status eines Objekts lässt sich ändern, indem das entsprechende Attribut in der Kategorie *General* mit der *idoitcli save* Funktion gesetzt wird (Listing 4.12). Das Bot Kommando dazu lautet *!cldb setstate <Objektname> <Status>*.

Listing 4.12: Statuswechsel eines Objekts

```
1 idoitcli save "server/mylittleserver/General" -a "CMDB_status"="
  ↳ scrapped"
```

### 4.2.8 Logbuch

Ein Logbucheintrag kann vorgenommen werden, indem die *log* Funktion des i-doit-cli unter Nennung des Objektnamens aufgerufen wird (Listing 4.13). Analog dazu kann der Eintrag mit *!cldb log <Objektname> <Eintrag>* per Bot Kommando vorgenommen werden.

Listing 4.13: Logbucheintrag

```
1 idoitcli log mylittleserver -m "Test"
```

## Kapitel 5

### Fazit

#### 5.1 Zusammenfassung und Schlussfolgerung

ChatOps zur Bedienung der CMDB hat sich als neuartiges Thema herausgestellt, das noch schwer greifbar ist, weswegen auch keine reine Literaturarbeit möglich war, sondern Experteninterviews durchgeführt wurden.

Die Experten haben allesamt großes Potential in ChatOps im Allgemeinen gesehen und konnten sich für den konkreten Anwendungsfall CMDB einige Funktionen bzw. Tätigkeiten vorstellen, die von einem Chatbot übernommen oder zumindest unterstützt werden können.

Auch die Rahmenbedingungen, die gesetzt sein müssen, damit ein CMDB Chatbot betrieben werden kann, konnten anhand der Experteninterviews in Form von Qualitätsanforderungen erarbeitet werden. Wichtig dabei war es vor Allem die spezielle Perspektive *Rechenzentrum* zu betrachten, da für Mitarbeiter, die in dieser Branche arbeiten Aspekte wie beispielsweise die Betriebssicherheit sehr wichtig sind.

Ob die Einführung eines CMDB Chatbots wirklich zu einer Arbeitserleichterung führt, kann sich nur in der Praxis zeigen, da es bisher noch keine dokumentierten Referenzvorhaben gibt, ChatOps im Rechenzentrumsumfeld und speziell zur Bedienung der CMDB einzusetzen. Mit der Befragung der Experten wurde jedoch schon ein wichtiger Schritt getan, um die Voraussetzungen dafür zu schaffen, dass das im Rahmen dieser Arbeit erstellte Chatbot Plugin für die CMDB bei der täglichen Arbeit im Rechenzentrum wirklich eine Arbeitserleichterung bringt.

Auch die Culture of Automation, Measurement and Sharing, die ChatOps mit sich bringt und welche eine hohe Transparenz bedeutet, wird sich in der Praxis behaupten müssen. Die befragten Experten sahen zwar kein Problem in der Einführung dieser Arbeitsweise, allerdings ist das noch kein Garant für das Gelingen der Umsetzung. Es wird sich zeigen müssen, wie gut diese von den Mitarbeitern angenommen wird. Für einige bedeutet ChatOps wahrscheinlich einen Bruch mit alten Gewohnheiten und eine Umstellung der eigenen Arbeitsweise.

Bei mangelnder Akzeptanz ließe sich auch die Transparenz vermeiden und Befehle nur per direktem Chat mit dem Bot absetzen. Technisch ist das durchaus möglich. Wie im Praxiskapitel beschrieben wurde, lässt sich die Konfiguration dahingehend anpassen. Allerdings bedeutet ein Verzicht auf Transparenz auch einen Verzicht auf die Mehrwerte von ChatOps.

## 5.2 Ausblick

Bereits in den Interviews hat sich gezeigt, dass sich die Experten ChatOps durchaus auch für andere Bereiche des Rechenzentrumsbetriebs vorstellen können, bzw. gar die Vision einer Verkettung verschiedener Chatbot Plugins haben, in der ein gesamter Prozess abgebildet wird. Konkret ist das z. B. das Anlegen eines neuen Servers. Dieser kann bereits jetzt per Chatbot in der CMDB geplant werden und es wäre denkbar auch die DNS Einträge so vorzunehmen und eine virtuelle Maschine aus einer Vorlage zu klonen.

All die beteiligten Tools dieses Prozesses verfügen über eine API. Wenn also das CMDB Chatbot Plugin die erwartete Arbeitserleichterung bringt und auf die nötige Akzeptanz stößt, wäre es durchaus denkbar in Zukunft auch Chatbot Plugins für diese Tools oder auch weitere, die im Rahmen dieser Arbeit noch nicht genannt wurden, zu implementieren.

## Kapitel 6

# Quellenverzeichnis

### Literatur

Betz, Charles (2016). *Managing Digital: Concepts and Practices*. 1. Aufl. Minneapolis: Digital Management Academy, Jan. 2016. ISBN: 978-0-9981346-0-4.

Bogner, Alexander, Beate Littig und Wolfgang Menz (2014). *Interviews mit Experten*. Wiesbaden: Springer Fachmedien, 2014. DOI: 10.1007/978-3-658-08349-6.

Buck, Konrad (2019). *Experteninterview Buck*. Köln, Düsseldorf, 2019.

Feindt, Jan (2006). „Gesprächskompetenz digitaler Agenten“. In: *Web-Kommunikation mit OpenSource : Chatbots, virtuelle Messen, Rich-Media-Content*. Berlin, Heidelberg: Springer Verlag, 2006, S. 79–85. DOI: 10.1007/3-540-29093-1\_8.

Feindt, Jan, Claudia Janßen und Stefan Sölbrandt (2006). „Gesprächskompetenz im I-can-EIB-System“. In: *Web-Kommunikation mit OpenSource : Chatbots, virtuelle Messen, Rich-Media-Content*. Berlin, Heidelberg: Springer Verlag, 2006, S. 147–212. DOI: 10.1007/3-540-29093-1\_13.

Gläser, Jochen und Grit Laudel (2010). *Experteninterviews und qualitative Inhaltsanalyse*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010. DOI: 10.1007/978-3-531-91538-8.

Gowrishankar, S. und A. Veena (2019). *Introduction to Python Programming*. CRC, 2019. ISBN: 9780815394372.

Grötz, Rudolf (2018). „Sprich mit mir! - Chatbots entwickeln“. In: *iX 6* (2018), S. 50–57.

Hand, Jason (2016). *ChatOps - Managing Operations from Group Chat*. Hrsg. von Brian Anderson und Virginia Wilson. 1. Aufl. Xpert.press. O'Reilly Media, Inc, 2016. ISBN: 978-1-491-96230-5.

Hemmerich, Alexander (2019). *Experteninterview Hemmerich*. Köln, 2019.

- Hitzler, Ronald (1994). „Wissen und Wesen des Experten: ein Annäherungsversuch - zur Einleitung“. In: *Expertenwissen: die institutionalisierte Kompetenz zur Konstruktion von Wirklichkeit*. Hrsg. von Ronald Hitzler, Anne Honer und Christoph Maeder. Opladen: Westdt. Verl., 1994, S. 13–30. ISBN: 3-531-12581-8.
- Kirsten, Daniel (2019). *Experteninterview Kirsten*. Köln, Düsseldorf, 2019.
- Littig, Beate (2008). „Interviews mit Eliten–Interviews mit ExpertInnen: Gibt es Unterschiede“. In: *Forum Qualitative Sozialforschung*. Bd. 9. 3. 2008, S. 37.
- Matthes-Nagel, Ulrike (1986). „Modelle und Methoden rekonstruktiver Theoriebildung“. In: *Subjektorientiertes Lernen und Arbeiten - Ausdeutung einer Gruppeninteraktion*. Hrsg. von Gerhard Ebert, Willi Hester und Klaus Richter. Bonn: Pädagogische Arbeitsstelle Deutscher Volkshochschul-Verband, 1986, S. 29–55.
- Mayring, Philipp (2015). *Qualitative Inhaltsanalyse – Grundlagen und Techniken*. 12. Aufl. Weinheim, Basel: Beltz Verlagsgruppe, 2015. ISBN: 978-3-407-25730-7.
- Meuser, Michael und Ulrike Nagel (1991). „ExpertInneninterviews – vielfach erprobt, wenig bedacht“. In: *Qualitativ-empirische Sozialforschung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 1991, S. 441–471. DOI: 10.1007/978-3-322-97024-4\_14.
- Meuser, Michael und Ulrike Nagel (1994). „Expertenwissen und Experteninterview“. In: *Expertenwissen*. Wiesbaden: Vieweg + Teubner Verlag, 1994, S. 180–192. DOI: 10.1007/978-3-322-90633-5\_12.
- Meuser, Michael und Ulrike Nagel (2010). „ExpertInneninterview“. In: *Handbuch Frauen- und Geschlechterforschung*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2010, S. 376–379. DOI: 10.1007/978-3-531-92041-2\_44.
- Middendorf, Jörg (2019). *Experteninterview Middendorf*. Köln, 2019.
- Olbrich, Alfred (2008). *ITIL kompakt und verständlich : effizientes IT Service Management ; den Standard für IT-Prozesse kennenlernen, verstehen und erfolgreich in der Praxis umsetzen*. 4. erw. und verb. Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2008. ISBN: 9783834804921.
- Pohl, Klaus und Chris Rupp (2015). *Basiswissen Requirements Engineering*. 4. Aufl. Heidelberg: dpunkt Verlag, 2015. ISBN: 9783864902833.
- Puscher, Frank (2018). „Gut zugehört - Wie Sprache das Dialog-Interface verändert“. In: *iX 6* (2018), S. 64–69.
- Rudkowski, Lena und Alexander Schreiber (2015). *Aufklärung von Compliance-Verstößen: Whistleblowing, Arbeitnehmerüberwachung, Auskunftspflichten*. Gabler Verlag, 2015. ISBN: 978-3-658-07044-1, 978-3-658-07045-8.



- Stary, Christian (1996). *Interaktive Systeme: Software-Entwicklung und Software-Ergonomie*. 2. Aufl. Wiesbaden: Vieweg+Teubner Verlag, 1996. ISBN: 978-3-322-83203-0. doi: 10 . 1007 / 978 - 3 - 322 - 83202 - 3.
- Sturm, Rick, Carol Pollard und Julie Craig (2017). „The Configuration Management Database and Configuration Management System“. In: *Application Performance Management (APM) in the Digital Enterprise*. Amsterdam: Elsevier Verlag, 2017, S. 253–266. doi: 10 . 1016 / b978 - 0 - 12 - 804018 - 8 . 15002 - 8.
- Vries, Holger de und Claus Möbus (2006). „Avatare im E-Learning und E-Business“. In: *Web-Kommunikation mit OpenSource : Chatbots, virtuelle Messen, Rich-Media-Content*. Berlin, Heidelberg: Springer Verlag, 2006, S. 69–77. doi: 10 . 1007 / 3 - 540 - 29093 - 1\_7.
- Weizenbaum, Joseph (1966). „ELIZA—a computer program for the study of natural language communication between man and machine“. In: *Communications of the ACM* 9.1 (Jan. 1966), S. 36–45. doi: 10 . 1145 / 365153 . 365168.

## Online

- ARD.de (2018). *Informations-Verarbeitungs-Zentrum (IVZ)*. 2018. URL: <http://www.ard.de/home/die-ard/fakten/abc-der-ard/555192/index.html> (besucht am 26.03.2019).
- Bendel, Oliver (2018). *Chatbot*. 19. Feb. 2018. URL: <https://wirtschaftslexikon.gabler.de/definition/chatbot-54248/version-277297> (besucht am 24.11.2018).
- Buck, Konrad (2018a). *Referenzen von Konrad Buck*. 2018. URL: <https://www.kbquadrat.com/referenzen-warum-sie-bei-konrad-buck-richtig-sind> (besucht am 01.06.2019).
- Buck, Konrad (2018b). *Über Konrad Buck*. 2018. URL: [https://www.kbquadrat.com/ueber\\_konrad\\_buck](https://www.kbquadrat.com/ueber_konrad_buck) (besucht am 01.06.2019).
- Donath, Andreas (2018). *Slack übernimmt Hipchat und schließt es im Februar 2019*. 27. Juli 2018. URL: <http://www.golem.de/news/chatdienste-slack-uebernimmt-hipchat-und-schliesst-es-im-februar-2019-1807-135715.html> (besucht am 05.05.2019).
- errbotio (2018a). *config-template*. 4. Apr. 2018. URL: <https://github.com/errbotio/errbot/blob/master/errbot/config-template.py> (besucht am 12.06.2019).
- errbotio (2018b). *Errbot*. 26. Feb. 2018. URL: <http://errbot.io/en/latest/> (besucht am 23.06.2019).
- errbotio (2018c). *Mentions*. 26. Feb. 2018. URL: [http://errbot.io/en/latest/user\\_guide/plugin\\_development/mentions.html](http://errbot.io/en/latest/user_guide/plugin_development/mentions.html) (besucht am 23.06.2019).
- errbotio (2018d). *Persistence*. 26. Feb. 2018. URL: [http://errbot.io/en/latest/user\\_guide/plugin\\_development/persistence.html](http://errbot.io/en/latest/user_guide/plugin_development/persistence.html) (besucht am 23.06.2019).

- errbotio (2018e). *Plugin development*. 26. Feb. 2018. URL: [http://errbot.io/en/latest/user\\_guide/plugin\\_development/basics.html](http://errbot.io/en/latest/user_guide/plugin_development/basics.html) (besucht am 23.06.2019).
- errbotio (2018f). *Setup*. 26. Feb. 2018. URL: [http://errbot.io/en/latest/user\\_guide/setup.html](http://errbot.io/en/latest/user_guide/setup.html) (besucht am 13.06.2019).
- Goodger, David (2001). *PEP 257 – Docstring Conventions*. 29. Mai 2001. URL: <https://www.python.org/dev/peps/pep-0257/> (besucht am 19.06.2019).
- Heisig, Benjamin (2019). *i-doit-cli*. 29. März 2019. URL: <https://github.com/bheisig/i-doit-cli> (besucht am 30.04.2019).
- Kirsten, Daniel (2017). *Daniel Kirsten*. 2017. URL: <https://www.computerwoche.de/ap/daniel-kirsten,2078> (besucht am 01.06.2019).
- Költzsch, Tobias (2019). *SLACK: Vom gescheiterten MMO zum 7,1-Milliarden-Dollar-Unternehmen*. 21. Jan. 2019. URL: <https://www.golem.de/news/slack-vom-gescheiterten-mmo-zum-7-1-milliarden-dollar-unternehmen-1901-138803.html> (besucht am 04.05.2019).
- Kuhn, Thomas (2015). *Sprachassistenten verändern unser Leben*. 28. Juli 2015. URL: <https://www.wiwo.de/technologie/digitale-welt/cortana-siri-google-now-sprachassistenten-veraendern-unser-leben/12093146.html> (besucht am 04.05.2019).
- LivePerson (2019). *Wie stehen sie generell der Nutzung von Chatbots zur Kommunikation mit Unternehmen gegenüber?* 2019. URL: <https://de.statista.com/statistik/daten/studie/797076/umfrage/wahrnehmung-des-chatbot-einsatzes-zur-kommunikation-mit-unternehmen-in-deutschland> (besucht am 15.04.2019).
- Luber, Stefan (2017). *Was ist eine API?* 8. März 2017. URL: <https://www.dev-insider.de/was-ist-eine-api-a-583923/> (besucht am 25.06.2019).
- Newland, Jesse (2013). *ChatOps at GitHub*. 7. Feb. 2013. URL: <https://speakerdeck.com/jnewland/chatops-at-github> (besucht am 04.05.2019).
- Python Software Foundation (2019). *Virtual Environments and Packages*. 20. Juni 2019. URL: <https://docs.python.org/3/tutorial/venv.html> (besucht am 20.06.2019).
- Sigler, Eric (2014). *What is Chatops*. 2. Dez. 2014. URL: <https://www.pagerduty.com/blog/what-is-chatops/> (besucht am 25.03.2019).
- Slack (2019). *Bots für deinen Workspace erstellen*. 2019. URL: <https://get.slack.help/hc/de/articles/115005265703-Bots-f%C3%BCr-deinen-Workspace-erstellen> (besucht am 17.06.2019).
- Sullivan, Mark (2018). *err-backend-cisco-webex-teams*. 26. Feb. 2018. URL: <https://github.com/marknull/err-backend-cisco-webex-teams> (besucht am 23.06.2019).

synetics (2019a). *Console*. 2019. URL: <https://kb.i-doit.com/display/de/Console> (besucht am 25.06.2019).

synetics (2019b). *i-doit-screenshot-ipam*. Jan. 2019. URL: <https://www.i-doit.com/wp-content/uploads/2019/01/i-doit-screenshot-ipam.png> (besucht am 30.04.2019).

synetics (2019c). *Logbuch*. 2019. URL: <https://kb.i-doit.com/display/de/Logbuch> (besucht am 10.06.2019).

synetics (2019d). *Report Manager*. 2019. URL: <https://kb.i-doit.com/display/de/Report+Manager> (besucht am 23.06.2019).

Weiss, Kevin (2019). *i-doit-bot*. 5. Juli 2019. URL: <https://github.com/kevin-weiss/i-doit-bot> (besucht am 05.07.2019).

Westdeutscher Rundfunk Köln (2019). *Sendetermine - Die Seite mit der Maus - WDR*. 2019. URL: <https://www.wdrmaus.de/sendetermine.php5> (besucht am 23.06.2019).

Zyane, Rania (2017). *How ChatOps Can Help You DevOps Better*. 18. Mai 2017. URL: <https://chatbotmagazine.com/how-chatops-can-help-you-devops-better-5-minutes-read-507438c156bf> (besucht am 01.05.2019).

## Anhang A

# Leitfaden für die Experteninterviews

### Einleitung ins Interview

- Bedeutung des Interviews erklären
- Vorgehen erklären (3 Themenkomplexe, Interviewer wird während des Gesprächs Notizen verfassen)
- Fragen seitens des zu Interviewenden?

### Themenkomplex 1: CMDB im Rechenzentrum

- Was verstehen Sie unter einer CMDB?
- Beschreiben Sie den Einsatzzweck einer CMDB aus Ihrer Sicht.
- Wozu nutzen Sie die CMDB bei der alltäglichen Arbeit?
- Welche Funktionen einer CMDB finden Sie besonders wichtig? (Zusammenarbeit und Schnittstellen)

### Themenkomplex 2: Chatbots allgemein

- Was verstehen Sie unter den Begriffen Chatbots und ChatOps?
- In welchem Zusammenhang hatten Sie schon mit Chatbots zu tun? Welche Erfahrungen haben Sie gemacht?
- Welchen Nutzen und welche Vorteile sehen Sie im Einsatz von Chatbots?
- Können Chatbots Ihrer Meinung nach auch sinnvoll unternehmensintern genutzt werden?

### Themenkomplex 3: Chatbots zur Bedienung der CMDB

- Welche Aufgaben kann ein CMDB Chatbot problemlos übernehmen, welche weniger gut?
- Welche Arbeitsschritte würde ein Chatbot vereinfachen?

- Welche Anforderungen muss ein Chatbot zur CMDB Interaktion erfüllen, in Bezug auf ...
  - Sicherheit? (Z. B. Berechtigung nur für bestimmte Gruppen)
  - Nachvollziehbarkeit? (Z. B. Änderungsnotizen in CMDB)
- Rechnen Sie mit Problemen beim Einsatz von Chatbots für die CMDB? Z. B. Akzeptanz der Mitarbeiter.
- Wie soll die interne Weiterentwicklung und Anpassung ablaufen? (Wahl des Bot Frameworks)
- Welche Persönlichkeit sollte ein CMDB Chatbot Ihrer Meinung nach haben? Welche Figur (z. B. aus Film und Fernsehen) assoziieren Sie damit?
  - Alter, Geschlecht, Auftreten
  - Erscheinung (Mensch, Tier, Maschine)
  - Geschichte und Herkunft
- Welche Merkmale sollten den CMDB Chatbot auszeichnen?
  - Humor, Offenheit, Redseligkeit
  - Auftreten (laut, leise, frech)
- Sollte ein CMDB Chatbot regelbasiert oder selbstlernend (KI-gestützt) sein? Gehen Sie dabei auf folgende Punkte ein:
  - Komplexität
  - Qualität
  - Betriebssicherheit
- Sehen Sie auch für andere Unternehmen Potential, einen CMDB Chatbot zu verwenden?
- Denken Sie, dass Chatbots auch für andere Rechenzentrumstätigkeiten sinnvoll sind?

## Anhang B

# Experteninterview Hemmerich

Kevin Weiss (**KW**) führte mit Alexander Hemmerich (**AH**) ein Experteninterview durch, um Anforderungen an einen CMDB Chatbot zu ermitteln. Herr Hemmerich ist Leiter des IVZ Rechenzentrums am Standort Köln.

### CMDB im Rechenzentrum

KW: Was verstehst du unter einer CMDB im Allgemeinen?

AH: Die CMDB, die wir im Einsatz haben nutzen wir für unsere Configuration Items im Rechenzentrum, vom Kabel über Server über Ausstattung innerhalb eines Rechnerraums. Das ist so eine große Anzahl von Komponenten: Lizenzen, Supportverträge. Die lässt sich gar nicht mehr anders handhaben und verwalten. Hinzu kommt, dass sie durch unsere ISO 27001 Zertifizierung sogar gefordert ist.

KW: Wie benutzt du denn die CMDB bei der täglichen Arbeit?

AH: Um Gegenstände, und Bestände abzufragen. Also: wie viele Kabel haben wir noch im Lager, wo ist der Server XY, ist der schon verschrottet, ist der eingelagert? Und auch für die Abrechnung mit unseren Kunden, die z. B. eine Anwendung, ein Service belegt, sei es Speicher, sei es CPU, um das zusammenzutragen und für die Abrechnung an unsere kaufmännische Leitung weiterzugeben.

KW: Beschreibe doch mal bitte einen konkreten Anwendungsfall, wie du eine CMDB nutzt, wie du einen Workflow durchläufst.

AH: Es kann vorkommen, dass ein Kunde sich zusätzliche virtuelle Maschinen wünscht. Wir prüfen dann über die CMDB, erstellen einen Report, schauen dann, welche Ressourcen sein bei uns bestellter Service zurzeit belegt, vergleichen das dann mit der Menge, die er bestellt hat und wenn da noch Luft ist, z. B. für CPU oder RAM Ressourcen kann man entsprechend der Anforderung eine Freigabe geben und die kann dann in die Umsetzung gebracht werden. Wenn die Ressourcen erschöpft sind, geht das ganze natürlich dann zu unserem Vertrieb und daraus entsteht in der Regel eine Preisinformation für den Kunden und er kann sich dann überlegen, ob er diese Ressourcen zusätzlich bestellt oder nicht.

KW: Ok und welche CMDB Funktionen findest du besonders wichtig? Also auch im Zusammenhang mit anderen Tools, die wir einsetzen?

AH: Wir haben uns dazu committet, dass die CMDB bei uns im Haus das führende System ist. Das heißt, dass alle anderen Systeme, sei es das Monitoring oder seien es irgendwelche Scanner, die automatisch nach Systemen scannen, dass dies zuliefernde Systeme sind und die CMDB eben als führendes System allein diese Daten aggregiert, um eine Gesamtsicht auf das Rechenzentrum, auf die Ausstattung und sogar auf die Standorte des Unternehmens zu bieten. Also da sind nicht nur Rechenzentrums Configuration Items drin, sondern auch die Gebäude, die Büros, die Mitarbeiter, die Telefone usw..

## Chatbots allgemein

KW: Was verstehst du unter den Begriffen Chatbots und ChatOps?

AH: Das sind zusammengesetzte Worstücke, also Bot und Chat. Ich nehme mal an, dass das Bots sind, mit denen man kommunizieren kann, die dann entweder Informationen liefern, die man angefragt hat oder vielleicht sogar eine Aktion ausführen, im definierten Umfang.

KW: Hattest du schon mit Chatbots zu tun und welche Erfahrungen hast du dabei gemacht?

AH: Im professionellen Umfeld hatte ich da noch keine Erfahrungen mit. Ich habe allerdings im Rahmen deiner Entwicklungen im Slack die Bots gesehen, die du getestet hast. Also Erfahrung wäre übertrieben. Ich finde die Idee gut und ich bin gespannt, was man damit noch alles machen kann.

KW: Welche Vorteile könnten wir aus ChatOps ziehen?

AH: Also ich denke, dass Chatbots nicht den Menschen ersetzen werden, sondern ihn unterstützen in der täglichen Arbeit. Ein Chatbot könnte einfache Tätigkeiten übernehmen und zur Automatisierung des täglichen Rechenzentrumsbetriebs beitragen. Also man könnte z. B. ,wenn man eine Information aus der CMDB braucht den Bot fragen, welche RAM-Ressourcen in unserem Rechenzentrum die deutsche Welle belegt. Er liefert als Suchmaschine halt eine Information innerhalb des Rechenzentrums zurück, er kann aus meiner Sicht aber auch als Operator einfache Tätigkeiten wie die Erweiterung eines Filesystems oder eine DNS-Eintragung anstelle eines Admins erledigen, um den Alltag und die Automatisierung im Rechenzentrum voranzutreiben.

## Chatbots zur Bedienung der CMDB

KW: Was denkst du, welche Aufgaben ein CMDB Chatbot gut übernehmen kann und welche weniger gut?

AH: Ich denke das hängt stark davon ab, welche Schnittstellen die eingesetzte CMDB bietet und welche Kommandos sie extern über diese Schnittstellen akzeptiert. Ich könnte mir gut vorstellen, dass mit der i-doit CMDB z. B. über einen Chatbot die nächste freie IP in einem Netz anzeigen lässt, dass man Objekte sucht, die es in der CMDB gibt, um eine Abfrage zu machen: *Ist dieses Gerät in der CMDB eingetragen?*. Das wären die ersten Schritte, die ich mir vortellen könnte.

KW: Also welche Funktionen würde ein Chatbot vereinfachen? Ich denke an Aktionen wie das Anlegen einer VM und das Heraussuchen einer freien IP. Denkst du, da kann man viel automatisieren?

AH: Es wäre denkbar im Vorhinein einen Server in der CMDB über den Bot zu planen, also den Servernamen einzutragen, eine freie IP in einem entsprechenden Netz zu reservieren und dann als

Prozess direkt in Puppet die Konfiguration vorzunehmen, die VM auszurollen und die Netzkonfiguration herzustellen. Ich denke nur so ist zu verhindern, dass es entsprechende Inkonsistenzen gibt zwischen der produktiven Landschaft und den Einträgen in der CMDB.

KW: Welche Anforderungen hast du an einen CMDB Chatbot, wenn so etwas eingeführt wird?

AH: Es ist natürlich wichtig, dass zu jeder Zeit die Integrität der CMDB besteht und geschützt wird. Es darf nicht passieren, dass durch die Fehlfunktion oder Fehlbedienung eines Bots Daten in der CMDB entfernt oder verfälscht werden. Ich denke da muss man um die Betriebssicherheit im Rechenzentrum zu gewährleisten, bei der Entwicklung einiges bedacht werden, dass nicht jemand Fremdes unsere CMDB Integrität beeinträchtigen kann.

KW: Also sollte es nachvollziehbar sein, wer Änderungen gemacht hat und die Berechtigungen sollten nur für einen gewissen Personenkreis gelten?

AH: Das auf jeden Fall.

KW: Denkst du es könnte beim Einsatz von Chatbots Probleme geben?

AH: Ja, kann ich mir vorstellen. Allerdings besteht der größte Teil unserer Mannschaft aus jungen, motivierten Kollegen, die auch in der Freizeit viel mit neuen Technologien arbeiten. Ich könnte mir vorstellen, dass das in dem Bereich sehr großen Anklang finden wird. Andere, die schon ein paar Jahrzehnte in der IT sind müssen sich da vielleicht noch dran gewöhnen und nutzen dann noch die manuellen Eingaben in die CMDB. Aber ich glaube schon, dass die Bots, je nach dem wie die sie gestaltet sind, angenommen werden.

KW: Ein Grundsatz von ChatOps ist die Transparenz, also dass jeder sieht, was die anderen im Chatraum machen. Denkst du das wird im Team gut angenommen?

AH: Ja, ich glaube nicht, dass es unter den Kollegen zu Problemen kommen wird, ich glaube eher, dass man das über Betriebsräte usw. vorher abklären muss. Da gibt es eventuell eine Mitbestimmungspflicht, weil es ist ja eine Art Arbeitsüberwachung. Also der eine Kollege könnte sehen, was der andere macht, bzw. wie intensiv er an einem Tag gearbeitet hat, aber ich glaube nicht, dass es aus dem Team her zu Problemen führt.

KW: Ein Bot kann je nach Bedarf weiterentwickelt werden. Es gibt verschiedene Frameworks mit den Sprachen Ruby, Python oder Coffeescript, was denkst du wäre die sinnvollste Variante zur internen Weiterentwicklung?

AH: Wir haben aktuell einige Kollegen, die im Python Umfeld großes Know-How haben, ich empfehle immer Entwicklungen in den Sprachen zu betreiben, wo das Know-How da ist und auch in Zukunft da sein wird. Denn es gibt nichts Schlimmeres als eine Anwendung, die ein Praktikant oder Azubi entwickelt hat in einer Sprache, mit der sich hier keiner auskennt und die nach seinem Weggang eben nicht mehr wartbar ist. In dem Fall würde sich jetzt Python prima anbieten.

KW: Ein Chatbot soll auch immer eine gewisse Persönlichkeit haben. Was kannst du dir da vorstellen, also z. B. eine Figur aus Film und Fernsehen?

AH: Es gibt viele Möglichkeiten, aber ich denke da wir in einem professionellen Rechenzentrumsbetrieb sind, muss es schon etwas Ernsthaftes sein. Da wir für die Medien arbeiten, könnte es vielleicht sogar die Maus oder der Elefant sein von der Sendung mit der Maus. Natürlich sorgen wir immer dafür, dass eine gewisse Humor-Stimmung hier im Alltag herrscht, ich könnte mir auch vorstellen, dass man da ein Bisschen was Humorvolles einbaut, wenn die Anfrage etwas Humor beinhaltet,



dass der Bot dann auch humorvoll antwortet. Aber natürlich dürfen wir nicht vergessen, dass es ein Werkzeug für den professionellen Einsatz sein soll und es auch äußerlich so aussehen sollte.

KW: Es wäre also schon denkbar, dass die Maus oder ein entsprechender Avatar freche Antworten gibt, wenn man die Antworten falsch formuliert oder Ähnliches?

AH: Ja, natürlich. So ein Charakterzug würde das Ganze ein Wenig aufpeppen.

KW: Es gibt verschiedene Ansätze für Chatbots, der eine ist regelbasiert, der andere selbstlernend, also KI-gestützt. Welche Variante ist für einen CMDB Chatbot zu bevorzugen?

AH: Ich denke im Hinblick auf die Betriebssicherheit, die wir herstellen bzw. einhalten sollten wir auf einen regelbasierten Bot setzen. Also wir entwickeln ganz klar das Regelwerk, wovon der Bot nicht abweichen darf. Sonst könnte ich mir vorstellen, dass es eine Art von Entwicklung gibt, die nicht in unserem Interesse ist. Da werden dann eventuell Aktionen durchgeführt, die nicht geplant waren, sondern die der Bot in dem Moment durch seine KI für richtig gehalten hat, was in dem Fall zu einer Katastrophe oder Störung führt.

KW: Und du denkst auch, dass die Mitarbeiter leicht die Sprache des Bots lernen?

AH: Ja, das glaube ich auf jeden Fall.

KW: Denkst du, dass Chatbots auch für andere Tätigkeiten im Rechenzentrum geeignet wären?

AH: Ja, das glaube ich. Die einfachen täglichen Routinetätigkeiten, Konfigurationstätigkeiten, wie die bereits erwähnten Filesystemerweiterungen und DNS-Einträge oder das Starten einer Wartung im Monitoring, vielleicht sogar das Aufsetzen eines neuen Server über einen Bot. Ich sehe da großes Potential. Man darf halt als Mensch, als Admin, nicht die Angst vor solcher Technologie entwickeln und fürchten, dass man ersetzt wird, sondern man muss es als sinnvolle Ergänzung der täglichen Arbeit sehen.

KW: Denkst du, dass andere Unternehmen, die schon eine CMDB einsetzen auch Verwendung für sowas haben könnten?

AH: Ich sehe da keine Beschränkung. Jeder der eine CMDB einsetzt, kann von so etwas nur profitieren.

## Anhang C

# Experteninterview Middendorf

Kevin Weiss (**KW**) führte mit Jörg Middendorf (**JM**) ein Experteninterview durch, um Anforderungen an einen CMDB Chatbot zu ermitteln. Herr Middendorf ist CMDB Verantwortlicher beim IVZ.

### CMDB im Rechenzentrum

KW: Was verstehst du unter einer CMDB?

JM: Eine CMDB ist eine Configuration Management Datenbank, in der alle Configuration Items, bzw. alle Gerätschaften eines Unternehmens dokumentiert sind und in Relation gesetzt werden, also wie sie zusammenhängen oder korrelieren.

KW: Und wofür wird die CMDB eingesetzt?

JM: Bei uns im Rechenzentrum wird die CMDB eingesetzt, um alles was wir betreiben, wie Server, Lizenzen und Aufstellungsorte zu dokumentieren und für Supportfälle, zur Dokumentation und für Planungstätigkeiten zu nutzen.

KW: Wozu nutzt du die CMDB bei der täglichen Arbeit?

JM: Zur Dokumentation neu erstellter Maschinen. Um nachzugucken, wo ein Server steht, z. B. wenn ein Supportfall da ist. Um Reports für Planungen zu erzeugen und zu erfassen, wie viele Server an einem Standort überhaupt sind.

Wir haben jetzt ein Projekt, bei dem Rechenzentren umgezogen werden, da müssen wir natürlich wissen, um wie viele Komponenten es sich handelt. Außerdem zur Dokumentation von IP-Netzen, wenn ich eine neue IP-Adresse vergeben oder gucken möchte, wofür sie genutzt wird.

KW: Und welche Funktionen findest du denn besonders wichtig, auch im Hinblick auf die Zusammenarbeit mit anderen Tools?

JM: Wir haben eine Schnittstelle zum Monitoring System, also man hat die Möglichkeit, Daten aus dem Monitoring abzufragen und auch aus der CMDB geplante Daten ins Monitoring zu transferieren, ohne sich im Monitoring anmelden zu müssen. Das ist eine gängiger Schnittstelle und wichtige Funktion. Ansonsten ist die Reporting-Funktion sehr wichtig im täglichen Doing.

## Chatbots allgemein

KW: Was verstehst du unter den Begriffen ChatOps und Chatbots?

JM: Unter ChatOps verstehe ich eine Art Toolchain, die Entwickler und Administratoren zusammenschaltet und Anforderungen definiert, die automatisch getestet werden und dann in der Produktion eingesetzt werden können.

KW: In welchem Zusammenhang hattest du schon mit Chatbots zu tun?

JM: Ein Chatbot ist ja eigentlich eine Software, die auf Eingaben, also Text oder Sprache, reagiert. Da hat man mit den gängigen Sprachassistenten, die heutzutage im Umlauf sind Kontakt, sei es Siri, Alexa oder was es da alles gibt. Im eigentlichen textbasierten Umfeld gibt es sowas, wenn man mit Supportanbietern kommuniziert. Da sind die dafür da, um erst mal das Anliegen aufzunehmen und zu klassifizieren. Da gibt es verschiedene Varianten, es gibt Chatbots, die reagieren auf Sätze, es gibt aber auch welche, die reagieren auf Befehle. Also mit solchen Dingen hatte ich schon Kontakt.

KW: Welche Vorteile siehst du allgemein im Einsatz von Chatbots?

JM: Im Rechenzentrumsbereich sehe ich viele Vorteile. Für Standardaufgaben müssen dann nicht mehr die Mitarbeiter herumlaufen und andere befragen, z. B. wo ein Server steht, wo ein Gerät sich befindet, welche Software darauf läuft und welche IP-Adresse es hat. Da hat man bisher andere gefragt oder Mails geschrieben und auf eine Antwort gewartet und mit einem Bot entfällt das. Die Leute im Rechenzentrum, die alle Experten sind, können sich dann um die eigentliche Aufgabe kümmern und müssen sich nicht mit diesen Standardaussagen auseinandersetzen.

## Chatbots zur Bedienung der CMDB

KW: Was denkst du, welche Aufgaben ein CMDB Chatbot gut übernehmen kann und welche weniger gut?

JM: Ein Chatbot, der mit der CMDB zusammenarbeitet bietet einige Vorteile. Ich kann in einem Chat-tool, das ich auf dem Handy oder Rechner habe abfragen, wo ein Server steht. Also angenommen ich gehe jetzt ins Rechenzentrum und möchte wissen, in welchem Rack der steht. Dann kann ich nachschauen und weiß schon auf dem Weg, wo ich gucken muss. Ich kann alles, was für diese Item konfiguriert ist abfragen, ich kann gucken welche Software darauf läuft und welche IP-Adresse das Gerät hat. Zu welchem Subnetz es gehört, welches VLAN konfiguriert ist, wie viel Festplattenplatz verfügbar ist und wie stark der RAM ausgelastet ist. Das kann ich alles in einem herkömmlichen Chat Tool abfragen, ohne die Software öffnen zu müssen, die dann vielleicht auf Mobilgeräten nicht richtig angezeigt wird. Ich kriege die ganzen Anforderungen dann in meinem Tool, das ich dabei habe.

KW: Ok, also welche Arbeitsschritte würde ein Chatbot vereinfachen?

JM: Z. B. wenn ich im Rechenzentrum einen Server einbaue, kann ich dem Chatbot sagen, dass der Server mit dem Namen *Sowieso* im Rack *XY* an Platz *123* steht und der dokumentiert das dann für mich in der CMDB oder legt mir den Server grundlegend an mit Name und IP Adresse, damit ich die grundlegenden Informationen vor Ort eintragen kann. Im Umkehrfall baue ich einen Server aus und setze ihn per Chatbot in der CMDB auf einen anderen Status, also *außer Betrieb* oder *verschrootet*. Das kann man direkt vor Ort vereinfachen.

- KW: Welche Anforderungen hast du an einen Chatbot der mit der CMDB interagiert?
- JM: Zum Einen sollte der Chatbot keine Anfragen annehmen von Personen, die in der CMDB keine Berechtigungen haben. Entweder sagt man dem Chatbot, dass er nur Anfragen von bestimmten Personen entgegennehmen darf oder, dass er nur in einer bestimmten Gruppe aktiv ist, in der nur berechtigte Personen eingeladen werden, die dann diese Tätigkeiten vornehmen können. Also zumindest für das Schreiben in die CMDB, für das Lesen kann man vielleicht eine andere Regelung finden, z. B. jeder der im Rechenzentrum ist darf Abfragen machen: *Wo ist der Server?*, *Wer ist dafür zuständig?*, usw. . Die Anforderung an den Bot wäre dann noch, dass er die gängigen Methoden der CMDB nutzt, also APIs oder die entsprechenden Command Line Tools, die der CMDB Hersteller anbietet und nicht direkt in die Datenbank schreibt.
- KW: Bei der Benutzung des Bots wird der Bot User als Ersteller vermerkt. Soll auch berücksichtigt werden, wer mit dem Bot interagiert hat?
- JM: Ja, in einem Log File oder im Chatverlauf sollte nachvollziehbar sein, wer welche Änderung vorgenommen hat und dies sollte auch dauerhaft abgelegt werden. Im Optimalfall würde der Chatbot bei einer Änderung direkt in der CMDB vermerken, von welchem Nutzer die Änderung kommt. Der Chatbot dient dann quasi nur als ausführende Kraft
- KW: Rechnet du mit Problemen beim Einsatz von Chatbots?
- JM: Eigentlich sehe ich da nur Vorteile. Wie ich bereits am Anfang erzählt habe, ist es aktuell so, dass die Mitarbeiter in die Büros gehen und fragen: *Was ist das für ein Server? Wo steht der?*. Sie stellen immer die gleichen Fragen und diese Fragen können sie dann über einen Chatbot selbst beantworten. Also eigentlich sehe ich da keine großen Probleme. Das Problem ist halt, wenn die Chat Software irgendwo außerhalb des Unternehmens steht und man unternehmenskritische Dinge darin postet. Da muss natürlich sichergestellt sein, dass die entsprechende Regularien haben, dass diese Daten nicht irgendwann im Netz auftauchen oder weiterverarbeitet werden.
- KW: Der ChatOps lebt von Transparenz, also es findet alles in einem Raum statt und man kann sehen, was die anderen machen. Denkst du es gibt bei den Mitarbeitern Probleme bezüglich der Akzeptanz?
- JM: Nein, eigentlich nicht. Das einzige was ich mir vorstellen könnte ist, dass es als Arbeitskontrolle wahrgenommen wird. Es gibt vielleicht einen Mitarbeiter, der am Tag 10 Änderungen macht und der andere nur eine. Da könnte man sagen, dass er zehn mal so viel arbeitet. Sogar darf man natürlich nicht auswerten. Aber ich sehe in der Transparenz auch viele Vorteile. Wenn jemand eine Änderung vornimmt können alle anderen sich das da abholen und sehen direkt, dass sich was in der Umgebung geändert hat.
- KW: Siehst du denn auch Vorteile darin dass andere dann vielleicht etwas lernen können, also wie man einen Arbeitsvorgang durchführt?
- JM: Ja, erstens das und sie können sich selbst über Sachen informieren, die passiert sind. Nicht wie früher, dass man eine Änderung vornimmt und dann eine E-Mail schreibt, die keiner liest. So kann man in den Chat Raum reingehen und z. B., wenn man aus dem Urlaub kommt nachschauen, was in den letzten beiden Wochen passiert ist.
- KW: Es gibt verschiedene Bot Frameworks, die in Ruby, Python oder Coffeescript geschrieben sind. Was denkst du wäre das Sinnvollste in Betrachtung der Weiterentwicklung?

- JM: Also ich würde Errbot präferieren, weil er in Python geschrieben ist und wir hier Python Know-How vor Ort haben und auch die meisten Leute im Rechenzentrum Python Code lesen können. Sofern der alle Anforderungen im Laufe der Zeit erfüllen kann, würde ich den bevorzugen.
- KW: Bei einem Chatbot ist es vorgesehen, dass er eine gewisse Persönlichkeit hat. Also z. B. eine Figur aus Film und Fernsehen. Welche Figur würdest du mit einem CMDB Chatbot verbinden?
- JM: Ich bin da relativ neutral eingestellt. Ich finde es ganz lustig, wenn die eher so einen Comedy Charakter haben. Also wenn ich die gleiche Abfrage zwei mal stelle, dass er dann irgendwie einen süffisanten Spruch von sich gibt. Das fände ich ganz lustig. Eine eigene Person würde mir da nicht einfallen.
- KW: Ok, also er sollte dann vielleicht eher einen frechen Charakter haben, bei Falscheingaben usw.?
- JM: Ja, da muss man gucken, dass das für alle Kulturen, die hier im Unternehmen vertreten sind auch lustig wirkt. Es kann ja schon mal sein, wenn man eine andere Kultur hat, dass die Sachen, die andere lustig finden dann verletzend finden. Da muss man natürlich gucken. Ich finde bei so einem Bot eher besser, dass er nüchtern reagiert, also wenn ich ihm einen Befehl gebe, soll er genau das tun, was ich sage und nichts interpretieren. Das fände ich wichtig. Wenn der Bot anfängt zu interpretieren, kann das Ergebnis ganz anders aussehen, als das, was man beabsichtigt hat. Man sollte natürlich drauf achten, dass er Bot Schritte automatisiert und verlässlich bearbeitet. Nicht dass er nachher irgendwelche Dienste im Rechenzentrum abschaltet oder herunterfährt, nur weil er es anders interpretiert hat, als der Mensch es meint.
- KW: Ok, also denkst du, man sollte bei kritischen Aktionen auch nochmal eine Nachfrage machen?
- JM: Ja, das würde Sinn machen. Ich fände es auch gut, wenn der Bot bei einer Eingabe in die CMDB auch Auswahlmöglichkeiten zur Verfügung stellt, wenn man Sachen eingeben will vielleicht schon in einer Form oder Schreibweise vorhanden sind. Wir haben jetzt so ein Beispiel beim Hersteller. Da gibt es *Hewlett Packard*, *HP*, dann gibt es *HPE*, also ganz viele Schreibweisen für die gleiche Firma. Da sollte er dann die Übersetzung machen, wenn man sich verschrieben hat und den korrekten Namen vorschlagen. Das Ganze sollte dann interaktiv sein.
- KW: Es gibt verschiedene Arten von Chatbots. Regelbasiert oder KI-basiert, also selbstlernend. Du hast jetzt die Zuverlässigkeit, die der Bot erfüllen soll schon erwähnt. Denkst du denn auch, dass er eher regelbasiert sein sollte?
- JM: Also zumindest im Rechenzentrumseinsatz bin ich ein Fan von regelbasierten Sachen. Da will man einfach wenn man was dokumentieren oder abfragen möchte, dass er auch nur das macht und nicht noch andere Sachen, die er dazu gelernt hat. Es kann aber auch durchaus sein, dass eine künstliche Intelligenz für andere Anwendungsfälle Sinn machen würde. Also z. B., wenn ein neuer Mitarbeiter ins Unternehmen kommt und fragt, wo die Dokumente sind. Da könnte man dann was Anderes nehmen, aber hier für den Rechenzentrumseinsatz würde ich auf jeden Fall eine regelbasierte Lösung bevorzugen.
- KW: Denkst du, dass die Mitarbeiter einfach die Sprache des Bots lernen können?
- JM: Ich denke nicht, dass das ein Problem ist. Die meisten sind hier auf Konsolen unterwegs und sind da recht firm drin. Wenn der Bot in Betrieb ist kann man ja auch eine Einführungsveranstaltung machen, um zu zeigen, welche Sachen der kann. Oder man kann den Bot fragen: *Welche Befehle kennst du denn?*. Der sollte halt so eine Hilfe ausgeben oder ein paar Beispiele.

KW: Siehst du auch Potential für andere Rechenzentrumstätigkeiten?

JM: Ja, also alles, was automatisiert werden kann oder eine API hat, kann man damit wunderbar bearbeiten. Ich habe z. B. schon einmal einen Fall gesehen, bei dem virtuelle Maschinen provisioniert werden. Man sagt dem Bot, dass man eine Maschine braucht, die in gewisser Weise ausgeprägt ist und im Hintergrund laufen die entsprechenden Skripte und der Bot teilt einem dann mit, wann die Maschine fertig erstellt ist und wie die Daten sind, um darauf zu kommen. So was könnte ich mir sehr gut vorstellen. Es gibt auch andere Möglichkeiten, sich an Puppet z. B. anzudocken und da Konfigurationen zu erstellen und abzufragen, DNS Einträge zu machen, abzufragen, zu ändern. Da gibt es unzählige Möglichkeiten, die man benutzen kann. In der Vergangenheit war es halt schwierig, weil die meisten Softwarelösungen keine APIs hatten, da musste man in die Datenbanken schreiben und mittlerweile bietet fast jede Software auch APIs an und darüber kann man sehr schön mit einem Bot Aktionen auslösen.

KW: Siehst du auch für andere Unternehmen, die vielleicht schon eine CMDB einsetzen Möglichkeiten, ChatOps einzuführen?

JM: Ja, also alles, was automatisierbar ist, ist ja im Interesse von Rechenzentrumsbetreibern. Wir wollen nicht alle Aktionen zehn mal machen, es gibt ja auch nicht unbegrenzt Personal für solche Dinge. Alles was automatisierbar und machbar ist, ist auch für andere Unternehmen sehr sinnvoll und man nimmt mit so einem Chatbot natürlich die ganze Komplexität raus, ich sage dem, dass ich eine virtuelle Maschine brauche und das, was im Hintergrund passiert, kann ja sehr komplex sein und für den Anwender, der diese Anforderung erstellt, kann das trivial wirken. So einen komplexen Vorgang zu vereinfachen ist natürlich ein schöner Nebeneffekt.

## Anhang D

# Experteninterview Kirsten

Kevin Weiss (**KW**) führte mit Daniel Kirsten (**DK**) ein Experteninterview durch, um Anforderungen an einen CMDB Chatbot zu ermitteln. Herr Kirsten ist Produktmanager für die i-doit CMDB.

### CMDB im Rechenzentrum

KW: Was verstehst du unter einer CMDB?

DK: Eine CMDB, streng genommen nach ITIL steht für Configuration Management Database. Grundsätzlich muss man hier ein Bisschen differenzieren. Die Definition streng nach ITIL umfasst, dass die Elemente einer IT-Infrastruktur zentral erfasst werden und die Konfigurationen dieser Elemente und die Konfigurationen sind nicht 100-prozentig genau definiert, was da alles dazu gehört und es ist auch so, dass das Verständnis im Allgemeinen sehr stark sich unterscheidet, das können reine Asset Management Informationen sein, das können aber tatsächlich auch technische Konfigurationen sein. Nach ITIL spricht man heutzutage auch gar nicht mehr von einer reinen CMDB, sondern von einem sogenannten CMS, was wiederum ein Konstrukt ist, was auch nicht unbedingt eine zentrale Softwareplattform sein muss, sondern eine Vereinigung mehrerer CMDBen, die letztendlich unterschiedliche Datenquellen sein können. Was ich jetzt letztendlich unter einer CMDB verstehe, ist halt sehr unterschiedlich, weil ich in meiner Rolle als Produktmanager natürlich diese offizielle Definition kenne, die ITIL letztendlich vertritt oder herausgegeben hat. Grundsätzlich, das Gros der IT-Administratoren und Anwender gehen relativ verschwenderisch mit dem Begriff um und es wird halt in ganz vielen unterschiedlichen Ausprägungen genutzt. Es kann halt ein reines Asset Management sein, es kann halt auch eine sehr technische Datenbank sein. Wenn mich so jemand fragen würde, würde ich halt entsprechend sagen, es sollte idealerweise die Datenbank sein, die alle Komponenten einer IT-Infrastruktur vereint, dokumentiert und im Idealfall mindestens die technischen Grundkonfigurationen dieser Elemente aufnimmt.

KW: Wozu nutzt du die CMDB bei der täglichen Arbeit? Habt ihr so was auch selbst im Einsatz für die Komponenten?

DK: Wir selber nutzen natürlich unsere CMDB auch hier inhouse. Natürlich, um unsere inhouse-technischen Komponenten zu dokumentieren, dabei dient das bei uns hier intern tatsächlich als Nachschlagewerk für die tägliche, praktische Arbeit. Das heißt, wir haben hier keine automatisierten Prozesse oder Ähnliches, sondern wir machen eine manuelle Datenerfassung und erfassen hier die Dokumentation unserer internen und in der Cloud genutzten IT-Komponenten.

KW: Welche Funktionen findest du da besonders wichtig?

DK: Besonders wichtig ist für uns vor allen Dingen tatsächlich die Erfassung der IP-Adressen und auch Passwörter. Das ist so das, was hauptsächlich hier verwaltet wird. Wir haben eigentlich nahezu gar keine IT mehr inhouse, haben tatsächlich vor drei Wochen unsere letzten größeren physikalischen Server abgelöst. Das heißt wir dokumentieren sehr viel, welche Komponenten jetzt tatsächlich wo in der Cloud laufen und welche Services auch damit verknüpft sind.

## Chatbots allgemein

KW: Was verstehst du unter Chatbots und ChatOps?

DK: Also der Begriff ChatOps ist mir vorher noch nicht untergekommen. Ich kenne natürlich den Begriff DevOps, weiß auch grundsätzlich, was damit gemeint ist. Ein Chatbot als Solches ist mir sehr geläufig. Ich hab tatsächlich lange Zeit, fast zehn Jahre lang Eggdrop TCL Skripte geschrieben für Eggdrop Bots, hab also quasi das Thema Chatbots sehr ausführlich behandelt, hab da einiges mit gemacht. Am Ende des Tages ist ein Chatbot immer eine Instanz, die letztendlich User Input nimmt und andere Datenquellen anzapft und den Input so normalisiert und runterstrippt, dass er in einem menschenlesbaren Format am Ende ausgegeben wird. Das Thema ChatOps habe ich jetzt in dem Zusammenhang mal kurz gegooglet, vorab zu den Interviews und so weit ich das verstanden habe, soll es ja letztendlich eine durch einen Chatbot unterstützter Prozess für DevOps sein.

KW: Genau. Welchen Nutzen und welche Vorteile siehst du in Chatbots?

DK: Also grundsätzlich, wie gesagt, habe ich sehr sehr viel Erfahrung mit diesen Chatbots, man muss immer ein bisschen vorsichtig sein, dass man nicht zu viele Informationen versucht in so einen Chatbot hereinzusetzen. Grundsätzlich hat es immer dann den Vorteil, wenn man schnell Informationen benötigt und in Anführungszeichen zu faul ist, die passende Applikation dazu aufzumachen. Also ich kann mir vorstellen, wenn ich jetzt auf eine CMDB zugreife und ich möchte einfach sehr sehr schnell Informationen haben, dann könnte ich mir vorstellen, dass ich die gängigsten Funktionen, die ich sonst in der CMDB nutze zur Verfügung stelle, um in der Lage zu sein, einfach sich diese 5 extra Klicks über einen Browser über eine Anmeldung etc. zu machen.

## Chatbots zur Bedienung der CMDB

KW: Was denkst du welche Aufgaben ein CMDB Chatbot problemlos übernehmen kann und welche eher weniger gut?

DK: Ich könnte mir sehr gut vorstellen, dass gewisse häufig genutzte Funktionen von so einem Chatbot übernommen werden können, also wichtig ist immer aus meiner Sicht, dass zu komplexe Prozesse nicht unbedingt von so einem Bot übernommen werden, weil einfach der Output limitiert ist. Ich kann keine Grafik darstellen, ich kann also nur Textinformationen darstellen. Ich kann mir zum Beispiel sehr gut vorstellen, dass ich IP-Adressen von Systemen abfrage, dass ich zum Beispiel Standortdaten von Systemen abfrage oder Kontaktpersonen. Das sind so Dinge, wo ich sage: Ich will jetzt schnell Informationen über ein System haben oder vielleicht, wenn ich den kompletten Namen oder irgendwas nicht weiß, so eine Minimalsuche habe, wo ich dann mit wildcards oder Ähnlichem versuche, aus der CMDB mir komplette Namen oder Ähnliches herauszusuchen. Was ich mir noch vorstellen könnte, wäre zum Beispiel, dass ich mir neue freie IP-Adressen reserviere,



dass ich vielleicht sage: Okay, ich triggere *getip*, gib den Netznamen an und bekomme die nächste freie IP-Adresse reserviert. Das sind Dinge, die ich mir vorstellen könnte. Was ich mir noch vorstellen könnte, wäre, dass ich als Administrator Änderungen am Logbuch sehr schnell erfasse. Dass ich zum Beispiel einen Patch eingespielt habe und ich habe jetzt auch wieder keine große Lust, auf das i-doit System zu connecten, sondern ich schreibe schnell *addnotes* "Gerade Patch eingespielt". Das sind Dinge, die ich mir gut vorstellen könnte, wobei hier, glaube ich, die Ausgabe von Informationen eher im Vordergrund steht, dass ich halt aktiv Dinge suche, bzw. über den Bot finde. Die Eingabe von Informationen ist wahrscheinlich eher im Hintergrund, da das von der Normalisierung her und von der Fehleranfälligkeit her nicht ganz so viel Sinn macht.

KW: Welche Anforderungen muss denn ein CMDB Chatbot erfüllen?

DK: Ja, das ist jetzt sehr allgemein gefragt.

KW: Z.b. in Richtung Sicherheit oder Nachvollziehbarkeit.

DK: Ein sehr schwieriges Thema bei so etwas ist natürlich immer das Rechtesystem. Ich muss natürlich gucken, dass der Chatbot nur die Informationen tatsächlich darstellt, die den Usern, denen der Chatbot zur Verfügung steht auch zugänglich sind. Wenn ich also mit einem sehr restrizierten Rechtesystem arbeite, wird es mit dem Einsatzszenario für den Chatbot eher schwierig, weil Leute auf die Idee kommen könnten, das Ding zu missbrauchen, um Informationen abzurufen, auf die sie keinen Zugriff haben sollten. Natürlich muss ich halt gucken, dass ich so einen Chatbot nur im internen Netz betreibe, um halt nicht meine CMDB ins Internet zu exposen. Da müsste man halt entsprechend vorsichtig sein. Ich sage mal, wenn ich jetzt so einen Chatbot bauen würde technologisch, würde ich halt sagen, Slack ist der absolute Standard heutzutage im Unternehmen. Das wäre halt das erste, worauf ich das Ding konzentrieren würde. Grundsätzlich würde ich aber gucken und da spricht der Produktmanager in mir, dass ich so einen Chatbot plattform-agnostisch schreibe, d.h. ich würde ihn so schreiben, dass er ins Backend auf die CMDB zugreifen kann und ich nachher in der Lage bin, verschiedene Chatsysteme daran anzubinden, sei es jetzt Slack oder andere.

KW: Rechnest du denn mit Problemen beim Einsatz von Chatbots, auch in Bezug auf die Akzeptanz der Mitarbeiter?

DK: Akzeptanz der Mitarbeiter in sofern nicht, weil mit solchen Dingen ist es immer so: Es muss halt einer nutzen und meistens werden diese Bots halt in öffentlichen Channels benutzt und das ist dann eher so was Virales. Einer fängt halt an, das zu nutzen, der Andere findet das cool, sieht das in nem Channel und dann verselbständigt sich das. Allerdings bekommt man auch null Feedback, ob da was fehlt oder was nicht passt. Deswegen sehe ich jetzt keine Gefahr oder Ähnliches in dem Sinne, weil wenn es nicht gut funktioniert, wird es einfach nicht genutzt. Dann ist auf gut deutsch gesagt für die Katz, so ne Entwicklung. Ich glaube aber, dass diese kleinen nützlichen Funktionen schon sehr gut akzeptiert werden.

KW: Es gibt ja verschiedene Bot Frameworks, die in Python, Ruby oder Coffeescript geschrieben sind. Was denkst du ist für die Weiterentwicklung da am Besten?

DK: Ich kenne mich da jetzt nicht so wahnsinnig aus in den aktuellen Technologien, also wenn ich jetzt Python höre, das ist natürlich eine sehr mächtige Scriptsprache, die nach wie vor sehr weit verbreitet ist. Da hätte ich auf jeden Fall wenig Bauchschmerzen, dass das kompatibel ist. Coffeescript sagt mir jetzt gar nicht so viel. Also neuere Technologien, die noch nicht ganz gesettled sind, da wäre ich vorsichtig an der Stelle.

- KW: Ein Chatbot sollte immer eine gewisse Persönlichkeit haben. Was kannst du dir da für einen CMDB Chatbot vorstellen?
- DK: Ich finde, das Ding zu personifizieren an der Stelle würde ich als nicht richtig erachtet, weil die Leute, die damit arbeiten sind keine Endkunden, sondern sind Techniker, die schnörkellos Informationen haben wollen oder eingeben wollen und nicht unbedingt mit Hallo und Guten Tag begrüßt werden wollen. Also ich würde, wenn überhaupt diesem Ding eine Roboter-Persönlichkeit geben. Dass das Ding, wenn überhaupt sich das personifiziert darstellt, sich auch als Roboter darstellt.
- KW: Es gibt ja verschiedene Ansätze für Chatbots, also regelbasiert oder selbstlernend, bzw. KI-gestützt. Was denkst du ist da für einen CMDB Chatbot die bessere Wahl?
- DK: Also ich denke primär würde ich das Ganze regelbasiert aufbauen. KI-gesteuert ist bestimmt interessant, nur glaube ich ist es schon sehr schwierig, den Bot darauf zu trainieren, weil der Anwender ja auch Feedback geben muss, ob das was er präsentiert bekommt überhaupt das ist, was er gesucht hat. Ich weiß nicht, ob diese Feedback Schleifen den Anwender nicht zu sehr frustrieren würden, denn er wird halt vortrainiert aber wenn ich jetzt einem Endanwender so etwas präsentiere und er muss quasi 3 mal erst mal sagen, das ist nicht das, was ich gesucht habe, oder zu viel oder zu wenig, dann ist da aus Anwendersicht schon irgendwie eine gewisse, also fehlt eine gewisse Ernsthaftigkeit und führt halt auch letztendlich zu Frustrationen auf Anwenderseite. Ich finde das interessant, technologisch. Das kann auch zu sehr interessanten Ergebnissen führen, ich würde mich aber primär erst einmal auf kleine, regelbasierte Aufgaben konzentrieren.
- KW: Siehst du auch Potential für andere Unternehmen, so einen CMDB Chatbot zu verwenden?
- DK: Ohne Frage. Ich glaube, wenn so etwas existiert, dann hat das ganz großes Potential, dass das häufig eingesetzt wird. Man sieht anhand von Slack, dass die Unternehmenskommunikation immer stärker in Richtung Instant Messenger geht, das hat viel damit zu tun, dass die Leute privat auch mit der WhatsApp-Nutzung auch sehr vertraut sind und mit dem Medium mittlerweile professionell umgehen können und das ist ein absolut zukunftssträchtiges Medium und da Schnittstelle zu haben ist mehr als sinnvoll.
- KW: Ok, dann wären wir auch schon durch.
- DK: Ich darf noch eine kurze Sache ergänzen: Ich wäre vorsichtig mit Funktionen, die automatisch Dinge darstellen. Es gibt vielfach Chatbots, die automatisch immer Neuigkeiten, Dinge, die passieren in irgendwelche Channel rein posten. Das wird ganz häufig genutzt von vielen Leuten und letztendlich guckt aber dann keiner hin, weil halt Informationen immer wieder verloren gehen. Ich würde es tatsächlich so designen, dass das auf Nachfrage reagiert und nicht proaktiv irgendwelche Informationen spammt.

## Anhang E

# Experteninterview Buck

Kevin Weiss (**KW**) führte mit Konrad Buck(**KB**) ein Experteninterview durch, um Anforderungen an einen CMDB Chatbot zu ermitteln. Herr Buck ist PR Manager für die i-doit CMDB.

### CMDB im Rechenzentrum

KW: Was verstehen Sie denn unter einer CMDB?

KB: Ja, also eine Configuration Management Database, die möglichst genial und dynamisch und clever alle mögliche Items aufnehmen und verwalten kann, um den IT-Betrieb zu gewährleisten, also einen sauberen, zeitsparenden, effektiven IT-Betrieb gewährleisten. Prozesse planen, Changes durchführen, später mal, aber zunächst einmal den IT-Betrieb organisiert sicherstellen.

KW: Ok, welche Funktionen finden Sie da besonders wichtig?

KB: Zunächst mal finde ich die UI total wichtig, also gibt es eine grafische Darstellung des Rechenzentrums oder des Netzwerks und kann man in die Räume herein zoomen. Also die UI halte ich für sehr wichtig, weil heute alles an Video und möglichst einfache bildliche Darstellung gewöhnt ist. Also diese ganze Listenanzeige etc. und diese ganzen Klickwüsten, die man häufig vorfindet, finde ich nicht so gut. Auch die Benutzerführung muss gut sein. Da finde ich es gut, wenn man ein System hat, das es einem erlaubt, möglichst viel zu machen, um sehr vieles von dem, was man selber in seinem Unternehmen machen möchte abzubilden und das dann zu einem Standard für seine CMDB Usage festzulegen. Also dass man das in einer entsprechenden Software tun kann, dass man also all diese möglichen Sachen aus dem Konfigurationsmanagement tun kann, also Wartungspläne, Notfallpläne, Arbeitsabläufe organisieren, eine Wissensdatenbank. Dann ganz wichtig, die Anbindung an weitere Systeme im IT-Servicemanagement, also Ticketing oder Inventarisierungssoftware. Schnittstellen muss sie bieten, damit man einen möglichst guten Überblick über das gesamte Geschehen hat.

### Chatbots allgemein

KW: Was verstehen Sie unter den Begriffen Chatbots und ChatOps?

- KB: Chatbots sind Programme, die es einem erlauben, z. B. das Thema Ticketing zu automatisieren innerhalb einer CMDB Nutzung. Dass es eben nicht mehr ein Admin machen muss oder ein User mit dem Herstellersupport sprechen muss oder mit einem Mitarbeiter im Servicemanagement des Unternehmens, sondern dass vorgefertigte Standard Antworten und Fragen eben vorgefasst sind und dann von einem Bot beantwortet werden. Was die KI noch in Zukunft bringt mit intelligenten Bots, die selber sich das System angucken und sehen, wenn etwas nicht richtig läuft und eben auch schon einem Anrufer sagen können: Moment einmal, ich sehe, dass etwas nicht richtig läuft, das könnte die Ursache sein, ich prüfe das mal nach. Also das wäre so von den ganz einfachen Chatbots, die eben auch nur in einer Messaging Form schriftlich stattfinden, bis hin zu natürlichsprachlichen Bots, die dann mit dir reden.
- Das ganze Thema ChatOps sagt mir noch nicht viel. Da muss ich gestehen, da habe ich mich noch nicht wirklich mit befasst.
- KW: Ja, das ist auch noch recht neu. In welchem Zusammenhang hatten Sie denn schon mit Chatbots zu tun?
- KB: Also ich persönlich live im Grunde gar nicht. Ich habe nur diese ganzen Anwendungen bei der Telekom oder irgendwelchen Providern, die einen durch so eine Anfangsschleife leiten, bis man dann an einen richtigen Mitarbeiter kommt mit dem entsprechenden Anliegen.

## Chatbots zur Bedienung der CMDB

- KW: Welche Aufgaben kann ein CMDB Chatbot problemlos übernehmen und welche weniger gut?
- KB: Er könnte problemlos alles das, was der Admin als Standard Anfragen, die immer wieder kommen und ihn immer wieder nerven übernehmen, indem man aus einer Wissensdatenbank die häufigsten Fragen zusammenfasst und intelligent in diese Chatbot Programme einbaut. Da könnte man dann eben diese Verbindung herstellen und wenn man das sauber macht, würde der User das auch nutzen. Der kommt sich ja selber wahrscheinlich manchmal blöde vor, wenn er schon wieder die selbe Frage stellt, weil er es sich einfach nicht merkt oder einfach kein Interesse daran hat, sich solche Banalitäten zu merken und dann muss der sich dann halt von einem Chatbot antworten lassen, also wiederkehrende Fragen.
- Und dann, das ist aber wahrscheinlich schon eine Forderung an die Zukunft, nächstes Thema, Fragen bezüglich auflaufender Probleme, also predictive Aussagen machen zu können, zu dem, wie es im Netz aussieht. Eine Frage an so einen Chatbot könnte sein: Kann ich morgen mein neues Windows aufspielen, sind überall alle Systemanforderungen gegeben? Und dann müsste der Chatbot die hintergelagerten Systeme durchchecken möglichst schnell und sagen: Nein, bei dem Server ist noch nicht genug RAM da. Das wäre so die nächste Stufe aus meiner Sicht. Das könnten die heute im Grunde auch schon machen, ich wüsste aber keine Software, die das jetzt schon tut, keine CMDB. Der nächste Schritt wäre eine voll automatisierte UI, wo eben kein Bildschirm mehr gebraucht wird, sondern, ich möchte jetzt keine Produktnamen nennen, wo dann irgendwelche Sprachboxen stehen und dann könnte man sich als Admin oder User noch die Oberfläche ziehen von der CMDB oder von der Management Konsole, aber man kann auch über den Lautsprecher fragen: Was ist denn hier los? Dann erkennt der Bot anhand der Emotionalität, die in der Stimme steckt, das kann die KI ja zum Teil schon, was er da wohl meint und anhand des Profils dessen, der dann fragt und dann entsprechende Antworten liefern. Und das wäre dann wahrscheinlich schon das Ergebnis von ChatOps.
- Nun gibt es bestimmt sehr viele Einzelaspekte innerhalb einer IT-Administration, die sehr tief in

einer CMDB Nutzung drin steckt, ganz tolle Anwendungen gestrickt hat. Aus meiner Zeit bei synthetics mit der CMDB i-doit kenne ich einige Anwender, die wirklich irre Sachen gemacht haben und die könnten vieles von dem, was sie da tun dann eben auch über Bots realisieren, vereinfachen auch. Da mit Sicherheit auch sowohl eine einfachere Handhabung machen für sich und ihre Kunden im Unternehmen und vielleicht auch eine gewisse Einsparung erzielen, wiederum an Zeit, um sich an anderen Dingen widmen zu können und auch das Thema Akzeptanz bei den Endanwendern finde ich ganz wichtig, also wenn der Chatbot clever genug ist, könnte das die Akzeptanz von CMDBs oder überhaupt ITSM sozusagen quasi verbessern.

KW: Was für Anforderungen haben Sie denn an einen CMDB Chatbot? Z. B. in Beziehung der Sicherheit oder Nachvollziehbarkeit.

KB: Also immer vorausgesetzt, dass ich kein dezidiert Anwender bin, sondern nur jemand, der über solche Software kommuniziert hat und daher die Dinge auch kennt. Also, ja ich meine man vertraut plötzlich einem Lautsprecher Informationen an als Unternehmen, genau so wie als Anwender. Der Anwender könnte jetzt zum Beispiel, wir waren ja eben bei diesem dümmsten anzunehmenden User, der zum 100. Mal fragt, wie etwas geht. Der könnte dann ja irgendwann einmal auch die Gefahr sehen, dass wenn er jetzt zum 101. Mal die gleiche Frage stellt, dass er dann wegen Blödsinn entlassen wird. Also sowas könnte ich mir vorstellen, dass Leute einem Menschen mehr vertrauen, als einem System, das eiskalt sagt: Das scheint mir nicht wirklich der richtige Mitarbeiter zu sein. Das bringt mich aber zu dem ganzen System Sicherheit, wobei das ja eigentlich egal ist. Die Sicherheit muss sowieso stimmen, ob da jetzt die Ausgabe ein Chatbot macht oder eine Textzeile, das ist glaube ich egal. Da verschiebt sich ja nur etwas in der Schnittstelle zwischen Mensch und Maschine. Insgesamt bietet dieser Bot dann auch noch Optimierungsmöglichkeiten in der Art und Weise in Anwendungsfeldern, die eine Tastaturkommunikation nicht hinkriegt. Da kann man Emotionalität nicht so gut erkennen. Das werden Chat Bots in Kürze können, aber die Sicherheit dahinter sollte immer gleich bleiben. Also ich finde z. B. auch das Thema interessant: Sicherheit nicht nur im Bezug auf Daten, sondern auch eben im Bezug auf Emotionalität des Users. Wenn wir das in Zukunft checken können mit KI-Chatbots, dann kommt da nämlich noch eine ganz andere Sicherheitsdimension dazu. Also heute kann keiner sehen, ob ich müde bin oder desinteressiert oder zerzaust bin, das wird KI in Zukunft sehr genau sehen können, wenn es das sehen darf. Das heißt, es kommt vor Allem durch KI eine weitere Dimension der Sicherheitsaspekte hinzu.

KW: Rechnen Sie denn mit Problemen bei der Akzeptanz der Mitarbeiter?

KB: Ich denke, dass Chatbots es erst mal sehr schwer haben werden, sich von dem, was man in dem Telefonie Bereich kennt zu distanzieren. Also wenn die in der CMDB ähnlich daher kommen, wie bei der Telekom oder bei einem Versicherer oder der Krankenkasse, dann wird das schwer mit der Akzeptanz. Die müssen da irgendwelche cleveren Angänge finden, um anders zu klingen, anders zu reagieren und so weiter.

KW: Ein Chatbot sollte immer eine gewisse Persönlichkeit haben, z. B. eine Figur aus Film und Fernsehen. Was würden Sie da für einen CMDB Chatbot sehen?

KB: Ich fände vielleicht Dschinni nicht schlecht, das ist aber eher eine spaßige Idee. Aber ich glaube bei Chatbots kommt es genau auf so etwas an, also wie schaffen wir es Vertrauen zu gewinnen und gleichzeitig auch die Professionalität auszustrahlen, die so ein Admin eben auch erwartet. Er will ja auf der einen Seite schon eben Bier trinken und Spaß haben, aber der will verdammt nochmal auch seinen Job machen. Das muss wirklich so eine Mischung aus Ernsthaftigkeit und Lockerheit sein.

- KW: Es gibt ja verschiedene Ansätze für einen Chatbot, einmal regelbasiert und einmal selbstlernend, also KI-gestützt. Was denken Sie wäre da für einen CMDB Chatbot besser geeignet?
- KB: Da ich mich selber mit KI sehr beschäftige, würde ich sagen, besser gleich mit KI-Unterstützung. Die KI gibt es, die speech recognition ist mittlerweile recht gut, die kann auch Emotionalität ganz gut hinkriegen. Also da würde ich mich gar nicht mehr mit regelbasierten Dingen befassen, allerdings gibt es viele kleinere Unternehmen, die so etwas erst einmal machen müssen, weil sie das Andere nicht bezahlen können.
- KW: Haben Sie keine Bedenken bezüglich der Betriebssicherheit? Falls da etwas falsch interpretiert wird z. B..
- KB: Da gehe ich von aus, dass die Leute die so etwas anbieten, bzw. für ein Unternehmen dann auch customizen, dass die das im Griff haben. Das würde ich voraussetzen, aber das ist vielleicht eine blauäugige Vorstellung von mir? Ich sehe da aber auch die Fallstricke. Aber die Möglichkeit, Fehlinterpretationen auszuschließen würde ich voraussetzen und ich glaube das tun die Anbieter von KI-unterstützten Systemen generell auch.
- KW: Sehen Sie denn Potential für andere Unternehmen als das IVZ, einen CMDB Chatbot zu verwenden?
- KB: Ja, ich meine, wenn Sie da ein cooles Ding aufsetzen und an der Stelle freue ich mich, dass Sie das tun. Das war nämlich im Vorfeld gar nicht so sichtbar oder klar. Von daher finde ich das sehr spannend und ich kann mir in der Tat vorstellen, dass das jeder haben will. Erstens, weil es neu ist, zweitens, wenn es KI-gestützt ist einfach sehr arbeitserleichternd sein wird und wir reden im Moment viel zu viel über KI, das ist ein Stück weit auch klar, weil das noch relativ neu ist und wir haben zu wenig konkrete Anwendungen und jede, die da gut funktioniert und endlich auf den Markt kommt ist gut. Da sehe ich große Chancen für. Ich glaube, wenn Sie das Thema Datensicherheit im Griff haben, dass da nichts falsch läuft, dann werden Sie da einen Blumentopf gewinnen und da fällt mir noch ein in dem Bereich: Es gibt so viele Möglichkeiten, sich zu vertippen und die kann eine Sprachsteuerung, wenn man so will, durch Plausibilitätsprüfung, die sie unbedingt machen muss, auch aus Sicherheitsgesichtspunkten, dann wunderbar abfangen. Also wenn Sie KI einsetzen, wird das System in diesem Bereich so wirkmächtig sein, dass es all diese Fehler dann auch mit abstellen kann und wird damit wahrscheinlich per typos zustandekommende Fehler oder auch andere Anwendungsfehler einfach vermeiden. Das sehe ich im übrigen auch noch als Entwicklungsprojekt bei vielen CMDBs, dass sie einfach diese eben angesprochene einfach vermeiden. Dass sie da so Plausibilitätsprüfungen oder andere Sachen einführen, wo dann gefragt wird: Willst du das jetzt wirklich durchführen?

# Eidesstattliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde/Prüfungsstelle vorgelegen hat. Ich erkläre mich damit nicht einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Köln , 4. Juli 2019

(Ort, Datum)

( Kevin Weiss)