

## Seminararbeit

---

# Entwicklung eines Chatbots im Rahmen des Produkts SYNCROTESS

---

im Studiengang **Angewandte Mathematik und Informatik**

von  
Lara Brunkow  
Matrikelnummer: 3516098

1. Prüfer: Prof. Dr. rer. nat. Alexander Voß
2. Prüfer: Fabian Lüttgens M. Sc. RWTH

7. Dezember 2023

---

**Fachhochschule Aachen, Campus Jülich**  
Fachbereich 9  
Medizintechnik und Technomathematik

**INFORM GmbH**  
Geschäftsbereich Industrielogistik  
und Healthcare Management

# Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Seminararbeit mit dem Thema

## **Entwicklung eines Chatbots im Rahmen des Produkts SYNCROTESS**

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war. Ich verpflichte mich, ein Exemplar der Seminararbeit fünf Jahre aufzubewahren und auf Verlangen dem Prüfungsamt des Fachbereiches Medizintechnik und Technomathematik auszuhändigen.

Name: Lara Brunkow

Aachen, den 7. Dezember 2023

Unterschrift der Studentin

A handwritten signature in black ink, appearing to read 'Lara Brunkow', written over a horizontal line.

## Zusammenfassung

In der heutigen digitalen Geschäftswelt gewinnt die Integration von Chatbots in elektronische Dienstleistungen zunehmend an Bedeutung. Chatbots haben sich als wertvolle Unterstützung bei der Verbesserung der Servicequalität erwiesen und bieten Unternehmen einen vielversprechenden Weg zur Optimierung der Kundenzufriedenheit. In einer Zeit, in der Internetnutzer immer höhere Erwartungen an einen schnellen und effizienten Service haben, müssen Unternehmen diesen Bedürfnissen gerecht werden.

Das Transportleitsystem SYNCROTESS ist eine Softwarelösung der INFORM GmbH für eine optimierte innerbetriebliche Logistik. SYNCROTESS steuert Informations- und Arbeitsabläufe von der Zuordnung von Aufträgen zu Ressourcen bis hin zur Überwachung und Steuerung der Ausführung.

Das Modul Auftragsmanagement in SYNCROTESS Enterprise ermöglicht die Erstellung und Verwaltung von Transportaufträgen. Berechtigte Benutzer können Transportaufträge anlegen, einsehen, ändern, löschen und (ent)sperren.

Die Site Logistics App ist eine mobile Anwendung, die Mitarbeiter an einem Standort bei der Erfüllung ihrer operativen Aufgaben im Bereich Logistik unterstützt. Über die Site Logistics App können Aufträge gezielt an Mitarbeiter erteilt werden. Mitarbeiter können ihrerseits aktiv Aufträge über die App abrufen oder generieren.

Die Erstellung von täglichen und monatlichen Transportstatistiken schafft Transparenz für die Disponenten, unterstützt die Materialflussplanung und ermöglicht eine beleglose Disposition mit umfassender Dokumentation.

In dieser Seminararbeit wird zunächst ein umfassender Überblick über drei ausgewählte Frameworks und deren charakteristische Eigenschaften gegeben. Anschließend erfolgt eine detaillierte Bewertung dieser Frameworks anhand von vier definierten Kriterien: Benutzerfreundlichkeit, Integration mit anderen Systemen, Dokumentation und Aktualität. Basierend auf den Ergebnissen dieser Analyse wurde Power Virtual Agents von Microsoft ausgewählt, da es für die Anwendungsfälle von SYNCROTESS am besten geeignet ist und auch langfristig eine geeignete Plattform für die Weiterentwicklung und Anpassung bietet.

# Inhaltsverzeichnis

|  |           |
|--|-----------|
| <b>Abbildungsverzeichnis</b>                           | <b>1</b>  |
| <b>1 Einleitung</b>                                    | <b>1</b>  |
| 1.1 SYNCROTESS: Transportmanagement Software . . . . . | 1         |
| 1.2 Chatbots und ihre Vorteile . . . . .               | 1         |
| 1.3 ChatGPT . . . . .                                  | 3         |
| 1.4 Anwendungsfälle . . . . .                          | 3         |
| 1.4.1 Kriterien . . . . .                              | 4         |
| <b>2 Rasa Open Source</b>                              | <b>6</b>  |
| 2.1 Kriterien . . . . .                                | 6         |
| 2.1.1 Benutzerfreundlichkeit . . . . .                 | 6         |
| 2.1.2 Integration in andere Systeme . . . . .          | 9         |
| 2.1.3 Dokumentation . . . . .                          | 10        |
| 2.1.4 Aktualität . . . . .                             | 11        |
| 2.2 Zusammenfassung . . . . .                          | 12        |
| <b>3 Dialogflow Essentials</b>                         | <b>13</b> |
| 3.1 Kriterien . . . . .                                | 13        |
| 3.1.1 Benutzerfreundlichkeit . . . . .                 | 13        |
| 3.1.2 Integration in andere Systeme . . . . .          | 15        |
| 3.1.3 Dokumentation . . . . .                          | 16        |
| 3.1.4 Aktualität . . . . .                             | 16        |
| 3.2 Zusammenfassung . . . . .                          | 16        |
| 3.3 Kurzer Ausblick auf Manychat . . . . .             | 17        |
| <b>4 Microsoft Power Virtual Agents</b>                | <b>18</b> |
| 4.1 Kriterien . . . . .                                | 18        |
| 4.1.1 Benutzerfreundlichkeit . . . . .                 | 18        |
| 4.1.2 Integration in andere Systeme . . . . .          | 19        |
| 4.1.3 Dokumentation . . . . .                          | 20        |
| 4.1.4 Aktualität . . . . .                             | 20        |
| 4.2 Zusammenfassung . . . . .                          | 21        |
| <b>5 Fazit und Ausblick</b>                            | <b>22</b> |
| 5.1 Fazit . . . . .                                    | 22        |
| 5.2 Ausblick . . . . .                                 | 23        |
| <b>Literaturverzeichnis</b>                            | <b>I</b>  |

## Abbildungsverzeichnis

|    |  |    |
|----|--|----|
| 1  | Frage an ChatGPT zu HTTP-Anfragen . . . . .                      | 3  |
| 2  | Schnittstellen von SYNCROTESS . . . . .                          | 5  |
| 3  | Erster Kontakt mit dem Moodbot von Rasa . . . . .                | 6  |
| 4  | Antworten in Rasa . . . . .                                      | 7  |
| 5  | Intent, Entität und Rollen in Rasa . . . . .                     | 7  |
| 6  | Story in Rasa . . . . .  | 8  |
| 7  | Visualisierung eines Forms in Rasa[1] . . . . .                  | 9  |
| 8  | Rasas Architektur[2] . . . . .                                   | 10 |
| 9  | Erstellen eines Intents in Dialogflow ES . . . . .               | 13 |
| 10 | Lokales Testen des Chatbots in Dialogflow ES . . . . .           | 14 |
| 11 | Integration mit eigenem System in Dialogflow ES . . . . .        | 15 |
| 12 | Lokales Testen des Chatbots in Microsoft Bot Framework . . . . . | 19 |

## 1 Einleitung

### 1.1 SYNCROTESS: Transportmanagement Software

Das Transportleitsystem SYNCROTESS ist eine Softwarelösung der INFORM GmbH zur Optimierung der innerbetrieblichen Logistik. Es führt Informations- und Arbeitsprozesse von der Zuweisung von Aufgaben an Ressourcen bis zur Überwachung und Verwaltung der Ausführung effizient und kostengünstig durch.

Die Erstellung von täglichen und monatlichen Transportstatistiken schafft Transparenz für die DisponentInnen, unterstützt die Planung des Materialflusses und ermöglicht eine papierlose Disposition mit umfassender Dokumentation. Durch die objektive Erfassung von Daten wird eine fundierte Entscheidungsfindung erleichtert und eine effiziente Ressourcennutzung ermöglicht.

SYNCROTESS Enterprise ist mandantenfähig, d.h. unterschiedliche Firmen können verschiedene Stammdaten und Systemparameter verwenden, jedoch auf derselben Datenbankinstanz arbeiten. BenutzerInnen haben nur Rechte für ihre jeweiligen Mandanten und können zwischen diesen wechseln. Das Auftragsmanagement von SYNCROTESS Enterprise ermöglicht die Einrichtung und Verwaltung von Transportaufträgen. Berechtigte NutzerInnen können Transportaufträge erstellen, einsehen, bearbeiten, entfernen und sperren.

Die Site Logistics App ist eine mobile Anwendung, welche bei der Erfüllung von logistischen Aufgaben am Standort Unterstützung für MitarbeiterInnen bietet. Über die Site Logistics App können Aufträge gezielt an MitarbeiterInnen zugewiesen werden. MitarbeiterInnen haben die Möglichkeit, Aufträge über die App abzurufen oder zu erstellen.

### 1.2 Chatbots und ihre Vorteile

Ein Chatbot ist ein technisches Dialogsystem, das den Dialog zwischen Mensch und technischem System ermöglicht. Die Konversation erfolgt dabei via gesprochener Sprache oder Textsprache. Ein Chatbot bietet dabei immer eine Option der Spracheingabe und -ausgabe. [...] Die Definition eines Chatbots spiegelt sich ebenso im Begriff „Chatbot“ selbst wider. Dieser setzt sich aus dem Verb „to Chat“, zu deutsch „sich unterhalten“ und „Robot“, also Roboter zusammen.[3]

Damit ein Chatbot die Intention des Benutzers hinter der Spracheingabe verstehen kann, nutzen Frameworks oftmals Natural Language Understanding (NLU). Dabei wird der Roh-Text eingegeben und der Chatbot erhält maschinenlesbare Informationen zurück. NLU nutzt zwei Methoden zur Textverarbeitung: die regelbasierte und die neuronale Methode:

Beim regelbasierten Ansatz kommen beispielsweise reguläre Ausdrücke (Regex) zum Einsatz. Regex ermöglicht die Erkennung und Extraktion von Entitäten, die einem bestimmten Muster folgen.

In der Informatik stellt eine Entität ein „einzigartiges und identifizierbares Informationsobjekt“[4] dar und besitzt mehrere Attribute<sup>1</sup>, die es beschreiben. Eine mögliche Entität in dem Kontext der Anwendungsfälle wäre ein Transportauftrag, der über die Attribute Auftragsnummer, Start- und Endposition sowie Auftragsstatus verfügt.

---

<sup>1</sup>Eigenschaften

Regex kann beispielsweise verwendet werden, um E-Mail-Adressen anhand ihrer vorhersehbaren Struktur zu erkennen.

Diese Methode hat den Vorteil, dass sie wenig oder keine Daten benötigt und schnell ausgeführt werden kann. Sie kann jedoch keine Informationen verarbeiten, die sie nicht kennt.

Der neurale Ansatz nutzt ein Model auf der Basis von Transformern<sup>2</sup>, um Text in Absichten zu kategorisieren. Dieses Modell benötigt zum Beispiel die Eingabetexte eines Benutzers und kann auch Informationen verarbeiten, die ihm unbekannt sind, indem es fundierte Vermutungen anstellt.

Damit der Chatbot nach dem Verstehen der Informationen entscheiden kann, was zu tun ist, greifen Frameworks auf verschiedene Dialogverfahren zurück. Diese können ebenfalls regelbasiert oder neuronal sein.

Regelbasierte Ansätze verwenden Dialogbäume mit fest definierten Pfaden, während neuronale Ansätze auf Basis der bisherigen Interaktionen den besten Gesprächsverlauf auswählen. Dadurch entsteht für den Leser ein natürlicheres Gesprächserlebnis.

Um sicherzustellen, dass der Chatbot korrekt funktioniert, sollten beide Ansätze genutzt werden. Manuelle Tests und Anpassungen sind notwendig, um die Funktionalität des Chatbots zu überprüfen. Fehler des Chatbots werden korrigiert, in die Trainingsdaten integriert und der Chatbot anschließend erneut trainiert und getestet. Chatbotframeworks sind dynamisch und passen sich kontinuierlich an, um eine stetige Verbesserung zu gewährleisten.[6]

Die Integration von Chatbots in E-Services nimmt immer mehr an Bedeutung zu, da sie das Serviceniveau positiv beeinflussen und somit einen vielversprechenden Ansatz für die Steigerung der Kundenzufriedenheit darstellen.[7, 8]

Im Laufe der Internetnutzung haben sich die Erwartungen und Bedürfnisse der NutzerInnen an Dienstleistungen verändert.[9] Diese werden mit Schnelligkeit, Effizienz und einer breiten Angebotspalette assoziiert und sollten aus Sicht eines Unternehmens nicht vernachlässigt werden.[10] Um den neuen Anforderungen des Marktes gerecht zu werden, können Standarddienstleistungen um digitale Dienste erweitert werden. Dadurch wird ein breiteres, gezielteres Angebot für den/die KundIn gewährleistet.[7]

Chatbots agieren als Vertreter des Unternehmens und unterstützen die AnwenderInnen bei der Problemlösung sowie der Bereitstellung von Informationen.[8, 11] Ihre zuverlässige Persönlichkeit, ihr empathisches und offenes Wesen sowie ihr sozial orientierter Interaktionsstil tragen zur Verbesserung der Qualität des Austauschs bei und somit zur Serviceleistung des Bots.[12]

Zusammenfassend lässt sich sagen, dass das Ziel des Einsatzes von Chatbots darin besteht, „die Bedürfnisse der [KundInnen] bestmöglich zu erfüllen, denn wenn diese erfüllt sind, ist das Ergebnis wahrscheinlich eine positive Einstellung [...] und Loyalität, kurz: Kundenzufriedenheit[..].“[13]

---

<sup>2</sup>„Form von neuronalen Netzen, welche sich einen sogenannten Attention-Mechanismus (dt. Aufmerksamkeit) zunutze machen. Dabei wird ein Teil einer Eingangsgröße (beispielsweise ein Wort, eine Wortsilbe eines Satzes oder ein Pixel eines Bildes) mit den restlichen Teilen einer Eingangsgröße in Beziehung gestellt.“[5]

### 1.3 ChatGPT

OpenAI's ChatGPT ist ein bekannter Ansatz im Bereich der dialogbasierten Chatbots. Die Beta-Version wurde im November 2022 veröffentlicht und zeichnet sich durch die Fähigkeit aus, basierend auf umfangreichen Textdaten aus dem World Wide Web Antworten auf eine breite Palette von Themen zu generieren.[14]

Es ist allerdings zu beachten, dass ChatGPT derzeit keine direkten Text-Schnittstellen unterstützt. Dies bedeutet, dass eine reibungslose Integration in bestehende Systeme, wie es für die Anwendungsfälle in SYNCROTESS erforderlich ist, nicht umsetzbar ist. Die Stärken von ChatGPT liegen eher in der allgemeinen Konversation und der Bereitstellung von Informationen zu verschiedenen Themen. Dies mag für den spezifischen Einsatz in einem Unternehmenskontext nicht die optimalste Lösung darstellen.[14]

Aufgrund der derzeitigen Beschränkungen in Bezug auf Textschnittstellen und spezifische Anforderungen für definierte Anwendungsfälle bietet sich ChatGPT derzeit nicht als geeignete Option für die Implementierung im SYNCROTESS-System.

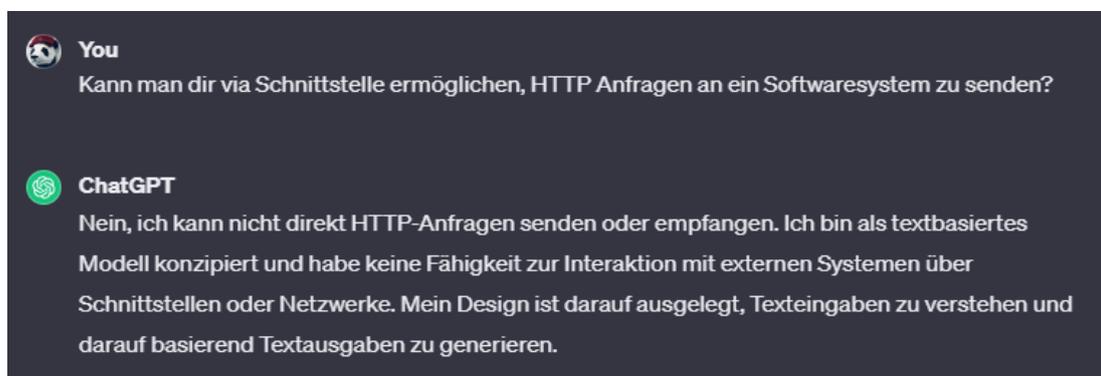


Abbildung 1: Frage an ChatGPT zu HTTP-Anfragen

### 1.4 Anwendungsfälle

In SYNCROTESS gibt es zwei Anwendungsfälle, bei denen ein Chatbot dem/der KundIn helfen würde. Der erste Fall beschreibt einen/eine DisponentIn, der/die außerhalb seines/ihres Büros ist und von einem/einer MitarbeiterIn angerufen wird, der/die wissen möchte, warum benötigtes Material nicht verfügbar ist. Der/Die DisponentIn hat keinen Zugang zum System, da er/sie normalerweise über einen Laptop arbeitet, und kann daher die Frage nicht beantworten. Diese Situation führt bei beiden MitarbeiterInnen zu Unzufriedenheit.

Durch die Unterstützung eines Chatbots kann der/die DisponentIn mithilfe der Materialnummer gezielt nach zugehörigen Aufträgen fragen. Der Chatbot gibt Auskunft darüber, ob das gesuchte Material aktuell einem Auftrag zugeordnet ist. Der/Die BenutzerIn kann diese Information nun an andere Beteiligte weitergeben und das Problem lösen. Alle involvierten Personen sind zufrieden und können ihre Arbeit ungestört fortsetzen.

Ein weiterer Anwendungsfall betrifft MitarbeiterInnen von Fremdfirmen, die sich zeitweise im Werk aufhalten. Wenn ein solche/r AuftragsmelderIn einen Auftrag erfassen möchte, ist er/sie darauf angewiesen, dass ihm das Stammpersonal bei

dieser Aufgabe behilflich ist, da ihm/ihr kein Endgerät mit installierter SiteApp zur Verfügung gestellt wird. Für die Nutzung dieser Anwendung muss ein Host angelegt werden, der das Endgerät und das System verbindet, sowie ein Benutzer mit den erforderlichen Rechten. Um Ressourcen zu sparen, sollte der Benutzer nach kurzer Zeit wieder aus der Datenbank gelöscht werden. Dies kann für eine/n MitarbeiterIn zu zeitaufwendig sein.

Der/Die Angestellte muss eine/n KollegIn kontaktieren - entweder per Telefon oder E-Mail - da er/sie keine Zugriffsmöglichkeit auf das System hat und demnach keinen Auftrag erstellen kann.

Verfügt der/die ZeitarbeiterIn über einen Chatbot, kann er seine Arbeit von jedem Endgerät aus erledigen. Daraus resultiert sowohl eine höhere Arbeitgeberzufriedenheit als auch ein optimierter Arbeitsablauf, da der/die ZeitarbeiterIn eigenständig arbeiten kann und nicht mehr auf die Unterstützung seiner/ihrer KollegInnen angewiesen ist.

#### **1.4.1 Kriterien**

Verschiedene Chatbot-Frameworks werden in dieser Arbeit vorgestellt und unter Verwendung objektiver Kriterien bewertet, um den am besten geeigneten Entwicklungsrahmen für die Umsetzung der Anwendungsfälle von SYNCROTESS zu ermitteln. Die Bewertung basiert auf den gesammelten Erfahrungen mit den jeweiligen Frameworks. Im Folgenden werden die Kriterien dargelegt:

**Benutzerfreundlichkeit:** Der Chatbot soll eine einfache und benutzerfreundliche Oberfläche bieten, um eine schnelle und intuitive Einarbeitung für EntwicklerInnen zu ermöglichen, die noch keine Erfahrung mit Chatbots und/oder dem Framework haben. Durch die Anwendung dieser Grundsätze wird sichergestellt, dass EntwicklerInnen das neue System gerne nutzen und zur Weiterentwicklung beitragen können.

**Integration in andere Systeme:** Der Chatbot soll nahtlos in andere Systeme integriert werden, um beispielsweise über eine Schnittstelle auf die Daten von SYNCROTESS zuzugreifen und eine von Benutzern angegebene Position zu validieren. Darüber hinaus sollte das Framework diverse Kommunikationskanäle bereitstellen, über welche NutzerInnen sich mit dem Chatbot austauschen können.

Einerseits ermöglicht dies einen Chatbot, der unabhängig von SYNCROTESS ist, und andererseits gewährleistet es, dass jeder/jede Mobiltelefon-BenutzerIn den Chatbot ohne die Installation von Apps oder die Erstellung eines neuen Accounts nutzen kann. Der Zugang zum Chatbot soll einfach und unkompliziert sein.

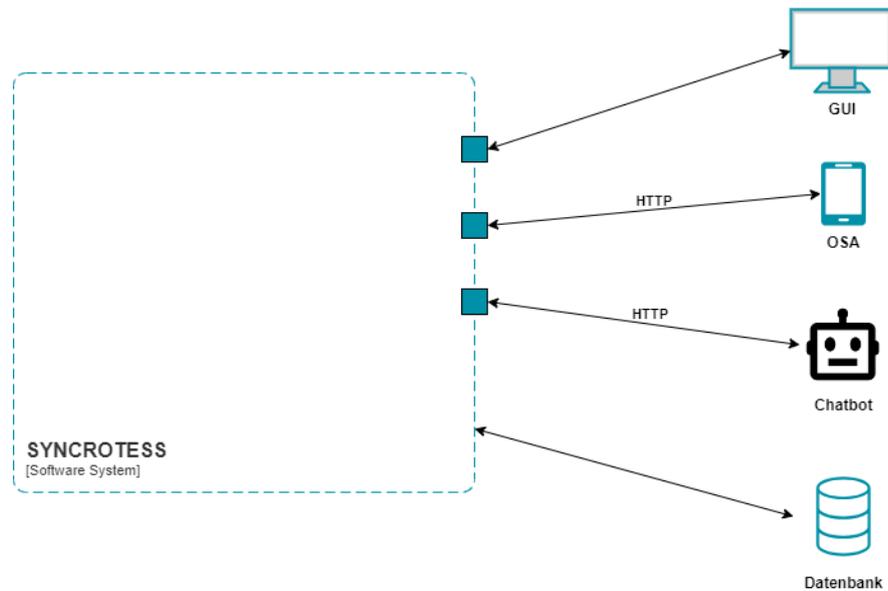


Abbildung 2: Schnittstellen von SYNCROTESS

**Dokumentation:** Eine ausführliche Dokumentation ermöglicht den Wissensaustausch innerhalb des Entwicklungsteams und zwischen unterschiedlichen Teams. Sie ermöglicht die Anwendung des produktspezifischen Chatbots auf mehrere Produkte und vereinfacht die Fehlerbehebung sowie Wartung.[15]

**Aktualität:** Eine aktuelle Software beinhaltet in der Regel die neuesten Patches, Updates und Sicherheitsfunktionen, um Sicherheitslücken zu vermeiden und die Effizienz des Chatbots zu gewährleisten. Durch Zugriff auf die aktuellsten Informationen kann sowohl Zeit eingespart werden als auch unnötige Verzögerungen vermieden werden. Ebenso profitieren Support-Teams bei der Beantwortung von Kundenanfragen.[16]

## 2 Rasa Open Source

Rasa Open Source ist ein aufgabenbasiertes Dialogsystem, das auf die spezifischen Bedürfnisse des/der Benutzers/Benutzerin eingeht, indem eine bidirektionale<sup>3</sup> Kommunikation eingesetzt wird. Das Framework basiert auf maschinellem Lernen<sup>4</sup>, Heuristiken<sup>5</sup> und Regeln. Der Fokus bei der Entwicklung eines Chatbots als Rasa-Assistenten liegt auf der Bereitstellung von Beispielanwendungen. Hier lernt das System, wie die Formulierungen des/der Benutzers/Benutzerin aussehen könnten und wie eine Konversation stattfinden sollte.

Rasa wurde entworfen, um hochgradig anpassungsfähig zu sein und sowohl Hobbyprojekte als auch komplexe Enterprise-Systeme<sup>6</sup> zu unterstützen.[6]

### 2.1 Kriterien

#### 2.1.1 Benutzerfreundlichkeit

Rasa wird ausschließlich mittels Programmiersprache bedient. Für die Erstellung eines Assistenten stellt Rasa ein Learning Center zur Verfügung, das schriftliche Anleitungen sowie Video-Tutorials beinhaltet.[6]

Zur Installation werden Visual Studio für C++ und Anaconda<sup>7</sup> benötigt. Nach der Einrichtung einer Anaconda-Umgebung mit Python 3.8<sup>8</sup> kann Rasa installiert werden und mithilfe des Befehls „rasa init“ der erste Kontakt mit einem Rasa-Assistenten hergestellt werden: der Moodbot.

```
2023-11-02 09:12:29 INFO      root - Rasa server is up and running.
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hi
Hey! How are you?
Your input -> I'm not feeling well
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input -> yes
Great, carry on!
Your input -> /stop
2023-11-02 09:13:37 INFO      root - Killing Sanic server now.
```

Abbildung 3: Erster Kontakt mit dem Moodbot von Rasa

Bevor auf die Elemente der Assistentenentwicklung eingegangen wird, wird die Struktur der Projektdatei erläutert. Zum Beispiel werden in der Datei „domain.yml“ die Antworten, Intents<sup>9</sup> und andere Elemente angegeben, die vom Assistenten verwendet werden können. In der „nlu.yml“ werden hingegen Intents und Entitäten definiert. Ein Assistent kann nur mittels vorgegebener Antworten mit dem/der BenutzerIn

<sup>3</sup>SenderIn und EmpfängerIn haben beide Rollen gleichzeitig inne oder können wechseln[17]

<sup>4</sup>„[...] Anwendung der künstlichen Intelligenz (KI). IT-Systeme lernen automatisch Muster und Zusammenhänge aus Daten und verbessern sich, ohne explizit programmiert zu sein“[18]

<sup>5</sup>Beurteilungskriterium für Theorien und Wissenschaftsprogramme auf Basis deren Informationsgehalt und das innewohnende Potential für die Weiterentwicklung des Erkenntnisstandes[19]

<sup>6</sup>skalierbare Softwarelösung, die in Firmen und anderen Organisationen verwendet wird[20]

<sup>7</sup>Distribution für die Programmiersprachen Python und R für die Datenwissenschaft, die die Paketverwaltung und -bereitstellung vereinfachen soll[21]

<sup>8</sup>Rasa 3.x unterstützt bisher nur bis Python 3.8

<sup>9</sup>Kategorisierung der Absicht eines Endnutzers für eine Unterhaltungsrunde[22]

kommunizieren. Hierbei können mehrere Texte angegeben werden, aus denen eine zufällig gewählt wird für ein natürlicheres Sprachempfinden. Es ist auch möglich, Variablen zu integrieren, um beispielsweise den/die BenutzerIn mit seinem Namen anzusprechen, wenn diese Information verfügbar ist. Rasa empfiehlt eine Antwortoption, die darauf hinweist, dass der Assistent ein Bot und kein Mensch ist. Antworten müssen einer bestimmten Namenskonvention folgen. Sie sollten als „utter\_<Anwendungsfall>“ benannt werden, z. B. utter\_greet für die Begrüßung oder utter\_default als Fallback<sup>10</sup>, falls der Bot die Benutzeranfrage nicht versteht.

```
responses:
  utter_greet:
    - text: "Hi! Wie geht's?"
    - text: "Hallo, {name}. Was gibt es?"

  utter_goodbye:
    - text: "Ciao!"
    - text: "Wiedersehen!"

  utter_default:
    - text: "Tut mir leid, das habe ich nicht verstanden, kannst du das anders formulieren?"

  utter_iamabot:
    - text: "Ich bin ein Bot, der via Rasa erstellt wurde."
```

Abbildung 4: Antworten in Rasa

Intents kategorisieren die Aussagen eines/einer Benutzers/Benutzerin und enthalten Beispielausdrücke. Der Assistent lernt, die Absichten des/der Benutzers/Benutzerin einordnen zu können, ohne auf den exakten Wortlaut der vorgegebenen Ausdrücke angewiesen zu sein.

Rasa verwendet Entitäten, um natürliche Sprache zu verstehen, Informationen zu sammeln und zu verarbeiten. Hiefür bietet Rasa vorgebaute Entitätsmodelle an: Duckling und SpaCy.

Duckling eignet sich gut zur Extraktion von Nummern, Daten, URLs und E-Mail-Adressen, während SpaCy maschinelles Lernen verwendet, um Namen, Orte und Unternehmen zu erkennen.

Durch die Verwendung von Synonymen, Rollen und Gruppen können Entitäten kategorisiert und präziser erkannt werden, beispielsweise Start- und Endpositionen.

```
nlu:
- intent: create_workorder
  examples: |
- Ich möchte einen Auftrag von [Lager A]{value: "Lager A", "entity": "position", "role": "origin"}
  zu [Lager B]{value: "Lager B", "entity": "position", "role": "destination"} erstellen
- Ich will einen Auftrag anlegen
- Auftrag erstellen
- Kannst du mir einen Auftrag anlegen
```

Abbildung 5: Intent, Entität und Rollen in Rasa

Für das Training des Assistenten sind Daten erforderlich, die sowohl benutzergenerierte Texte als auch Gesprächsmuster enthalten.

Rasa betrachtet Daten von BenutzerInnen, die mit den Assistenten interagieren, als

<sup>10</sup> „Rückfalllösung zur Verhinderung des Totalausfalls eines Systems“ [23]

wichtigste Trainingsdaten, da deren Analyse eine gezielte Verbesserung und Anpassung des Assistenten ermöglicht.

Zu den von dem/der EntwicklerIn bereitgestellten Daten gehören in Rasa Stories und Regeln.

Mittels Stories wird dem Assistenten beigebracht, was er als nächstes tun soll. Sie bestehen unter anderem aus Schritten (Steps), die Intents und Actions (z. B. eine Antwort) enthalten, um das Verhalten des Assistenten zu steuern. Mehrere Intents können mit derselben Action verknüpft werden. Folgender Austausch als Beispiel: Der/Die BenutzerIn bedankt sich bei dem Assistenten für die Hilfe. Der Assistent fragt, ob er dem/der BenutzerIn noch weiter behilflich sein kann. Egal ob der/die BenutzerIn die Frage verneint oder sich verabschiedet, der Assistent verabschiedet sich.

```
- story: ask for help
  steps:
  - intent: thanks
  - action: utter_ask_help
  - or:
    - intent: deny
    - intent: goodbye
  - action: utter_goodbye
```

Abbildung 6: Story in Rasa

Checkpoints ermöglichen den Übergang von einer Story zur nächsten.

Regeln beschreiben kurze Interaktionen, die jedes Mal gleich ablaufen sollen, wie zum Beispiel die Begrüßung. Während der Assistent aus einer Story lernt, ist eine Regel ein Muster, das strikt befolgt werden muss.

Rasas Slots-Funktion ermöglicht das Speichern von Informationen für den späteren Verlauf der Konversation. Mittels des Tags „influence\_conversation“ kann der Assistent Vermutungen für die nächste Action anstellen. Die Abhängigkeit des Ablaufs kann dabei vom Wert selbst oder dem Setzen des Werts abhängen. Falls der/die BenutzerIn beabsichtigt, einen Transportauftrag zu bearbeiten, wird der Assistent nach der zugehörigen Auftragsnummer fragen müssen. Sollte diese bereits bekannt sein, kann direkt nach der Art der Änderung gefragt werden.

Slot Mapping ermöglicht es, festzulegen, wann ein Slot mit Werten gefüllt werden soll. Dies kann verhindern, dass Informationen in einem Slot gespeichert werden, wenn die Entität vom Benutzer/von der Benutzerin außerhalb des Kontexts erwähnt wird. Zum Beispiel kann mit dem Tag „intent“ festgelegt werden, dass das Mapping nur angewendet wird, wenn der angegebene Intent vorhergesagt wurde. „not\_intent“ hingegen verhindert das Mapping für den angegebenen Intent. Außerdem können Rollen und Gruppen als Bedingungen berücksichtigt werden.

Wie ein Wert gespeichert werden soll, kann sowohl über vordefinierte Slot-Mappings als auch durch selbst programmierte Actions festgelegt werden.

In einigen Fällen müssen vom Benutzer/von der Benutzerin mehrere Informationen gesammelt werden, bevor der Assistent eine Action ausführen kann. Ein Beispiel hierfür wäre das Erstellen eines Transportauftrags, für den mindestens Angaben zur

Transportart<sup>11</sup> und Kapazität<sup>12</sup> notwendig sind.

Um diese Information einfach sammeln zu können, bietet Rasa das „Form“-Feature. Dabei beschreibt Rasa ein Formular als eine Schleife, die den/die BenutzerIn so lange nach Informationen fragt, bis genug Slot-Werte gefüllt wurden. Ziel ist es, dass der Assistent keine Fragen stellen muss, auf die bereits geantwortet wurde.

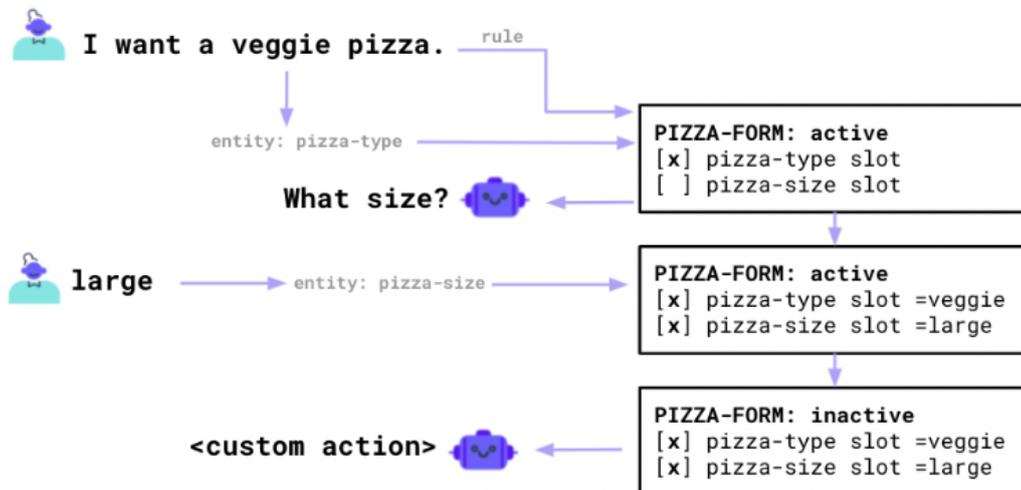


Abbildung 7: Visualisierung eines Forms in Rasa[1]

Um zu vermeiden, dass BenutzerInnen ungültige Daten eingeben, können benutzerdefinierte Aktionen (Custom Actions) genutzt werden. Der Assistent fragt dann erneut nach der fälschlicherweise angegebenen Information. Diese Actions zur Validierung entsprechen der strengen Namenskonvention „validate\_<form\_name>“, z. B. validate\_workorder\_create\_form.

Es ist ebenso möglich, die Bearbeitung mittels individuell gestalteter Formulare (Custom Forms) abzubrechen, wenn der/die BenutzerIn seine/ihre Meinung ändert oder Zwischenfragen zu beantworten sind, wie zum Beispiel „Bist du ein Bot?“, und anschließend mit dem Formular fortzufahren.

Um den Assistenten lokal testen zu können, muss er erst auf die aktuellen Trainingsdaten trainiert werden mittels „rasa train“. Wenn alle Dateien korrekt formatiert und angepasst wurden, kann per „rasa shell“ mit dem trainierten Assistenten kommuniziert werden.

### 2.1.2 Integration in andere Systeme

Um sicherzustellen, dass der Assistent über SYNCROTESS auch komplexe Aufgaben wie die Erstellung eines Auftrags oder die Validierung von Informationen durchführen kann, werden Custom Actions verwendet. Aufgaben wie das Erstellen eines Auftrags oder das Abrufen von Informationen aus der Datenbank erfordern geschriebenen Code und gehen über eine einfache Antwort hinaus.

Rasa trennt in diesem Zusammenhang die Verarbeitung von benutzerdefiniertem

<sup>11</sup>transportfähige, einzelidentifizierbare Entitäten bzw. Ladeeinheiten, z. B. Anhänger, Behältnisse (Container, Kisten), Schwerlast-Wagen bzw. elektrisch, auf einer Schiene verfahrbare Wagen

<sup>12</sup>Kapazitätsbeanspruchung der Transportart, numerisch. Dieser Wert gibt an, wie viel von der Kapazität einer entsprechenden Ressource von dieser Transportart in Anspruch genommen wird

Python-Code (Rasa SDK) von dem Dienst zur Ableitung der Intentionen des Benutzers und der nächsten Aktionen (Rasa Open Source). Das Framework bietet dafür vorgefertigte Codeausschnitte an, die zur Übernahme und Orientierung zur Verfügung stehen.

Zur Integration des Assistenten mit einer Website stehen bereits Tracker-Speicher (Tracker Store), Lock-Speicher (Lock Store) und eine Art Dateisystem zum Speichern von Trainingsdaten und der trainierten Modelle zur Verfügung.

Der Tracker Store speichert alle Unterhaltungen und ist insbesondere dann sinnvoll, wenn Tausende BenutzerInnen zu erwarten sind und eine Datenbank verwendet werden sollte.

Rasa verwendet einen Ticket-Lock-Mechanismus, um sicherzustellen, dass eingehende Nachrichten für eine bestimmte Konversations-ID in der korrekten Reihenfolge verarbeitet werden. Dies geschieht, indem die Konversation während der aktiven Verarbeitung gesperrt wird. Auf diese Weise können mehrere Rasa-Server<sup>13</sup> gleichzeitig als replizierte Dienste betrieben werden, und Clients<sup>14</sup> müssen keinen bestimmten Knoten ansprechen, um Nachrichten für eine bestimmte Konversations-ID zu senden. Das erwähnte Dateisystem kann sich entweder auf der Festplatte eines lokalen Computers oder auf dem Speicher einer Cloud-Ressource befinden.

Während der Entwicklungsphase wird der Assistent über eine Konsole<sup>15</sup> (z. B. Anaconda Powershell Prompt oder das Terminal von Visual Studio Code) mittels „rasa shell“ bedient. Hierfür wird ein kleiner Server lokal gestartet und die Kommunikation erfolgt mit Rasa über HTTP. Dies ist lediglich einer der unterstützten Kanäle von Rasa. In der „credentials.yml“ können weitere Ein-/Ausgangskanäle konfiguriert werden, wie WhatsApp, Messenger (Facebook), Slack oder andere von Rasa unterstützte Kanäle.

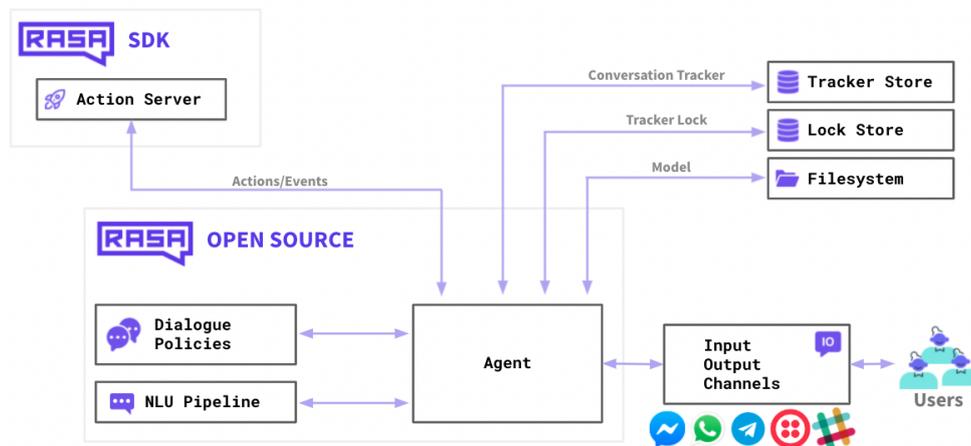


Abbildung 8: Rasas Architektur[2]

### 2.1.3 Dokumentation

Rasa bietet eine umfangreiche Dokumentation zu seinen Produkten an. Die Übersicht enthält Verweise auf die jeweilige Dokumentation beider Produkte und direkte Links

<sup>13</sup> „Ein softwarebasierter Server ist ein Programm, das einen speziellen Dienst anbietet, der von anderen Programmen, sogenannten Clients (englisch für Kunden), lokal oder über ein Netzwerk in Anspruch genommen werden kann“ [24]

<sup>14</sup> s. Fußnote 13

<sup>15</sup> „Endgerät zur Eingabe und Anzeige von Daten“ [25]

zum Kennenlernen, zur Installation und zur Unterstützung bei unbeantworteten Fragen. Letzteres bezieht sich auf Rasas Community-Forum sowie die Möglichkeit für Kunden, das Support-Team zu kontaktieren und das Learning Center zu nutzen.

Die Rasa-Dokumentation enthält eine kurze Einführung sowie Anleitungen zur Installation auf den Betriebssystemen Ubuntu, macOS und Windows. Darüber hinaus liefert sie alle Informationen, die für die Erstellung eines Rasa-Assistenten und dessen Anpassung an verschiedene Bedürfnisse erforderlich sind. Die Verwendung von visuellen Beispielen durch Rasa veranschaulicht den Austausch zwischen BenutzerIn und Assistent sowie den zugrunde liegenden Code. Es werden Schritt-für-Schritt-Anleitungen für Konversationsmuster bereitgestellt, die von den Assistenten von Rasa am häufigsten verwendet werden, wie zum Beispiel Smalltalk, FAQ, Handhabung von Geschäftslogik oder die Bewältigung von Assistentenfehlern. Rasa beschreibt bewährte Verfahren zur Entwicklung eines Assistenten, die auf eigenen Erfahrungen basieren. Dies umfasst beispielsweise den Prozess des Conversation-Driven Development, bei dem der Chatbot für NutzerInnen veröffentlicht wird und Erkenntnisse daraus verwendet werden, um den Assistenten zu verbessern. Auch das Erstellen von NLU- und Konversationsdaten wie Stories und Regeln gehört dazu.[26] Sollte die Dokumentation für manche EntwicklerInnen zu theoretisch sein, stellt Rasa das Learning Center zur Verfügung.

Dort werden die wichtigsten Konzepte von Rasa einfach und verständlich zusammengefasst, so dass die Entwicklung direkt beginnen kann. Zur Unterstützung gibt es eine Auswahl an schriftlichen und audiovisuellen Erläuterungen.[6]

Jegliche Videos aus dem Learning Center und mehr finden sich auf dem offiziellen YouTube Kanal von Rasa. Dieser beinhaltet Tutorials, Aufzeichnungen von Livestreams zu Live-Coding sowie Vorträge des Bots Berlin Events.[27]

#### **2.1.4 Aktualität**

Rasa bietet eine Zusammenstellung aller Richtlinien zur Produktfreigabe und Softwarewartung an. Diese Richtlinien zeigen, für wie lange bestimmte Versionen als unterstützt gelten und weiterhin aktiv entwickelt und gewartet werden.

Zunächst wird auf die Bezeichnung der Versionen sowie auf das Tempo der Veröffentlichung von Haupt- (major), Neben- (minor) und Patch-Versionen hingewiesen. Als Beispiel wird erläutert, wie die Version „Rasa Open Source 3.2.4“ interpretiert werden kann: Die Zahl 3 steht für die Hauptversion, die Zahl 2 für die Nebenversion und die Zahl 4 für die Patchversion. Anschließend erklärt Rasa, welche Änderungen jeweils mit den verschiedenen Versionstypen einhergehen. Etwa alle ein bis zwei Jahre werden mit der Hauptversion wesentliche Änderungen oder rückwärtskompatible Änderungen an der API oder dem Datenmodell eingeführt. Patch-Versionen hingegen werden eingesetzt, um abwärtskompatible Fehlerkorrekturen zu implementieren und falsches Verhalten zu beheben, wann immer es erforderlich ist.[28]

Die derzeitige Version 3.6.x von Rasa Open Source wird voraussichtlich bis zum 3. April 2024 gewartet und bis zum 3. Oktober 2024 für technische Unterstützung durch das Support-Team zur Verfügung stehen. Es ist anzunehmen, dass bald eine neue Version verfügbar sein wird, die genutzt werden kann. Die letzte Patch-Version von Rasa Open Source wurde am 23. Oktober 2023<sup>16</sup> veröffentlicht und behebt einen Programmfehler.[29]

---

<sup>16</sup>Stand 13.11.2023

## **2.2 Zusammenfassung**

Rasa ist ein vielseitiges, kostenloses Framework für die Entwicklung von Chatbots, das eine umfangreiche Methodenauswahl bereitstellt, um sich mit dem Thema auseinanderzusetzen. Die kompakte Einführung in NLU und Dialogverfahren sowie die Bereitstellung eines initialen Chatbot-Projekts geben einen Einblick in die Funktionsweise von Chatbots. Die Anpassungsmöglichkeiten an individuelle Anforderungen sowie die kostenlose Verfügbarkeit ohne Nutzerkonto stellen klare Vorteile dar. Ein klarer Vorteil ist die Möglichkeit, einen Bot auf eine beliebige Sprache zu trainieren, ohne eine manuelle Übersetzung vornehmen zu müssen. Auch ermöglicht die Nutzung von Forms eine dynamischere Abfrage von benötigten Daten.

Trotz dieser positiven Aspekte gibt es einige Herausforderungen im Umgang mit Rasa. Die Auffindbarkeit des kostenlosen Open-Source-Produkts auf der Website gestaltet sich schwierig, da der Fokus eher auf kostenpflichtigen Produkten liegt. Es gibt allerdings keine genaue Kostenaufstellung für Letzteres. Diese Details müssen durch eine Angebotsanforderung bei einem Experten erfragt werden. Die Nutzung über Konsole und Visual Studio Code könnte aufgrund ihrer Komplexität für Neulinge eine Herausforderung darstellen. Während der Entwicklung können Schwierigkeiten bei der Fehlerbehebung auftreten, die teilweise auf Einstellungen des Computers zurückzuführen sind. Daher erfordert es zusätzlichen Aufwand, in Foren zu suchen oder eigene Beiträge zu verfassen. Eine umfassende Dokumentation ist ebenfalls erforderlich, wobei die Bereitstellung einer großen Menge an Informationen auf einmal als überwältigend empfunden werden könnte.

### 3 Dialogflow Essentials

Dialogflow ist eine NLU-Plattform von Google Cloud, die es durch Training eines Agenten ermöglicht, die Absichten eines Benutzers zu verstehen und Antworten automatisch zu generieren. Hierbei können sowohl Text- als auch Audioeingaben analysiert werden.

Dialogflow Essentials (ES) ist die Standardversion von Dialogflow. Die Testversion ist kostenlos mit Kontingenten und Limits wie z. B. 180 Textanfragen pro Minute. Bei Überschreitung dieser Grenzwerte muss die Essentials Edition genutzt werden, die monatlich entsprechend der Anzahl der Anfragen abgerechnet wird. Zum Beispiel kostet jede Anfrage 0,002 \$.[22]

#### 3.1 Kriterien

##### 3.1.1 Benutzerfreundlichkeit

Die Erstellung eines Chatbots mit Dialogflow Essentials (ES) erfolgt nahtlos über ein Google-Konto. Für sämtliche, insbesondere unerfahrene BenutzerInnen steht eine Dokumentation als Einstiegspunkt zur Verfügung. Um eine klare Benutzerführung zu gewährleisten, werden alle verfügbaren Optionen ausführlich erklärt.

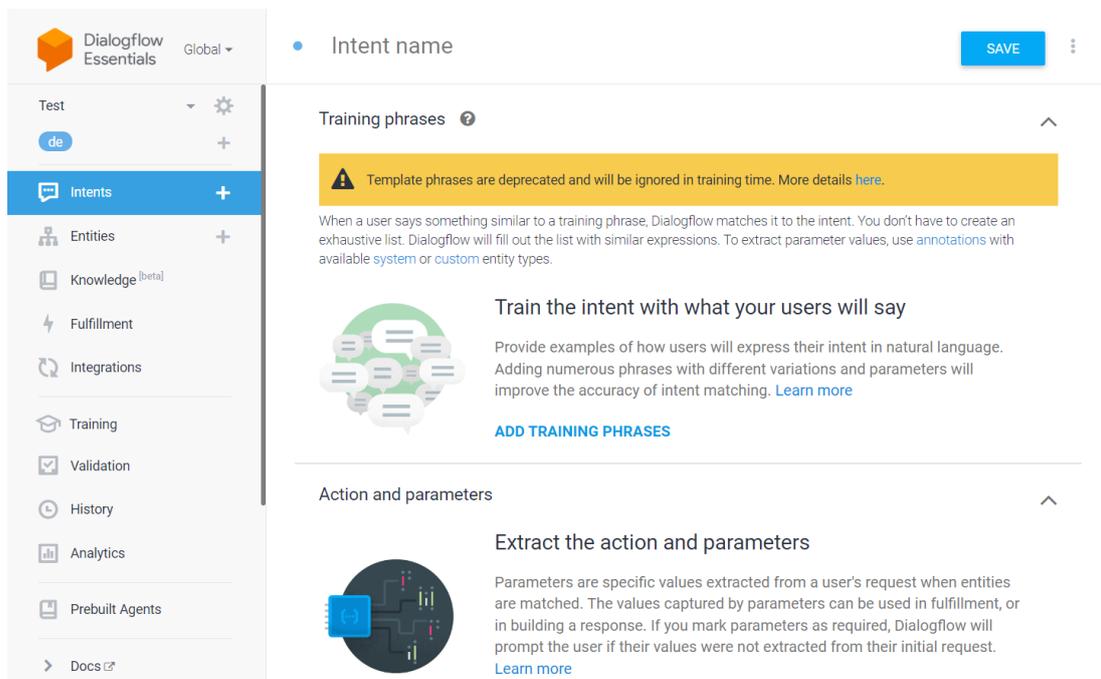


Abbildung 9: Erstellen eines Intents in Dialogflow ES

Die Verwendung von Entitäten ermöglicht eine klare Differenzierung zwischen unterschiedlichen Anforderungen wie beispielsweise dem Transportauftrag, dem Kundenauftrag oder dem Routineauftrag. Es ist möglich, Synonyme wie Routineauftrag, Routine oder Standing Order zu verwenden. Der Chatbot erkennt bereits eingebaute Entitäten wie ausgeschriebene Daten und Zahlen und speichert diese als Parameter. Bei Bedarf können mehrere Parameter einer Entität als Liste gespeichert werden, um die Flexibilität zu erhöhen.

Mithilfe von Intents können eigene Trainingsphrasen definiert werden, die während

der Konversation auftreten können. Dadurch ist es dem Chatbot möglich, ähnliche Absichten des Benutzers zu verstehen, die nicht explizit hinterlegt wurden. Aus verschiedenen Antwortmöglichkeiten wählt der Chatbot zufällig eine aus, um eine natürlichere Konversation zu erzeugen. Weiterhin können bestimmte Parameter vom Benutzer/von der Benutzerin angegeben werden, bevor der Intent als abgeschlossen betrachtet wird.

Die Nutzung von Kontexten ermöglicht es, auf Intents aufzubauen, die sich auf vorherige Nachrichten des Chatbots beziehen. Diese Technik erlaubt die Erstellung von Flows<sup>17</sup> und bietet eine Lösung für das Problem, dass Bedingungen nicht in den Prozess integriert werden können. Es kann pro Intent lediglich eine Rückmeldung realisiert werden. Dies führt dazu, dass die Erstellung von Intents unübersichtlich und kompliziert wird, da eine große Anzahl an Intents generiert wird.

Die Möglichkeit des direkten Testens in Dialogflow erleichtert die Entwicklung und Fehlerbehebung erheblich. Dabei kann sofort festgestellt werden, ob und wie der erstellte Intent sowohl aus Sicht des Entwicklers als auch aus Sicht des Chatbot-Benutzers funktioniert.

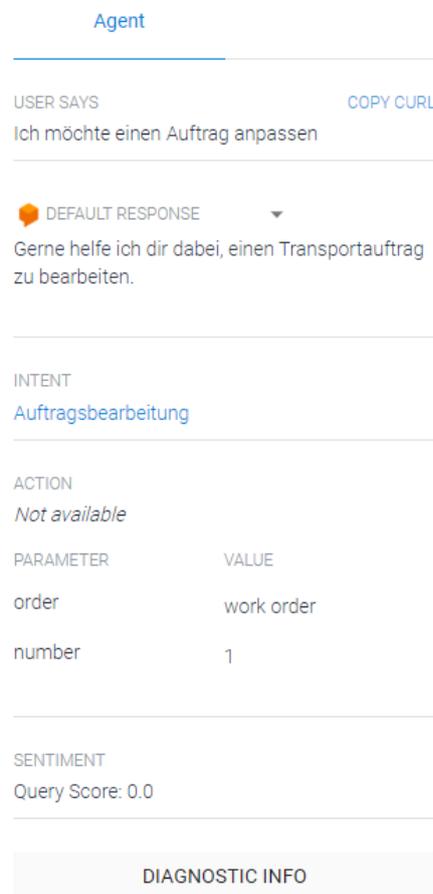


Abbildung 10: Lokales Testen des Chatbots in Dialogflow ES

Unverstandene Anfragen werden gespeichert und sind leicht abrufbar, um die Interaktion langfristig zu optimieren.

<sup>17</sup>visuelle Darstellungen einer Chatbot-Konversation mit Konversationssträngen, die zeigen, wie der/die BenutzerIn die einzelnen Stufen der Kommunikation durchlaufen kann[30]

Google bietet funktionsfähige Chatbots an, die in Kombination mit eigenen Bots genutzt werden können. Es gibt beispielsweise einen Agenten für Smalltalk und einen Agenten „Date“, der Wochentage, Feiertage und Zeiträume angeben kann. So kann Zeit gespart werden, indem diese Funktionen nicht selbst entwickelt werden müssen. Zudem bieten die vorhandenen Beispiele eine weitere Verständnishilfe für das Framework.

### 3.1.2 Integration in andere Systeme

Dialogflow ES bietet mehrere Integrationsoptionen an, um den Chatbot problemlos mit anderen Systemen zu verbinden. Auf diese Weise kann der entwickelte Chatbot auf WhatsApp, SMS, Messenger und Skype für die BenutzerInnen verfügbar gemacht werden. Dazu wird lediglich ein eigener Account<sup>18</sup> für den Chatbot benötigt.

Durch die Funktion „Fulfillment“ wird die Integration des Chatbots mit anderen Systemen ermöglicht. Wenn beispielsweise ein Transportauftrag bearbeitet werden muss, kann der Chatbot die Daten über HTTP-Requests an das System senden, wo sie verarbeitet werden. Für Fulfillment stehen zwei Tools zur Verfügung:

Der Inline-Editor von Google Cloud Functions ermöglicht eine flexible Anpassung des Chatbots. Mit der Auswahl von JavaScript (JS) oder JSON sowie einem Abrechnungskonto können benutzerdefinierte Funktionen erstellt werden, um auf die Intent-Trigger<sup>19</sup> zu reagieren. Diese Funktionen können dem/der BenutzerIn Informationen bereitstellen oder programmgesteuerte Aktionen ausführen, um die Interaktion persönlicher zu gestalten.

Webhook<sup>20</sup> bietet dieselben Funktionen auch für einen anderen Dienst, der nicht von Google stammt.

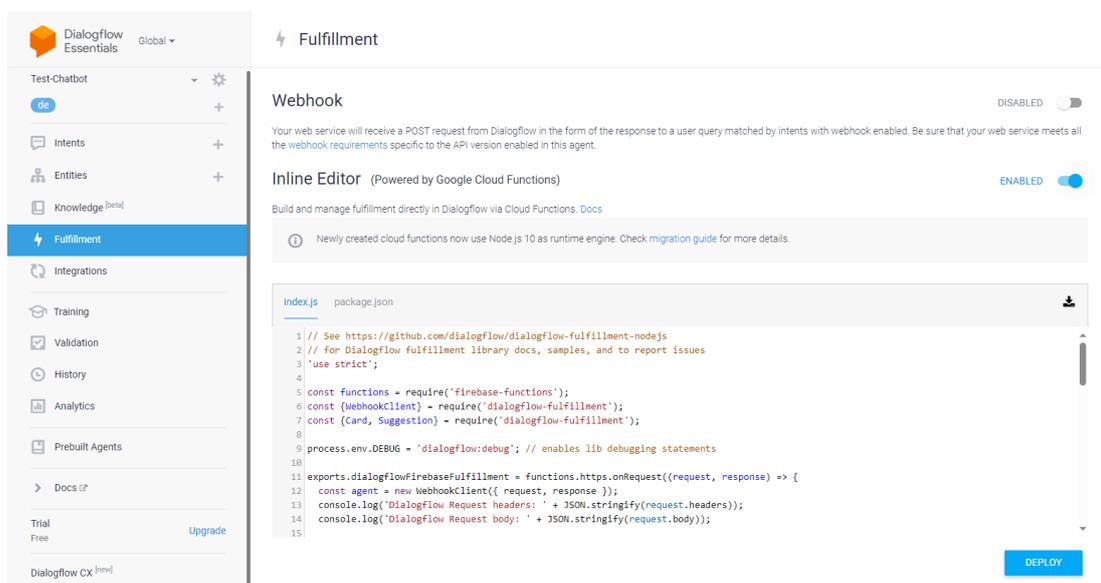


Abbildung 11: Integration mit eigenem System in Dialogflow ES

Bis zu zwei Millionen HTTP-Anfragen pro Monat können über den Service Google Cloud Functions kostenlos an einen Server gesendet werden. Für jede weitere Anfrage

<sup>18</sup>Text Messaging Dienste wie WhatsApp und SMS brauchen ein Twilio Konto

<sup>19</sup>Trigger: dt. Auslöser

<sup>20</sup>HTTP-basierte Rückmeldungsfunktion, die eine schlanke, event-gesteuerte Kommunikation zwischen zwei APIs ermöglicht[31]

fallen Kosten von 0,40 \$ an.

### 3.1.3 Dokumentation

Die Standardausgabe von Dialogflow Essentials (ES) umfasst eine umfangreiche und aktive Community, die wertvolle Unterstützung und Ressourcen bereitstellt. Zahlreiche YouTube-Kanäle bieten Erklärvideos und Tutorials zur Veranschaulichung der Verwendung von Dialogflow ES.

Die Dokumentation zu Dialogflow ES ist sehr ausführlich und bietet eine Vielzahl von Ressourcen. Sie vermitteln nicht nur die Grundlagen, sondern auch Anwendungsbeispiele, Diagramme zur visuellen Unterstützung sowie detaillierte Erklärungen aller Konzepte, wie zum Beispiel Agenten, Intents, Entitäten und mehr. Kurzanleitungen sind verfügbar, um die Einrichtung von Dialogflow ES zu erleichtern. Dies beinhaltet die Erstellung eines Agenten, die Konfiguration der Aufgabenabwicklung und die Interaktion mit Dialogflow ES in verschiedenen Szenarien, einschließlich der Integration mit anderen Plattformen und der Verwendung der API. Zusätzlich steht ein Kundenservice zur Verfügung, um Fragen zu beantworten, die über die Dokumentation und die Tutorials hinausgehen. Dies trägt zur Schaffung eines umfassenden Support-Netzwerks für die Nutzer von Dialogflow ES bei.

### 3.1.4 Aktualität

Dialogflow ES wird regelmäßig aktualisiert, um neue Funktionen zu integrieren, bestehende Probleme zu beheben und die Benutzerfreundlichkeit des Frameworks kontinuierlich zu verbessern. Das jüngste Update<sup>21</sup> wurde am 25. Oktober 2023 veröffentlicht und bringt eine Verbesserung für nicht-englischsprachige Agenten mit sich: die Aktualisierung der automatischen Sprachanpassung. Darüber hinaus stellt Google eine umfassende Anleitung zur Behebung von Problemen bei der Spracherkennung bereit, um Entwicklern und Nutzern ein reibungsloses Erlebnis zu ermöglichen.[32]

## 3.2 Zusammenfassung

Dialogflow ist ein benutzerfreundliches Framework zur Entwicklung von Chatbots, das speziell für Einsteiger durch seine intuitive Nutzung und das Prinzip des „Learning by Doing“<sup>22</sup> überzeugt. Die Dokumentation wird nur bei komplexeren Umsetzungen benötigt, wodurch der Einstieg erleichtert wird. Ein weiterer Vorteil besteht in der kostenlosen Nutzung innerhalb der behandelten Anwendungsfälle, welche lediglich einen Google Account erfordert.

Es gibt jedoch keine Möglichkeit, die erstellten Flows vom System übersetzen zu lassen. Dazu müssen alle Daten manuell in die gewünschte Sprache übersetzt werden. Dialogflow bietet viele Elemente, um einen guten Chatbot zu erstellen. Es fehlen jedoch Bedingungen innerhalb des Flows, um verschiedene Kommunikationsstränge auf Basis des Kontextes zu erstellen. Die Verwendung von Kontexten als Lösung des Problems ist zwar möglich, kann aber schnell zu Verwirrung führen. Es wird daher empfohlen,

---

<sup>21</sup>Stand 27.10.2023

<sup>22</sup>Kern der Idee ist der Erfolg beim Erlernen neuer Fähigkeiten durch eigenes Ausprobieren und Ausüben[33]

Dialogflow nicht allein, sondern in Kombination mit einem Framework zu verwenden, um die Erstellung von Flows zu erleichtern.[34]

### **3.3 Kurzer Ausblick auf Manychat**

Genau diese Anforderungen erfüllt Manychat. Als kostenloses, cloudbasiertes Chatbot-System wird es zwar hauptsächlich von Unternehmen für Chatmarketing<sup>23</sup> genutzt, bietet aber eine intuitive und visuelle Generierung von Flows. Mit der Integration von Digitalflow ermöglicht Manychat somit eine einsteigerfreundliche Erstellung von Chatbots mit KI-basiertem Sprachverständnis.

Manychat unterstützt die Nutzung auf verschiedenen Plattformen einschließlich eigener Websites, Facebook, Instagram, WhatsApp, Telegram, SMS (über Twilio) und E-Mail. Allerdings ist zur Nutzung dieser Anwendung ein Facebook-Konto erforderlich.

Das Chatbot-System verfügt über eine visuelle Flow-Erstellung, die aufgrund ihres Designs intuitiv zu bedienen ist. Ein Flow in Manychat besteht aus mehreren Blöcken, die miteinander verbunden sind und so den Ablauf einer möglichen Konversation zwischen dem Chatbot und einem/einer BenutzerIn beschreiben. Dabei kann ein Block je nach definierten Bedingungen unterschiedliche Wege nehmen. Manychat bietet diverse Chatbot-Elemente in Form von Blöcken an, welche beispielsweise eine Nachricht, eine Aktion oder eine Bedingung darstellen können. Ein Flow wird stets durch einen Auslöser gestartet, wie beispielsweise das Öffnen eines Chats mit dem Chatbot auf einer bestimmten Plattform.

Weitere Funktionen umfassen das Markieren von Chats als „nicht zugewiesen“, um sie später fortzusetzen, das Schließen von Chats, die Zuweisung von Chats an Teammitglieder sowie die Benachrichtigung von Administratoren per E-Mail, insbesondere bei Verbesserungsvorschlägen von Benutzern. Darüber hinaus ist es möglich, BenutzerInnen aus dem Chat zu entfernen und wieder hinzuzufügen. Die Optionen „Quick Replies“ und „Buttons“ dienen als vordefinierte Antwortauswahl, um verschiedene Verzweigungen im Chat-Verlauf zu erstellen.

Einer der Vorteile dieses Frameworks ist ein benutzerfreundliches Dashboard mit intuitiver visueller Flussentwicklung, das den Einstieg in die Entwicklung eines Chatbots erleichtert. Es ermöglicht das direkte Testen des Flows in der Manychat-Oberfläche oder auf dem gewünschten Kommunikationskanal. Die Plattform überprüft die vom Chatbot ausgegebenen Daten und bietet eine organisierte Übersicht. Sie unterstützt mehrere Sprachen, allerdings ist eine Übersetzungsfunktion nicht enthalten. Mit Ausnahme der Nutzung von SMS und E-Mail ist sie kostenlos und bietet Analysen zu Nutzerzahlen, versendeten Nachrichten und Klickraten auf Buttons. Manychat beinhaltet 19 vorgefertigte Templates, ermöglicht das Teilen von Templates, das Kopieren von Feeds und Elementen und bietet A/B-Testing zur Optimierung. Zusätzlich stellt das System eine umfassende Dokumentation, Kundensupport und Tutorials bereit. Außerdem verfügt es über eine große Facebook-Community mit über 100.000 Mitgliedern, die Fragen beantworten kann.[36]

---

<sup>23</sup>Ansatz zur Ansprache, Gewinnung und Bindung von Kunden mithilfe einer Messaging-Software wie dem Live-Chat, Chatbots und anderen Messaging-Kanälen[35]

## 4 Microsoft Power Virtual Agents

Microsoft Bot Framework ist eine Plattform, die es ermöglichen soll, „[Bot-EntwicklerInnen] alles zur Verfügung zu stellen, was sie benötigen, um intelligente und interaktive Bots zu entwickeln.“[37] Microsoft stellt verschiedene Produkte für Chatbot-Lösungen bereit, abhängig vom Anwendungsfall: Power Virtual Agents konzentriert sich auf die Natürlichsprachverarbeitung, Datenintegration und Multi-Channel-Unterstützung. Die Zielgruppe umfasst Fusionsteams, in denen EntwicklerInnen und ExpertInnen kooperieren, Citizen Developer<sup>24</sup> sowie spezialisierte AnbieterInnen. Die Leistungsfähigkeit der Power Virtual Agents lässt sich durch eine Vielzahl von Programmiersprachen wie .Net, JavaScript, TypeScript, Python und Java erweitern.[39, 40]

### 4.1 Kriterien

#### 4.1.1 Benutzerfreundlichkeit

Um einen eigenen Bot zu erstellen, wird eine E-Mail-Adresse von einer Firma, Schule oder Universität als Zugangsberechtigung benötigt. Diese Adresse wird genutzt, um zu überprüfen, ob die entsprechende Organisation die Verwendung von Power Virtual Agents erlaubt.

Für NutzerInnen ohne Lizenz besteht die Option, den Dienst kostenlos sieben Tage lang zu testen und diese Testphase um weitere 30 Tage zu verlängern. Nach dem Ablauf des Tests bleibt der erstellte Bot weitere 90 Tage funktionsfähig.

Nach der Erstellung eines Chatbots wird dem/der NutzerIn eine Übersicht angezeigt, welche Test- und Veröffentlichungsoptionen sowie Links zur Dokumentation und zum offiziellen YouTube-Kanal mit informativen Produktvideos enthält.

Die Erstellung von Bots wird durch die Verwendung anpassbarer Themen<sup>25</sup> unterstützt, welche individuell aktiviert oder deaktiviert werden können. Es stehen sieben vordefinierte Themen zur Verfügung, welche als Beispiele und Lektionen dienen. Aktivierte Themen werden entweder durch einen Triggerausdruck ausgelöst oder wenn von einem anderen Thema eine Umleitung an dieses Thema erfolgt. Die erste Variante erfordert fünf bis zehn verschiedene Beispielausdrücke, die der/die BenutzerIn nutzen könnte. Jene Beispiele ermöglichen dem Bot eine Intention auf der Basis von Wörtern zu erkennen und nicht auf der Basis von exakten Worten. Die erkannten Ausdrücke können in mehreren thematischen Übereinstimmungen verwendet werden, die dem/der BenutzerIn präsentiert werden. Die Priorität der Trigger kann angepasst werden.

Themen werden durch eine Abfolge von verschiedenen Schritten (Knoten/Elementen) erstellt. Dazu gehören das Senden von Nachrichten, das Stellen von Fragen, das Setzen von Bedingungen, das Verwalten von Variablen, das Wechseln zwischen Themen sowie das Senden von HTTP-Anfragen oder Ereignissen und die Verwendung selbsterstellter Elemente. EntwicklerInnen können die erstellten Themen direkt vor Ort testen, indem sie die Benutzerrolle simulieren und das System zeitgleich die verwendeten Kommunikationsstränge farblich markiert.

---

<sup>24</sup>technisch versierte MitarbeiterInnen, die IT-Anwendungen für ihren eigenen Fachbereich erstellen[38]

<sup>25</sup>Flows auf Basis von Intents (s. 3.3)

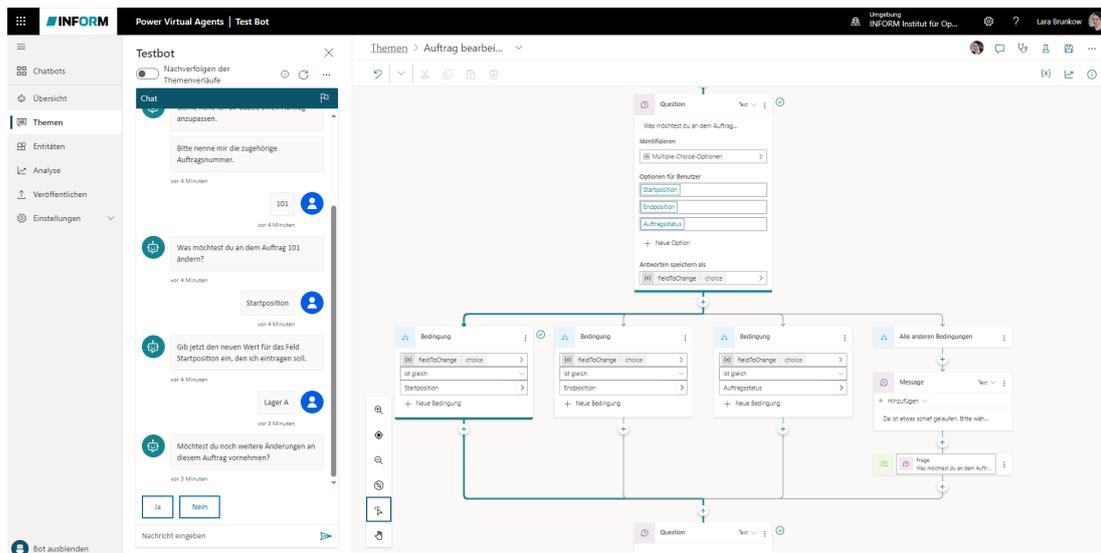


Abbildung 12: Lokales Testen des Chatbots in Microsoft Bot Framework

Microsoft bietet vorgefertigte Entitäten, z. B. Alter, Boolean<sup>26</sup>, E-Mail und Event an. Zur Erstellung von Entitäten können zwei Methoden verwendet werden: Regex und geschlossene Listen.

Geschlossene Listen bestehen aus einer Liste von Elementen, wie beispielsweise eine Entität mit dem Namen „Auftrag“ und Elementen wie „Transportauftrag“ und „Routineauftrag“, die jeweils Synonyme enthalten können. Die Option „Intelligente Übereinstimmung“ ermöglicht dem Bot, Benutzereingaben anhand von Fuzzylogik<sup>27</sup> zu interpretieren, Rechtschreibfehler zu korrigieren und die semantische Übereinstimmungslogik zu erweitern. Zusätzlich bietet Microsoft die Möglichkeit, Elemente als Antwortoptionen oder Bedingungen zu nutzen.

Die Analysefunktion in Power Virtual Agents ermöglicht die Zusammenfassung eines ausgewählten Zeitraums anhand von verschiedenen Auswertungen in Form von Diagrammen. In diesen Diagrammen werden unter anderem Gesamtsitzungen, Lösungsquoten und Abbruchquoten aufgeführt. Zusätzlich bietet Microsoft eine Auswertung der Kundenzufriedenheit an. Die Aufzeichnung der einzelnen Sitzungen erfolgt themenbezogen und kann als Excel-Datei heruntergeladen werden. Ebenso umfasst diese Funktion die Abrechnung der Sitzungen, die einen Überblick über die jeweiligen Kosten liefert.

Die Veröffentlichung des fertigen Bots umfasst dessen Aktivierung. Daraufhin kann eine Testumgebung auf einer bereitgestellten Demo-Website genutzt werden, um die einwandfreie Funktion des Bots sicherzustellen. Um den Bot einem breiteren Publikum zugänglich zu machen, können diverse Kommunikationskanäle konfiguriert werden.

#### 4.1.2 Integration in andere Systeme

Microsoft bietet verschiedene Möglichkeiten, um den Chatbot in andere Systeme zu integrieren. Dazu gehört die Bereitstellung über unterschiedliche Kanäle wie Microsoft Teams, benutzerdefinierte Websites, mobile Anwendungen, E-Mail und Twilio-Integrationen. Bei Anpassungen des Bots werden diese automatisch auf allen

<sup>26</sup>Wahrheitswert

<sup>27</sup>Ansatz zur Datenverarbeitung, der auf Wahrheitsgraden basiert und nicht auf der üblichen booleschen Logik wahr oder falsch (1 oder 0), mit der moderne Computer arbeiten[41]

Plattformen synchronisiert.[42]

Eine weitere Möglichkeit der Integration besteht darin, dass der Bot direkt HTTP-Anfragen an externe APIs sendet. Das Element ermöglicht den Abruf von Daten und die Ausführung von Aktionen, die für die Funktionalität des Bots notwendig sind. Um dies zu erreichen, wird das Knotenelement „HTTP-Anforderung senden“ genutzt. Hierfür ist die URL des gewünschten API-Endpunkts erforderlich. Es stehen verschiedene Anforderungsmethoden zur Verfügung und es bietet sich die Möglichkeit, diverse Einstellungen vorzunehmen. Die Kopfzeile kann durch die Verwendung von Schlüssel/Wert-Paaren<sup>28</sup> erweitert werden. Dabei kann entschieden werden, ob und wie (JSON- oder Rohinhalt) Informationen gesendet werden sollen. Es wird außerdem festgelegt, welches Verhalten bei einem Fehler auftritt. Der Datentyp der Antwort kann ausgewählt werden und es wird definiert, wo die HTTP-Anfrageantwort gespeichert werden soll.[44]

### 4.1.3 Dokumentation

Microsoft bietet zwei Möglichkeiten zum Einstieg in das Produkt Power Virtual Agents. Chatbot-EntwicklerInnen können zwischen einer interaktiven Tour oder dem kostenlosen Direkteinstieg in die Programmierung wählen.[45]

Die interaktive Tour gewährt einen eingehenden Einblick in die Funktionsweise von Power Virtual Agents. Sie führt anhand anschaulicher Beispiele und erläuternder Schritt-für-Schritt-Anleitungen durch den gesamten Prozess von den ersten Schritten über das Erstellen und Designen eines Chatbots bis hin zur Verwaltung eines voll funktionsfähigen Bots.[46]

Die offizielle Dokumentation zu Microsoft Power Virtual Agents bietet Online-Kurse, Dokumente und kurze Videos, die von den Grundlagen bis zur Umsetzung komplexer Funktionen reichen. Zusätzlich verweist Microsoft auf eine Support-Community und ein Virtual Agents Idea Forum, in denen sich EntwicklerInnen austauschen und voneinander lernen können. Dort finden sich auch hilfreiche Tipps zur Bot-Erstellung.[40]

Für Microsoft Power Virtual Agents existiert ein offizieller YouTube-Kanal, der Schritt-für-Schritt-Anleitungen sowie Beispiele zur Veranschaulichung der Dokumentation in Form von Videos bereitstellt.[47] Im bereitgestellten GitHub-Repository<sup>29</sup> finden sich weitere Beispiele in verschiedenen Programmiersprachen.[48]

### 4.1.4 Aktualität

Die Power Virtual Agents-Dokumentation enthält eine Liste geplanter Neuerungen und Funktionen. Diese werden in halbjährlichen Veröffentlichungszyklen angeboten und in verschiedene Kategorien wie Bot-Konfiguration, Service und Laufzeit unterteilt. Weitere Einzelheiten können in den entsprechenden Abschnitten gefunden werden. Für jeden Zyklus ist ein YouTube-Video verfügbar, welches die Highlights vorstellt und einen schnellen Überblick über die neuen Funktionen verschafft.[49]

Im Release Planner können je nach Produkt die geplanten, bevorstehenden und bereits testbaren Funktionen eingesehen werden. Die letzte Verbesserung von Power Virtual Agents fand am 17. Oktober 2023 statt.[50]

---

<sup>28</sup> „geordnete Paare (key, value), die zur Benennung von Werten dienen. Der Schlüssel (englisch: key) dient dabei als Name des zugehörigen Wertes (englisch: value).“[43]

<sup>29</sup> grundlegendes Element von GitHub zum Speichern von Code, Dateien und Revisionsverlauf

## **4.2 Zusammenfassung**

Power Virtual Agents von Microsoft präsentiert sich als vielseitiges und benutzerfreundliches Chatbot Framework, das durch die Unterstützung verschiedener Programmiersprachen und die Möglichkeit des Testens mit einem Microsoft Business Account punktet. Die kostenlose Verfügbarkeit für MitarbeiterInnen der INFORM GmbH im Rahmen des „Enterprise Agreement“ stellt einen klaren Vorteil dar. Microsofts gut strukturierte Dokumentation und die vorhandenen Lern- und Austauschmöglichkeiten erleichtern die Einarbeitung und Weiterentwicklung.

Die intuitive Bedienbarkeit und die Übersichtlichkeit des Frameworks ermöglichen eine effiziente Entwicklung, wobei die Möglichkeit, Flows während der Entwicklungsphase zu testen und Kommunikationsstränge einzusehen, den Entwicklungsprozess weiter optimiert. Die umfangreichen Analysefunktionen bieten einen zusätzlichen Mehrwert bei der Bewertung und Optimierung der Chatbot-Performance. Ein weiterer Pluspunkt ist die Fähigkeit des Bots, verschiedene Sprachen zu erkennen und entsprechend zu antworten, ohne dass redundante Daten in jeder gewünschten Sprache erstellt werden müssen.

Auf der Negativseite steht jedoch die Herausforderung, komplexe Aktionen beispielsweise das Anlegen einer Bestellung effizient zu gestalten. Die fehlende Möglichkeit, die benötigten Informationen automatisch kontextsensitiv abzufragen, erfordert wohl möglich umfangreiche Bedingungsschleifen und kann zu wiederholten Abfragen bereits gesetzter Werte führen.

## 5 Fazit und Ausblick

### 5.1 Fazit

Alle untersuchten Frameworks erfüllen die definierten Bewertungskriterien ausreichend. Jedoch zeigen sich bei einer detaillierteren Betrachtung gravierende Nachteile, die bei der Entscheidung für ein Framework bedacht werden sollten.

Im Vergleich zu den anderen Frameworks erscheint Rasa als das benutzerunfreundlichste. Das Fehlen einer benutzerfreundlichen Oberfläche mit visuellen Darstellungen der Kommunikationswege erschwert den Einstieg in die Programmierung, insbesondere für AnfängerInnen. Auch das hohe Informationsaufkommen in der Dokumentation, im Learning Center und in den Tutorialvideos erfordert eine umfangreiche Einarbeitung, um alle Aspekte und Werkzeuge der Entwicklung zu verstehen.

Dialogflow erschwert hingegen die Erstellung von Flows, einem zentralen Aspekt der Chatbot-Entwicklung. Es wird empfohlen, es nicht allein zu nutzen und sich mit anderen Frameworks wie Manychat zu verbinden, was mit zusätzlichen Kosten verbunden ist. Weitere Herausforderungen bestehen darin, dass die Nutzung mehrerer Accounts als notwendig erachtet wird, was die Verwaltung erschwert.

Der Nachteil von Microsofts Power Virtual Agents liegt vor allem darin, dass es nicht möglich ist, benötigte Informationen automatisch abzufragen. Dies führt zu einem zeitaufwändigen Prozess, da jede Information einzeln abgefragt werden muss.

Diese Einsichten betonen insgesamt die Bedeutung einer gründlichen Abwägung der Vor- und Nachteile der einzelnen Frameworks, um die optimalen Entscheidungen für die individuellen Anforderungen eines Chatbot-Projekts zu treffen.

Die Analyse der Frameworks von Rasa, Google Cloud und Microsoft ergibt, dass alle drei Tools grundsätzlich geeignet sind, die definierten Anwendungsfälle für das Softwaresystem SYNCROTESS umzusetzen. Jedoch ist Microsofts Power Virtual Agents aufgrund mehrerer überzeugender Merkmale besonders geeignet.

Power Virtual Agents von Microsoft bietet im Gegensatz zu Rasa eine besonders benutzerfreundliche Plattform mit einer übersichtlich strukturierten Benutzeroberfläche. Die intuitive Entwicklung wird durch vielseitige Lern- und Austauschmöglichkeiten unterstützt, was insbesondere für EinsteigerInnen in die Programmierung von Chatbots von Vorteil ist. Die visuelle Umsetzung des lokalen Testens ermöglicht eine effektive Überprüfung des Entwicklungsfortschritts.

Zusätzlich bietet Power Virtual Agents umfassende Analysemöglichkeiten, die die Optimierung von Chatbots erleichtern. Die kostenlose Nutzung im Rahmen des „Enterprise Agreement“ für Mitarbeitende der INFORM GmbH bietet einen deutlichen Vorteil gegenüber den anderen Frameworks. Darüber hinaus kann das Produkt allein für die Entwicklung genutzt werden, im Unterschied zu Dialogflow. Ein weiterer Vorteil ist die automatische Synchronisierung von Anpassungen am Chatbot über verschiedene Kanäle hinweg, welche eine reibungslose Aktualisierung und authentische Konsistenz sicherstellt.

Angesichts dieser Eigenschaften kann die Entscheidung für Microsofts Power Virtual Agents als das geeignete Chatbot-Framework für SYNCROTESS eine optimale Unterstützung bei der Umsetzung der definierten Anwendungsfälle bieten.

## **5.2 Ausblick**

Die vorliegende Analyse hat eine ausführliche Bewertung der Chatbot-Frameworks Rasa Open Source, Dialogflow Essentials und Microsoft Power Virtual Agents ermöglicht, einschließlich ihrer Vor- und Nachteile im Kontext der definierten Anwendungsfälle für SYNCROTESS. Dabei wurden alle relevanten Aspekte sorgfältig betrachtet.

Der derzeitige Standpunkt markiert das Ende der systematischen Analyse. Power Virtual Agents erscheint als vielversprechende Option für die Umsetzung der Anwendungsfälle. Der Fokus liegt jetzt auf den nächsten Schritten zur zielführenden Entwicklung eines Chatbots mit Power Virtual Agents.

Um eine nahtlose Integration und Kommunikation mit der vorhandenen Softwareumgebung zu gewährleisten, muss eine (REST-)Schnittstelle in SYNCROTESS implementiert werden, die relevante Daten anwendungsfallbezogen liefern kann. Diese Schnittstelle bildet die Grundlage für die reibungslose Zusammenarbeit zwischen dem Chatbot und der bestehenden Softwareumgebung.

Ein weiterer Schritt umfasst die detaillierte Ausgestaltung des Chatbot-Dialogs, welcher speziell auf die Bedürfnisse der DisponentInnen und MitarbeiterInnen von Fremdfirmen zugeschnitten ist.

Abschließend können Tests und Optimierungen des entwickelten Chatbots erfolgen, um sicherzustellen, dass er den Anforderungen der Anwendungsfälle gerecht wird. Dieser Überblick markiert den Übergang von der Analyse zu konkreten Entwicklungsschritten, die einen handlungsorientierten Ansatz für die Umsetzung der definierten Anwendungsfälle mit Microsoft Power Virtual Agents bieten.

**Literaturverzeichnis**

- [1] <https://learning.rasa.com/images/basic-prefilled-form.png>.  
Zugriffsdatum: 10.11.2023.
- [2] <https://learning.rasa.com/images/rasa-architecture-04.png>, .  
Zugriffsdatum: 10.11.2023.
- [3] Was ist ein chatbot? funktionen, vorteile und einsatzmöglichkeiten. <https://www.moin.ai/was-ist-ein-chatbot#Was-ist-ein-Chatbot>, . Zugriffsdatum: 23.10.2023.
- [4] Entität (informatik). <https://www.alexanderthamm.com/de/data-science-glossar/entitaet-informatik/>, . Zugriffsdatum: 31.10.2023.
- [5] <https://www.alexanderthamm.com/de/data-science-glossar/transformer-maschinelles-lernen/>. Zugriffsdatum: 02.11.2023.
- [6] <https://learning.rasa.com/>, . Zugriffsdatum: 13.11.2023.
- [7] Minjee Chung, Eunju Ko, Heerim Joung, and Sang Jin Kim. Chatbot e-service and customer satisfaction regarding luxury brands. *Journal of Business Research*, 117:587–595, 2020. ISSN 0148-2963. doi: <https://doi.org/10.1016/j.jbusres.2018.10.004>. URL <https://www.sciencedirect.com/science/article/pii/S0148296318304776>.
- [8] Zumstein and Dariusand Hundertmark and Sophie. Chatbots – An Interactive Technology for Personalized Communication, Transactions and Services. *IADIS International Journal on WWW/Internet*, 15:96–109, 2017.
- [9] Gruber Thorsten, Komarova Loureiro Yuliya, Solnet David. Understanding generation y and their use of social media: a review and research agenda. *Journal of Service Management*, 24:245–267, 2013.
- [10] A. Gunasekaran, H.B. Marri, R.E. McGaughey, and M.D. Nebhwani. E-commerce and its impact on operations management. *International Journal of Production Economics*, 75(1):185–197, 2002. ISSN 0925-5273. doi: [https://doi.org/10.1016/S0925-5273\(01\)00191-8](https://doi.org/10.1016/S0925-5273(01)00191-8). URL <https://www.sciencedirect.com/science/article/pii/S0925527301001918>. Information Technology/Information Systems in 21st Century Production.
- [11] Holzwarth and Martin and Janiszewski and Chris and and Neumann and Marcus M. The influence of avatars on online consumer shopping behavior. *Journal of Marketing*, 70:19–36, 2006.
- [12] Curti Lorenzo Rapp, Amon and Arianna Boldi. The human side of human-chatbot interaction: A systematic literature review of ten years of research on text-based chatbots. *International Journal of Human-Computer Studies*, 151:245–267, 2021.
- [13] Chiara Valentina Mischia, Flora Poetze, and Christine Strauss. Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science*, 201:421–428, 2022. ISSN 1877-0509. doi: <https://doi.org/10.1016/j.procs.2022.03.055>. URL <https://www.sciencedirect.com/science/article/pii/S1877050922004689>. The 13th International Conference on Ambient Systems,

- Networks and Technologies (ANT) / The 5th International Conference on Emerging Data and Industry 4.0 (EDI40).
- [14] <https://chatopenai.de/>, . Zugriffsdatum: 17.11.2023.
- [15] Anja Bethge and Thilo Lange. Wartbarkeit von software – teil 1: Dokumentation. <https://www. adesso.de/de/news/blog/wartbarkeit-von-software-teil-1-dokumentation.jsp>, 2021. Zugriffsdatum: 19.10.2023.
- [16] Makenzie Buening. Fünf best practices für die it-dokumentation in 2022. <https://www.ninjaone.com/de/blog/it-dokumentation-best-practices/>, 2023. Zugriffsdatum: 19.10.2023.
- [17] <https://www.octopus-office.de/info/bidirektionale-kommunikation/#:~:text=Als%20bidirektional%20wird%20eine%20Kommunikation,innehaben%20oder%20zumindest%20wechseln%20k%C3%B6nnen>. Zugriffsdatum: 02.11.2023.
- [18] <https://datasolut.com/was-ist-machine-learning/>. Zugriffsdatum: 17.11.2023.
- [19] <https://wirtschaftslexikon.gabler.de/definition/heuristik-34474>. Zugriffsdatum: 02.11.2023.
- [20] <https://bsh-ag.de/it-wissensdatenbank/enterprise-software/#:~:text=Bei%20einem%20unternehmenseigenen%20Enterprise%20System,Software%20als%20Anlageverm%C3%B6gen%20zu%20betrachten>, . Zugriffsdatum: 02.11.2023.
- [21] <https://domino.ai/data-science-dictionary/anaconda>. Zugriffsdatum: 02.11.2023.
- [22] Dialogflow. <https://cloud.google.com/dialogflow/docs?hl=de>, . Zugriffsdatum: 23.10.2023.
- [23] <https://www.nfon.com/de/los-gehts/cloud-telephonie/lexikon/knowledgebase-detail/fallback#:~:text=In%20der%20Informationstechnik%20bezeichnet%20der,Verhinderung%20des%20Totalausfalls%20eines%20Systems>. Zugriffsdatum: 08.11.2023.
- [24] <https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-server-ein-begriff-zwei-definitionen/>. Zugriffsdatum: 10.11.2023.
- [25] [https://de.wikipedia.org/wiki/Terminal\\_\(Computer\)#:~:text=Ein%20Terminal%2C%20auch%20als%20Konsole,Eingabe%20und%20Anzeige%20von%20Daten](https://de.wikipedia.org/wiki/Terminal_(Computer)#:~:text=Ein%20Terminal%2C%20auch%20als%20Konsole,Eingabe%20und%20Anzeige%20von%20Daten). Zugriffsdatum: 10.11.2023.
- [26] <https://rasa.com/docs/>, . Zugriffsdatum: 13.11.2023.
- [27] <https://www.youtube.com/@RasaHQ>, . Zugriffsdatum: 13.11.2023.
- [28] <https://rasa.com/rasa-product-release-and-maintenance-policy/>, . Zugriffsdatum: 13.11.2023.
- [29] <https://rasa.com/docs/rasa/changelog/>, . Zugriffsdatum: 13.11.2023.

- [30] Kevin Dees. Conversational map / conversational flow. <https://botfriends.de/blog/botwiki/conversational-map/#:~:text=Conversational%20Map%20%2F%20Conversational%20Flow&text=Bei%20einer%20Conversational%20Map%20handelt,Stufen%20der%20Kommunikation%20durchlaufen%20kann.> Zugriffsdatum: 23.10.2023.
- [31] Was ist ein webhook? <https://www.redhat.com/de/topics/automation/what-is-a-webhook#:~:text=Ein%20Webhook%20ist%20eine%20HTTP,von%20anderen%20Anwendungen%20zu%20empfangen.> Zugriffsdatum: 24.10.2023.
- [32] Versionshinweise. <https://cloud.google.com/dialogflow/docs/release-notes?hl=de,.> Zugriffsdatum: 27.10.2023.
- [33] <https://karrierebibel.de/learning-by-doing/#:~:text=Die%20Idee%20des%20Learning%20by,ausprobiert%20und%20ausge%C3%BCbt%20werden%20kann.> Zugriffsdatum: 14.11.2023.
- [34] Joren Wouters. Dialogflow review. [https://chatimize.com/reviews/dialogflow/.](https://chatimize.com/reviews/dialogflow/) Zugriffsdatum: 18.10.2023.
- [35] Chat-marketing: Beispiele, vorteile und tipps zur umsetzung. <https://www.moin.ai/chatbot-lexikon/chat-marketing-beispiele-vorteile-tipps#:~:text=Dies%20kann%20%C3%BCber%20einen%20Chat,Rabattangebote%20und%20%C3%84hnlichem%20zu%20%C3%BCbermitteln.,.> Zugriffsdatum: 23.10.2023.
- [36] [https://www.facebook.com/groups/manychat/.](https://www.facebook.com/groups/manychat/) Zugriffsdatum: 30.10.2023.
- [37] Stephan Bisser. Einführung in das microsoft bot framework. [https://link.springer.com/chapter/10.1007/978-3-662-66472-8\\_2.](https://link.springer.com/chapter/10.1007/978-3-662-66472-8_2.) Zugriffsdatum: 24.10.2023.
- [38] Was ist ein citizen developer? <https://www.02100.com/ratgeber/was-ist-ein-citizen-developer#:~:text=Der%20Begriff%20Citizen%20Developer%20bedeutet,f%C3%BCr%20ihren%20eigenen%20Fachbereich%20erstellen.> Zugriffsdatum: 24.10.2023.
- [39] Microsoft power virtual agents-dokumentation. [https://learn.microsoft.com/de-de/power-virtual-agents/.](https://learn.microsoft.com/de-de/power-virtual-agents/) Zugriffsdatum: 30.10.2023.
- [40] [https://learn.microsoft.com/de-de/power-virtual-agents/.](https://learn.microsoft.com/de-de/power-virtual-agents/) Zugriffsdatum: 13.11.2023.
- [41] Fuzzy logic. <https://www.computerweekly.com/de/definition/Fuzzy-Logic.> Zugriffsdatum: 30.10.2023.
- [42] <https://learn.microsoft.com/de-de/power-virtual-agents/publication-fundamentals-publish-channels?tabs=web#configure-channels,> Zugriffsdatum: 13.11.2023.
- [43] <https://glossar.hs-augsburg.de/Schl%C3%BCssel/Wert-Paar.> Zugriffsdatum: 13.11.2023.
- [44] <https://learn.microsoft.com/de-de/power-virtual-agents/authoring-http-node,> Zugriffsdatum: 13.11.2023.

- [45] [https://powervirtualagents.microsoft.com/de-de/?ef\\_id=\\_k\\_Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL\\_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw\\_wcB\\_k\\_&OCID=AIDcmm8g1tzn46\\_SEM\\_\\_k\\_Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL\\_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw\\_wcB\\_k\\_&gad=1&gclid=Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL\\_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw\\_wcB](https://powervirtualagents.microsoft.com/de-de/?ef_id=_k_Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw_wcB_k_&OCID=AIDcmm8g1tzn46_SEM__k_Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw_wcB_k_&gad=1&gclid=Cj0KCQiAr8eqBhD3ARIsAIe-buPk2h2T79Chn45ZhAUL_Zj2RyBubukm3dpMNFdyU1tIXHGL0kpp51YaAjR4EALw_wcB), . Zugriffsdatum: 13.11.2023.
- [46] <https://powervirtualagents.microsoft.com/en-us/guidedtour/power-platform/power-virtual-agents/1/1/>, . Zugriffsdatum: 13.11.2023.
- [47] <https://www.youtube.com/channel/UCn8PsB1VsGUt8J3tZCKsMPA>, . Zugriffsdatum: 13.11.2023.
- [48] <https://github.com/microsoft/botbuilder-samples#readme>, . Zugriffsdatum: 13.11.2023.
- [49] <https://learn.microsoft.com/de-de/power-platform/release-plan/2023wave2/power-virtual-agents/>, . Zugriffsdatum: 13.11.2023.
- [50] <https://releaseplans.microsoft.com/de-de/?app=Power+Virtual+Agents&status=new>, . Zugriffsdatum: 13.11.2023.