zakI.D
Zentrum für angewandte Künstliche Intelligenz Duisburg

Fraunhofer IMS

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

KROHNE

Federführung: Fraunhofer-Institut für Mikroelektronische Schaltungen und Systeme IMS
Krohne Messtechnik GmbH
Universität Duisburg-Essen

Abschlussbericht MARA
(Modularer Anruf- und Routingassistent)

29.10.2026

Koordination:
Ole Werger (Projektleitung)

Fraunhofer IMS

Telefon 0203/3783-281
E-Mail ole.werger@ims.fraunhofer.de

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

Die Landesregierung Nordrhein-Westfalen

5-StandorteProgramm

Inhalt

1 Zusammenfassung 3

2 Wissenschaftliche Ergebnisse 4

2.1 Projektergebnisse und Zielerreichung 4

2.1.1 AP1 Systementwurf 4

2.1.2 AP2 Modellierung 4

2.1.3 AP3 Integration und Routing 5

2.2 Lösungskonzept 6

2.2.1 Allgemeines Vorgehen 6

2.2.2 Systemarchitektur 6

2.2.3 Modell- und Servingkonzept 8

2.2.4 Dialogsteuerung und Routing 11

2.2.5 Wissensbasierte Beantwortung 13

2.2.6 SQL-gestützte Serviceauskünfte 14

2.2.7 Sprachschnittstelle 14

2.2.8 Bedienoberfläche 15

2.2.9 Qualitätssicherung und technische Grenzen 16

2.3 Verwertung 18

3 Zusammenarbeit 20

4 Einordnung im ZaKI.D-Kontext 21

5 Glossar 22

1 Zusammenfassung

MARA („Modularer Anruf- und Routingassistent“) ist ein Proof of Concept (PoC) für die automatisierte Unterstützung einer Kundenhotline der Walzer Elektronik Vertriebsgesellschaft mbH. Ausgangspunkt sind wiederkehrende Serviceanfragen, bei denen Informationen aus unterschiedlichen Quellen zusammengetragen werden müssen. Dazu gehören allgemeine Fragen zu Produkten und Serviceabläufen ebenso wie konkrete Auskünfte zu Ersatzteilen, Preisen, Lieferstatus, Reparaturen und Aufträgen. Ziel des Projekts war es zu untersuchen, wie solche Anliegen sprachbasiert aufgenommen, fachlich eingeordnet und innerhalb eines zusammenhängenden Dialogs beantwortet werden können.
Hierfür wurde ein mehrstufiges Lösungskonzept entwickelt. Eine Anfrage wird zunächst als Text erfasst oder aus einer Spracheingabe transkribiert. Anschließend klassifiziert eine KI-Komponente das Anliegen in eine vordefinierte Bearbeitungsroute, beispielsweise FAQ-Beantwortung, konkrete Datenbankabfrage, Authentifizierung, Rückfrage, Übergabe an Mitarbeitende, Behandlung nicht unterstützter oder fachfremder Anliegen sowie Klärung unverständlicher Eingaben. Sie entscheidet noch nicht über die endgültige Antwort, sondern beschreibt strukturiert, welche Art von Information benötigt wird, ob Angaben fehlen und ob der aktuelle Satz ein neues Anliegen, eine Rückfrage oder eine Korrektur darstellt. Eine regelbasierte Dialogsteuerung prüft diese Entscheidung und legt den nächsten Verarbeitungsschritt fest.

Für die Informationsbeschaffung wurden zwei getrennte Wege umgesetzt. Allgemeine Service- und Produktfragen werden über eine semantisch durchsuchbare FAQ-Wissensbasis bearbeitet. Dabei werden klassische Textsuche und Ähnlichkeitssuche kombiniert und die gefundenen Inhalte nach ihrer Relevanz sortiert. Konkrete Auskünfte, etwa zu Preis, Verfügbarkeit, Ersatzteilen oder einem Auftragsstatus, werden dagegen über fest definierte Abfragefunktionen aus strukturierten Daten ermittelt. Auf diese Weise werden frei formulierte Kundenanliegen mit nachvollziehbaren Faktenquellen verbunden. Ein Gesprächszustand hält fest, welches Anliegen aktuell bearbeitet wird, welche Angaben bereits bestätigt wurden und ob eine Rückfrage, Authentifizierung oder Übergabe aussteht. Personenbezogene Reparatur- und Auftragsinformationen werden erst nach einem vorgeschalteten Authentifizierungsschritt abgefragt. Nicht unterstützte Vorgänge werden als solche kenntlich gemacht. Wenn eine anrufende Person einen menschlichen Ansprechpartner verlangt, fragt das System zunächst nach einer Bestätigung und erzeugt anschließend eine kurze Zusammenfassung für die Übergabe.
Die Sprachschnittstelle besteht aus Spracherkennung, Sprachsynthese und einer Unterbrechungslogik. Eingehendes Audio wird in zusammenhängende Äußerungen unterteilt und in Text umgewandelt. Die erzeugte Antwort kann anschließend wieder als Sprache ausgegeben werden. Beginnt die anrufende Person während der Ausgabe erneut zu sprechen, kann die laufende Sprachausgabe unterbrochen und die neue Eingabe verarbeitet werden. Für die Darstellung und Erprobung wurde eine browserbasierte Oberfläche entwickelt, die sowohl Text- als auch Mikrofoneingaben unterstützt und den Gesprächsverlauf sichtbar macht.

Eine reale Telefonieanbindung und die Weiterleitung über einen Telefonieanbieter sind nicht Bestandteil des entwickelten Systems.

2 Wissenschaftliche Ergebnisse

2.1 Projektergebnisse und Zielerreichung

Im Projekt MARA sollte nachgewiesen werden, dass wiederkehrende Anfragen an eine Kundenhotline automatisiert aufgenommen, eingeordnet und mithilfe unterschiedlicher Informationsquellen beantwortet werden können. Neben allgemeinen Wissensfragen sollten auch konkrete Serviceauskünfte und mehrstufige Gesprächsverläufe berücksichtigt werden.
Dazu wurde der Gesamtprozess in die drei Arbeitspakete Systementwurf, Modellierung sowie Integration und Routing gegliedert. Die Arbeitspakete bilden den Weg von der fachlichen und technischen Struktur über die Auswahl und Ausgestaltung der KI-Komponenten bis zur gemeinsamen Anwendung ab.

2.1.1 AP1 Systementwurf

Entwicklung eines Systemkonzepts für die Aufnahme, Verarbeitung und Beantwortung eingehender Serviceanfragen. Dazu wurden die benötigten Funktionsbereiche festgelegt und zu einer Verarbeitungskette verbunden. Zusätzlich wurde ein Testkonzept für Sprachverarbeitung, Wissenszugriff, Dialogführung und Antwortausgabe erstellt.

Zielerreichung: Das Gesamtsystem wurde in die sechs Bereiche Benutzerschnittstelle, Sprachverarbeitung, Dialogsteuerung, Wissenssuche, Zugriff auf strukturierte Servicedaten und Antworterzeugung unterteilt. Für den Austausch zwischen diesen Bereichen wurden einheitliche Nachrichten und ein gemeinsamer Gesprächszustand definiert. Damit wurden alle sechs vorgesehenen Funktionsbereiche sowie die vier vorgesehenen Prüfbereiche des Testkonzepts umgesetzt.

Ergebnis: Es wurde ein modulares Systemkonzept entwickelt (siehe Abschnitt 2.2.2). Alle vorgesehenen Funktions- und Prüfbereiche wurden konzeptionell umgesetzt, dies entspricht einer konzeptionellen Zielerreichung von 100 %. Text- und Spracheingaben verwenden denselben fachlichen Ablauf. Die einzelnen Komponenten können getrennt betrachtet und geprüft werden. Modellbasierte Entscheidungen, Faktenzugriff und regelbasierte Schutzmechanismen sind voneinander abgegrenzt.

2.1.2 AP2 Modellierung

Auswahl und Ausgestaltung der KI-Komponenten für Spracherkennung, Einordnung des Kundenanliegens, semantische Wissenssuche, Antwortformulierung und Sprachausgabe. Ein Schwerpunkt lag auf der Frage, welche Aufgaben durch Modelle übernommen und welche Entscheidungen durch feste Regeln abgesichert werden.

Zielerreichung: Für die Einordnung eines Anliegens wurde ein strukturiertes Ausgabeformat entwickelt. Die Wissenssuche in der FAQ-Wissensbasis verbindet lexikalische und semantische Suchverfahren mit einer nachgelagerten Relevanzbewertung. Für die strukturierten Geschäftsdaten, die in einer SQL-Datenbank abgelegt sind, wird eine passende SQL-Funktion ausgewählt und ausgeführt. Für Antworterzeugung, Spracherkennung und Sprachausgabe wurden jeweils getrennte Modellaufgaben definiert und getestet.

Ergebnis: Die Modelle liefern klar begrenzte Zwischenergebnisse, die vor der weiteren Verwendung geprüft werden. Im FAQ-Test wurden Von 701 FAQ-Testanfragen wurden 596 dem vorgesehenen FAQ-Pfad zugeordnet (85,02 %). Für die Teilmenge der korrekt zugeordneten Anfragen mit erzeugter Suchfrage wurde anschließend die Suchqualität bewertet: In 92,45 % dieser Fälle lag der erwartete FAQ-Inhalt unter den ersten drei Produktionstreffern. Diese Kennzahl bewertet das Auffinden des Inhalts, nicht die fachliche Richtigkeit der erzeugten Antwort. Bei den strukturierten Geschäftsdaten-Abfragen waren 1.665 von 1.727 Abfragen erfolgreich, dies entspricht einer Retrieval-Passrate von 96,41 %. Die Sprachtests ergaben eine sprachformbezogene Wortfehlerrate von 14,75 % sowie eine Verständlichkeitsrate der Sprachausgabe von 94,12 %.

2.1.3 AP3 Integration und Routing

Zusammenführung der entwickelten Komponenten zu einem vollständigen Dialogsystem. Das Routing entscheidet zwischen FAQ-Wissen, strukturierten Serviceauskünften, Rückfragen, nicht unterstützten Vorgängen und der Übergabe an einen Menschen. Dabei müssen Folgefragen und Korrekturen dem laufenden Anliegen zugeordnet werden.

Zielerreichung: Sprach- und Texteingabe, Klassifikation des Anliegens, Faktenzugriff, Zustandssteuerung, Antworterzeugung und Sprachausgabe wurden in einer gemeinsamen Verarbeitungskette verbunden. Für geschützte Auskünfte wurde ein Authentifizierungsschritt integriert. Der Übergabepfad erzeugt nach Bestätigung eine Zusammenfassung des bisherigen Gesprächs.

Ergebnis: Das entwickelte System bearbeitet allgemeine Wissensfragen und die vorgesehenen Serviceauskünfte über mehrere Dialogschritte. In der Ende-zu-Ende-Evaluation bestanden 25 von 30 vollständigen Testszenarien mit drei Personen (83,33 %). Von den bewerteten Antworten wurden 96,43 % als hilfreich eingestuft. Die Bedienung erfolgt über eine browserbasierte Oberfläche mit Text- und Spracheingabe.

2.2 Lösungskonzept

2.2.1 Allgemeines Vorgehen

Die intelligente Telefonassistenz MARA trennt vier Aufgaben: *Verstehen*, *Entscheiden*, *Nachschlagen* und *Formulieren*. Ein Sprachmodell interpretiert die Eingabe und schlägt eine definierte Bearbeitungsroute vor. Die regelbasierte Dialogsteuerung prüft diesen Vorschlag und legt den nächsten Schritt fest. Danach werden Fakten aus der zuständigen Quelle abgerufen. Erst auf dieser Grundlage formuliert das Sprachmodell in einem zweiten Aufruf, allerdings mit anderer Anweisung und Kontext Information, die Antwort.
Das Sprachmodell verwaltet weder den Gesprächszustand noch den Datenzugriff. Für jede Informationsart ist festgelegt, welche Quelle verwendet werden darf, welche Angaben benötigt werden und ob eine Authentifizierung erforderlich ist.

Technischer Ablauf eines Dialogschritts

Die Verarbeitung ist als Folge expliziter Verträge aufgebaut. Jede Stufe erhält nur die für ihre Aufgabe notwendigen Daten und liefert ein prüfbares Zwischenergebnis. Dadurch werden probabilistische Modellentscheidungen von deterministischen Zustands- und Datenzugriffsregeln getrennt.

Stufe	Technischer Vertrag
Eingabe	Text wird direkt übernommen; Audio wird durch Voice Activity Detection (VAD) segmentiert und anschließend durch Speech-to-Text (STT) transkribiert.
Controller	Ein Sprachmodell erzeugt genau eine strukturierte Routingentscheidung mit Intent, Beziehung zum laufenden Anliegen, Suchfrage, fehlenden Angaben und optionalem freigegebenem Werkzeugaufruf.
Prüfung und Kontext	Parser und Resolver normalisieren Kennungen, verwerfen unzulässige Kombinationen und beziehen Kontext ausschließlich aus einem CallState, FAQ-Retrieval, freigegebenen SQL-Funktionen (Zugriff auf die Geschäftsdaten) oder dem Authentifizierungspfad.
Zustandsübergang	Eine deterministische Zustandsmaschine setzt die wirksame Aktion fest, zum Beispiel Antworten, Rückfragen, Authentifizieren oder Übergabe anbieten.
Ausgabe	Das Sprachmodell formuliert innerhalb des freigegebenen Faktenkontexts und des Aktionsvertrags. Text wird gestreamt; bei aktivierter Ausgabe folgt Text-to-Speech (TTS) als PCM-Audiostream.

2.2.2 Systemarchitektur

MARA ist als modulare Verarbeitungskette aufgebaut. Ein- und Ausgabe, Dialogsteuerung, Informationsbeschaffung, Antworterzeugung und Modellbereitstellung sind voneinander getrennt. Dadurch übernimmt das Sprachmodell nicht den gesamten Ablauf, sondern liefert klar begrenzte Zwischenergebnisse. Text- und Spracheingaben verwenden nach der Transkription denselben fachlichen Verarbeitungspfad.

Abbildung 1 zeigt den Datenfluss durch das System. Die Bedienoberfläche nimmt Text oder Mikrofonaufnahmen entgegen und gibt Transkript, Antwort, Gesprächsstatus und optional Audio aus. Spracheingaben werden zunächst durch eine Sprachaktivitätserkennung in zusammenhängende Äußerungen unterteilt und anschließend mittels Speech-to-Text transkribiert. Danach übergibt die Oberfläche den Text an die zentrale Dialogsteuerung.

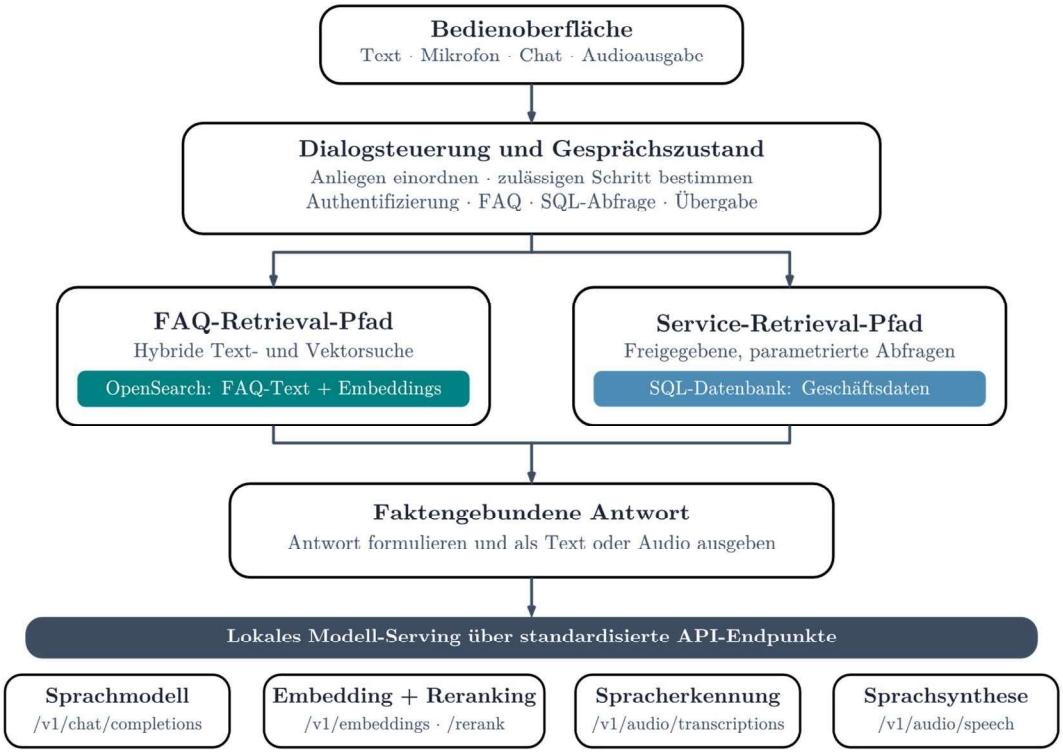


Abbildung 1: Konzeptioneller Aufbau der MARA Systemarchitektur

Die Dialogsteuerung ordnet die Eingabe dem laufenden Gespräch zu. Das Sprachmodell erzeugt dazu einen strukturierten Entscheidungsvorschlag mit Art des Anliegens, Bezug zum bisherigen Dialog, Suchfrage oder Abfrageart sowie gegebenenfalls erkannten und fehlenden Kennungen. Eine deterministische Zustandsmaschine prüft diesen Vorschlag und legt die tatsächlich zulässige Aktion fest. Der Gesprächszustand speichert das aktuelle Anliegen, bestätigte Angaben, offene Rückfragen, ausstehende Authentifizierungen und bereits gefundene Geräte- oder Artikeldaten. Folgefragen können dadurch auf vorherige Ergebnisse zurückgreifen, ohne dieselbe Abfrage erneut auszuführen.

Für die Informationsbeschaffung existieren zwei getrennte Pfade. Allgemeine Produkt-, Service- und Prozessfragen werden über den FAQ-Retrieval-Pfad bearbeitet. Dieser kombiniert lexikalische Textsuche und semantische Vektorsuche in OpenSearch und bewertet die Treffer anschließend mit einem Reranker. Konkrete Auskünfte zu Ersatzteilen, Preisen, Verfügbarkeit, Kompatibilität, Reparaturen oder Aufträgen werden über den Service-Retrieval-Pfad ermittelt. Das Sprachmodell erzeugt dabei kein SQL, sondern wählt ausschließlich eine freigegebene Abfrageart. Dadurch wird der Datenbankzugriff auf vorab definierte Operationen beschränkt und das Risiko unzulässiger oder manipulierter Abfragen, etwa durch SQL-Injection, reduziert. Die Anwendung prüft deren Parameter und führt eine parametrisierte Datenbankfunktion aus.

Beide Pfade liefern einen begrenzten Faktenkontext an die Antworterzeugung. Das Sprachmodell wird über den Systemprompt angewiesen nur aus diesen Informationen und aus bestätigten Gesprächsdaten eine Antwort zu formulieren. Die erzeugte Antwort wird als Text gestreamt und kann zusätzlich durch Text-to-Speech als Audio ausgegeben werden. Sprachmodell, Embedding, Reranking, Spracherkennung und Sprachsynthese laufen als getrennte lokale Dienste hinter standardisierten API-Endpunkten. Die unterste Ebene der Abbildung verdeutlicht damit, dass die fachliche Anwendung von den konkreten Modellimplementierungen entkoppelt ist und einzelne Modelle nach erneuter Evaluation ausgetauscht werden können. Diese Dienste sind in der untersten Ebene der Abbildung mit ihren jeweiligen API-Endpunkten dargestellt. Die farbig hinterlegten Zeilen der beiden Retrieval-Pfade benennen dagegen die Datenhaltungssysteme OpenSearch und SQL-Datenbank.

2.2.3 Modell- und Servingkonzept

Das Modell- und Servingkonzept beschreibt die Auswahl der KI-Modelle und ihre technische Bereitstellung als aufrufbare Dienste. Für jede Modellrolle wurde nach öffentlich verfügbaren Modellgewichten mit einer freien und grundsätzlich kommerziell nutzbaren Lizenz gesucht. Alle ausgewählten Modelle werden unter Apache 2.0 veröffentlicht. Das ermöglicht lokale Tests und eine spätere lokale Bereitstellung. Vor einem Produktbetrieb bleibt trotzdem eine eigene Lizenz- und Compliance-Prüfung erforderlich.
Zusätzlich sollte jedes Modell unter den Randbedingungen seiner Aufgabe nahe am aktuellen Stand der Technik liegen. Diese Randbedingungen unterscheiden sich: Beim Sprachmodell zählen Instruktionsverständnis, strukturierte Ausgabe und Speicherbedarf. Für Embedding und Reranking sind mehrsprachige Retrievalqualität und kurze Antwortzeiten wichtiger. Bei der Spracherkennung zählt die Fehlerrate für Deutsch; bei der Sprachsynthese zählen Verständlichkeit, Stimme und Audiostreaming. Gemeint ist deshalb keine allgemeine Bestenliste, sondern eine zum Projektabschluss leistungsstarke Auswahl, die sich innerhalb der jeweiligen Größen-, Lizenz- und Betriebsgrenzen möglichst nahe am Stand der Technik bewegt.

Zentrale Sprachverarbeitung – Qwen3.6-35B-A3B-FP8

Dieses Modell übernimmt die zuvor beschriebenen Aufgaben: Es klassifiziert eingehende Anliegen, erzeugt das festgelegte Entscheidungsschema, formuliert Antworten auf Grundlage der freigegebenen Fakten und erstellt Übergabezusammenfassungen für die menschliche Hotline. Qwen3.6-35B-A3B verfügt über rund 35 Milliarden Parameter, von denen aufgrund der Mixture-of-Experts-Architektur pro Token etwa 3 Milliarden aktiviert werden. Dadurch verbindet das Modell eine hohe Gesamtkapazität mit einem deutlich geringeren Rechenaufwand als ein dichtes Modell vergleichbarer Größe. Die verwendete FP8-Variante reduziert zudem den Speicherbedarf gegenüber BF16 erheblich, während die veröffentlichten Leistungswerte laut Modellkarte nahezu unverändert bleiben. Die vorhandenen multimodalen Fähigkeiten werden im Projekt nicht genutzt. [1]
Die Modellauswahl folgte dem Prinzip „so groß wie nötig, so klein wie möglich“. Kleinere Modelle im Bereich von etwa 7 bis 14 Milliarden Parametern erwiesen sich in den projektinternen Vorversuchen für den vorgesehenen Einsatz als nicht zuverlässig genug. Schwächen zeigten sich insbesondere bei der konsequenten Einhaltung des vorgegebenen Ausgabeschemas, der Beschränkung auf freigegebene Fakten und bei mehrstufigen Dialogen. Eine höhere Modellkapazität erwies sich daher für den Prototyp als sinnvoll.

Gleichzeitig wurde bewusst eine einzelne Grafikkarte mit 48 GB Videospeicher als Zielplattform gewählt. Diese Speicherklasse bietet für den vorgesehenen Betrieb einen vertretbaren Kompromiss aus Anschaffungskosten, Leistungsfähigkeit und Integrationsaufwand. Grafikkarten mit deutlich mehr Speicher sind zwar verfügbar, gehen jedoch mit einem erheblichen Preissprung einher. Die Modellauswahl wurde daher nach oben durch die Vorgabe begrenzt, den Betrieb weiterhin auf einer einzelnen GPU dieser Klasse zu ermöglichen. Qwen3.6-35B-A3B passt in seiner FP8-Variante gut zu diesen Randbedingungen. Der veröffentlichte FP8-Checkpoint umfasst rund 37,5 GB und lässt auf einer 48-GB-GPU noch ausreichend Speicher für Laufzeitdaten und den bei der Verarbeitung laufender Gespräche benötigten Cache. Dieser zusätzliche Speicher ist insbesondere bei längeren Kontexten und mehreren gleichzeitig verarbeiteten Sitzungen relevant. Damit kann ein Modell dieser Größenordnung betrieben werden, ohne den verfügbaren Videospeicher bereits vollständig durch die Modellgewichte auszuschöpfen.

Qwen3.6 wurde im Verlauf der Projektverarbeitung veröffentlicht. Da es die gesetzten Anforderungen innerhalb derselben Hardwareklasse erfüllte und gegenüber früher verfügbaren Modellen eine höhere Leistungsfähigkeit bietet, bildet es damit den zum Zeitpunkt des Projektabschlusses aktuellen Stand innerhalb der gesetzten technischen Randbedingungen ab. Die Entscheidung für ein Modell dieser Größe bedeutet nicht, dass 35 Milliarden Parameter für einen späteren Produktivbetrieb grundsätzlich erforderlich sind. Bei einem stabilen und klar abgegrenzten Aufgabenprofil wäre ein kleineres, gezielt auf den Anwendungsfall angepasstes Modell ein naheliegender Ansatz, um Hardwarebedarf und Betriebskosten weiter zu reduzieren.

Semantische Suche – Qwen3-Embedding-0.6B

Das Modell wandelt Kundenfragen und FAQ-Inhalte in Vektoren um. Ähnliche Bedeutungen liegen im Vektorraum näher beieinander. Das Modell unterstützt mehr als 100 Sprachen, Kontexte bis 32.000 Token und anpassbare Vektordimensionen. Mit 0,6 Milliarden Parametern bleibt es für häufige Suchanfragen vergleichsweise sparsam. In der veröffentlichten mehrsprachigen MTEB-Auswertung erreicht es für diese Größenklasse ein hohes Retrievalniveau. Damit passte es zu den Anforderungen an deutsche Suchqualität, kurze Latenz und lokalen Betrieb. [2]

Relevanzbewertung – BAAI bge-reranker-v2-m3

Der Reranker liest die Suchfrage und einen FAQ-Kandidaten gemeinsam und vergibt dafür einen Relevanzwert. Das erfasst feine Bedeutungsunterschiede besser als der reine Abstand zweier vorher berechneter Vektoren, kostet aber mehr Rechenzeit. Deshalb verarbeitet der Reranker nur die bereits gefundene Kandidatenmenge. Das Modell ist mehrsprachig, rund 0,6 Milliarden Parameter groß und für schnelle Inferenz ausgelegt. Es ergänzt die schnelle Suche um eine genauere letzte Auswahl. [3]

Spracherkennung – Cohere Transcribe 03-2026

Das Audio-zu-Text-Modell besitzt 2 Milliarden Parameter und unterstützt Deutsch sowie 13 weitere Sprachen. Zum Veröffentlichungszeitpunkt erreichte es auf dem offenen ASR-Leaderboard die beste durchschnittliche englische Wortfehlerrate der dort verglichenen offenen Modelle. Dieser Wert ist kein Nachweis für deutsche Hotline-Sprache. Deshalb waren die eigenen deutschsprachigen Sprachtests das zweite Auswahlkriterium. Hinzu kamen der überschaubare Ressourcenbedarf und die direkte Unterstützung durch vLLM. [4]

Sprachausgabe – Qwen3-TTS-12Hz-1.7B-Base

Das Modell unterstützt Deutsch, Streaming und die Übernahme einer Stimme aus einer kurzen Referenzaufnahme. Es kann erste Audioblöcke ausgeben, bevor der vollständige Text vertont ist. Das verkürzt die wahrgenommene Wartezeit im Dialog. Die Base-Variante wurde genutzt, um eine gleichbleibende Projektstimme aus einer freigegebenen Referenz zu erzeugen. Für einen späteren Einsatz müssen Einwilligung, Nutzungsrechte und Schutz vor missbräuchlicher Stimmnachbildung geregelt werden. [5]

Standardisierte Modellendpunkte

Jedes Modell läuft als eigener HTTP-Dienst. Die fachliche Anwendung lädt keine Modellgewichte und verwendet keine modellspezifischen Klassen. Sie kennt nur die Adresse des Dienstes, die Modellkennung und den API-Vertrag. Die Verträge orientieren sich an verbreiteten OpenAI-kompatiblen Schnittstellen:

Modellrolle	API-Endpunkt	Ein- und Ausgabe
Sprachmodell	/v1/chat/completions	Nachrichten hinein; Text- oder strukturierte Ausgabe hinaus.
Embedding	/v1/embeddings	Text hinein; numerischer Vektor hinaus
Reranking	/rerank	Suchfrage und Kandidaten hinein; Relevanzwerte hinaus
Spracherkennung (STT)	/v1/audio/transcriptions	Audiodaten hinein; Transkript hinaus
Sprachsynthese (TTS)	/v1/audio/speech	Text und Stimme hinein; gestreamte Audioblöcke hinaus

Serving mit vLLM und vLLM-Omni

vLLM stellt im entwickelten System Chat Completions, Embeddings, Reranking und Audio-Transkription bereit. vLLM-Omni übernimmt die mehrstufige Sprachsynthese. Beide Frameworks kapseln das Laden der Gewichte, die GPU-Ausführung, die Anfragebündelung und das Streaming hinter den genannten Endpunkten. vLLM wurde gewählt, weil es unter anderem PagedAttention, kontinuierliches Batching, Prefix-Caching, Quantisierung und gestreamte Antworten unterstützt. Diese Funktionen senken den Speicherbedarf und verbessern die Auslastung bei mehreren gleichzeitigen Anfragen. [6][7][8]

Modell, Serving-Framework und Anwendung sind getrennte Ebenen. Erscheint ein besseres Modell, kann der Inferenzdienst dasselbe API-Format mit einer neuen Modellkennung anbieten. Unterstützt ein anderes Inferenz-Framework dieselben Endpunkte, kann auch vLLM ersetzt werden. Die Dialogsteuerung, die Wissenslogik und die Bedienoberfläche müssen dafür nicht neu gebaut werden. Kleinere Abweichungen im Anfrageformat werden in einem schmalen Serving-Adapter gekapselt. Lokal betriebene Endpunkte halten Audio, Gesprächsverlauf und Geschäftsdaten außerdem innerhalb der kontrollierten Infrastruktur.

Ein kompatibler Endpunkt allein garantiert noch kein gleiches Verhalten. Ein neues Sprachmodell muss das Entscheidungsschema zuverlässig einhalten und mit den Dialogtests geprüft werden. Bei einem neuen Embedding-Modell muss die FAQ-Wissensbasis vollständig neu eingebettet und indiziert wer-

den. Für einen anderen Reranker sind Relevanzschwelle und Trefferzahl neu zu bestimmen. Bei Spracherkennung und Sprachausgabe müssen Audioformat, Deutschqualität, Streaming und Latenz erneut geprüft werden. Die Modellnamen sind daher eine erprobte Referenzkonfiguration, aber keine feste Produktabhängigkeit.

Technische Quellen zur Modellwahl (Abruf: 30.07.2026)

[1] Qwen3.6-35B-A3B-FP8 – Modellkarte: https://huggingface.co/Qwen/Qwen3.6-35B-A3B-FP8
[2] Qwen3-Embedding-0.6B – Modellkarte: https://huggingface.co/Qwen/Qwen3-Embedding-0.6B
[3] bge-reranker-v2-m3 – Modellkarte: https://huggingface.co/BAAI/bge-reranker-v2-m3
[4] Cohere Transcribe 03-2026 – Modellkarte: https://huggingface.co/CohereLabs/cohere-transcribe-03-2026
[5] Qwen3-TTS-12Hz-1.7B-Base – Modellkarte: https://huggingface.co/Qwen/Qwen3-TTS-12Hz-1.7B-Base
[6] vLLM – Inference- und Serving-Dokumentation: https://docs.vllm.ai/en/latest/
[7] vLLM – Modell- und API-Endpunkte: https://docs.vllm.ai/en/latest/models/pooling_models/
[8] vLLM-Omni – Sprachsynthese-API: https://docs.vllm.ai/projects/vllm-omni/en/latest/serving/speech_api/

2.2.4 Dialogsteuerung und Routing

Jeder Dialogschritt beginnt mit der Einordnung der neuen Eingabe. Sie kann ein neues Anliegen, eine Folgefrage, eine Korrektur oder ein erneuter Klärungsversuch sein. Danach wählt das System eine Route: allgemeines Wissen, konkrete Serviceauskunft, Authentifizierung, Rückfrage, nicht unterstützter Vorgang, themenfremde Anfrage oder Menschenwunsch.

Die Klassifikation wird in einer festen Struktur ausgegeben. Sie enthält die gewählte Route, eine Suchfrage oder Abfrageart, erkannte Kennungen und gegebenenfalls das fehlende Detail. Ungültige oder unvollständige Entscheidungen werden nicht direkt ausgeführt. Die Dialogsteuerung übersetzt sie in eine zulässige Aktion und aktualisiert danach den Gesprächszustand.

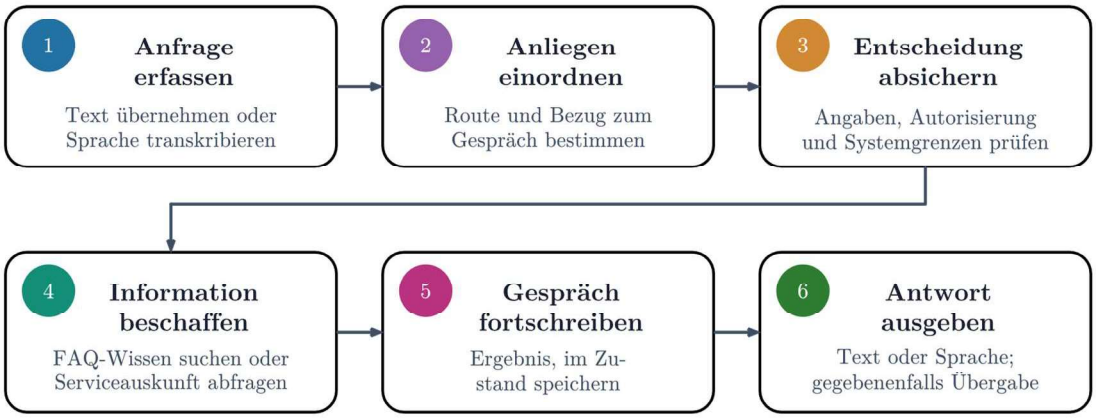


Abbildung 2: Ablauf eines Dialogschritts von der Eingabe bis zur Antwort.

Explizites Zustandsmodell des Gesprächs

Der CallState ist der fachliche, pro Sitzung fortgeschriebene Gesprächszustand. Er wird nicht frei vom Sprachmodell geschrieben. Das Modell liefert nur eine TurnDecision; erst die Zustandsmaschine erzeugt daraus den nächsten CallState und die wirksame Aktion. Davon getrennt hält die Conversation-Runtime rein betriebliche Daten wie Sperren, laufende Streaming-Generationen, Verbindungszähler und Audio-Puffer.

CallState-Bereich	Gespeicherte Bedeutung
phase	Aktiver Dialog, Warten auf Übergabebestätigung, Warten auf Authentifizierung oder terminal angeforderte Übergabe.
active_issue	Art, Status, Suchfrage, Fokus, Werkzeugaufruf und fehlendes Detail des aktuell bearbeiteten FAQ- oder Serviceanliegens.
pending_tool_request	Expliziter Wartezustand für eine fehlende Kennung, Authentifizierung oder Auswahl aus mehreren Ersatzteiltreffern.
confirmed_brand / confirmed_product	Nur ausdrücklich bestätigte Angaben werden über Folgeturns weiterverwendet; ein neues Anliegen löscht diese issue-bezogenen Slots.
current_device / current_part	Bereits aus SQL aufgelöste Geräte- und Artikelfakten für natürliche Folgefragen, ohne dieselbe Abfrage erneut auszuführen.
Klärungszähler	Drei ungelöste FAQ-Versuche beziehungsweise mehr als zwei SQL-Klärungen führen zum Angebot einer menschlichen Übergabe.
Unterbrechungsstatus	Kennzeichnet, dass eine TTS-Antwort hörbar unterbrochen wurde und im nächsten Turn nicht als vollständig gehört vorausgesetzt werden darf.
tool_auth_verified	Sitzungsbezogene Freigabe des PoC-Authentifizierungsgates für geschützte Reparatur- und Auftragsabfragen.

Deterministische Übergänge und Aktionsverträge

Ausgangssituation	Übergang	Ergebnis
Geschützte Serviceabfrage	active → awaiting_tool_auth	Der Werkzeugaufruf wird gespeichert, aber vor erfolgreicher Authentifizierung nicht ausgeführt.
Erfolgreiche Authentifizierung	awaiting_tool_auth → active	Die gespeicherte Abfrage wird mit ihrer ursprünglichen Suchfrage und Kennung fortgesetzt.
Expliziter Menschenwunsch	active → awaiting_handoff_confirmation	Die Übergabe wird zunächst nur angeboten; ein bloßer Modelltext kann sie nicht auslösen.
Bestätigte Übergabe	awaiting_handoff_confirmation → handoff_requested	Der Dialog wird terminal, TTS wird deaktiviert und eine interne Zusammenfassung erzeugt.
Neues Anliegen	Issue-bezogener Reset	Alte Slots, Zähler und aufgelöste SQL-Entitäten werden nicht in ein neues Thema übernommen.

Zusätzlich hängt die Zustandsmaschine an sicherheitskritische Aktionen einen strukturierten Aktionsvertrag. Dieser legt beispielsweise fest, dass nur nach dem Passwort gefragt werden darf, vor Authentifizierung keine privaten Daten ausgegeben werden und eine Übergabe erst nach einer kurzen Ja/Nein-Frage gestartet wird. Die Formulierung bleibt modellbasiert, die erlaubte Handlung jedoch deterministisch.
Für eine Übergabe wird zuerst die Zustimmung der anrufenden Person eingeholt. Nach der Bestätigung endet die automatisierte Bearbeitung und es wird eine sachliche Zusammenfassung aus dem bisherigen Dialog erzeugt.

2.2.5 Wissensbasierte Beantwortung

Allgemeine Fragen werden aus einer vorbereiteten FAQ-Sammlung beantwortet. Jedes FAQ-Dokument enthält Titel, Frage, Antwort und eine für die Suche erzeugte Zusammenfassung. Marken, Produkte, Fehlerbilder, Prozessschritte und wichtige Einschränkungen werden dabei ausdrücklich erhalten. Beim Aufbau der Wissensbasis werden diese Felder als Text sowie als Embedding in OpenSearch gespeichert. Textindex und Vektorindex beziehen sich damit auf denselben fachlich freigegebenen Inhalt. Für jede Anfrage bildet die Dialogsteuerung eine Suchfrage aus der aktuellen Äußerung und dem bestätigten Gesprächskontext. OpenSearch führt zwei Suchen aus: eine lexikalische Suche über gewichtete Textfelder und eine k-nächste-Nachbarn-Suche über alle Fragevektoren. Mittels Reziprocal Rank Fusion (RRF) werden beide Ranglisten in der Suchpipeline zusammengeführt. Danach bewertet der Cross-Encoder-Reranker die Suchfrage gemeinsam mit jedem Kandidatentext. Nur die besten FAQ-Einträge oberhalb der Relevanzgrenze werden an die Antworterzeugung weitergegeben.
Die Antworterzeugung darf nur Informationen aus den ausgewählten Inhalten und aus bestätigten Angaben des Gesprächs verwenden. Bedingungen, die nur für bestimmte Marken, Produkte oder Fälle gelten, werden nicht ohne entsprechende Zuordnung auf die aktuelle Anfrage übertragen.

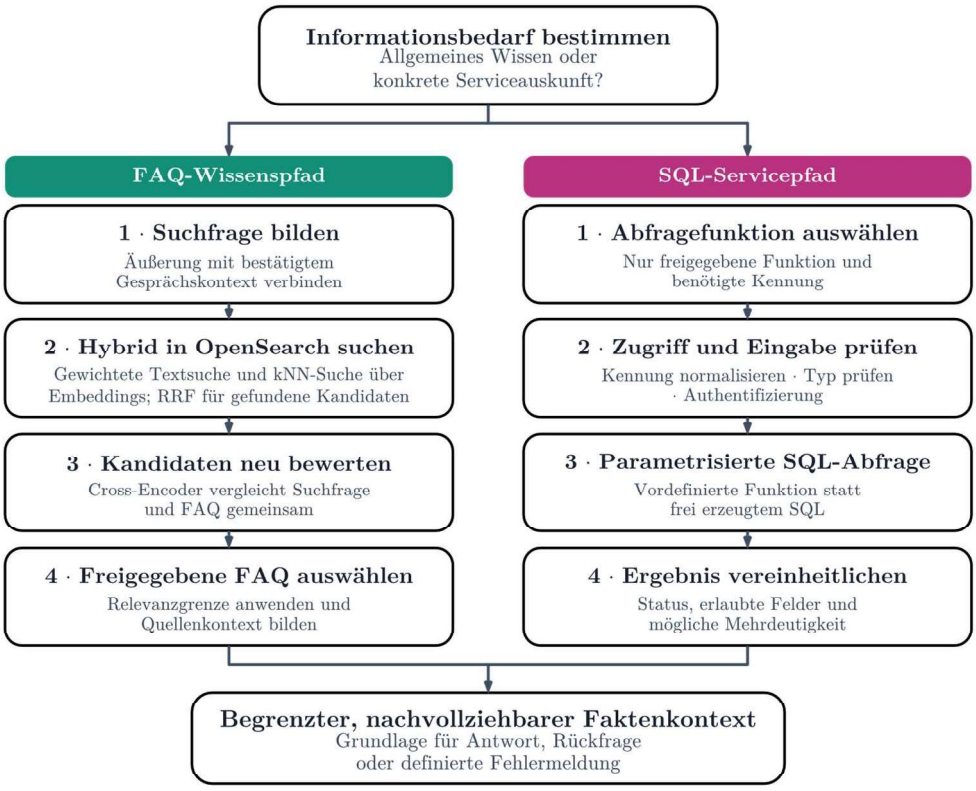


Abbildung 3: Getrennte Informationswege für FAQ-Wissen und strukturierte Serviceauskünfte.

2.2.6 SQL-gestützte Serviceauskünfte

Konkrete Serviceauskünfte kommen aus einer relationalen SQL-Datenbank. Das Sprachmodell erzeugt keine SQL-Anweisung. Es wählt eine freigegebene Abfrageart und liefert die erkannte Kennung in einem festen Entscheidungsformat. Eine Registry ordnet diese Auswahl einer implementierten Datenbankfunktion zu. Das Modell kann deshalb weder beliebige Tabellen ansprechen noch eigene Abfragebedingungen einschleusen. Dadurch bleibt technisch festgelegt, auf welche Daten das System zugreifen darf und welche Abfragen überhaupt ausgeführt werden können.

Auskunft	Benötigte Angabe	Authentifizierung
Ersatzteile und Zubehör	Artikel-, Typ- oder Modellnummer	Nein
Preis und Lieferstatus	Artikelnummer	Nein
Kompatibilität	Gerätekennung und Artikelnummer	Nein
Reparaturstatus	Reparaturnummer oder Kundenreferenz	Ja
Auftragsstatus	Auftrags- oder Internetbestellnummer	Ja

Vor der Ausführung werden Abfrageart, Anzahl und Typ der Parameter geprüft. Produkt-, Modell- und Vorgangskennungen werden in eine standardisierte Schreibweise überführt; die eigentliche Datenbankabfrage verwendet parametrisierte Werte. Fehlt eine notwendige Kennung, fragt das System gezielt danach und speichert den Zweck der offenen Abfrage. Bei einer Korrektur wird die neue Kennung dem bestehenden Anliegen zugeordnet. Für geschützte Informationen wird die vorgesehene Abfrage zurückgehalten, bis die Authentifizierung erfolgreich abgeschlossen ist.

Die Datenbankfunktion liefert nur die für den jeweiligen Zweck benötigten Felder. Das Ergebnis wird in einen einheitlichen Status und einen strukturierten Faktenkontext überführt: Ergebnis vorhanden, kein Eintrag gefunden, leeres Ergebnis oder technischer Fehler. Bei mehrdeutigen Ersatzteiltreffern speichert die Dialogsteuerung die Kandidaten und lässt die anrufende Person auswählen, ohne dieselbe Abfrage erneut auszuführen. Interne Felder und konkrete Lagerbestandsmengen werden nicht automatisch in die Kundenantwort übernommen. Erst dieser begrenzte Kontext wird an das Sprachmodell zur Formulierung weitergegeben. Die verwendeten Geschäftsdaten sind im Format and die Geschäftsdaten der Walzer Elektronik Vertriebsgesellschaft mbH angepasste, in diesem Projekt jedoch zu testzwecken vollständig synthetisch generiert.

Für eine Produktumsetzung sollte die SQL-Schicht als Adapter verstanden werden. Die hier verwendete relationale Datenquelle kann durch ein betriebliches Datenbanksystem oder eine abgesicherte Fach-API ersetzt werden, solange Abfragevertrag, Authentifizierungsregel, Ergebnisstatus und freigegebene Antwortfelder erhalten bleiben. Datenbankzugänge gehören ausschließlich in den Adapter; das Sprachmodell erhält weder Zugangsdaten noch direkten Datenbankzugriff.

2.2.7 Sprachschnittstelle

Die Spracheingabe wird fortlaufend auf Sprachaktivität untersucht. Erst wenn eine zusammenhängende Äußerung erkannt und abgeschlossen wurde, wird sie an die Spracherkennung übergeben. Sehr kurze oder überwiegend geräuschhafte Abschnitte werden verworfen. Das erzeugte Transkript durchläuft anschließend denselben Dialogablauf wie eine Texteingabe.

Speech-to-Text (STT) und Voice Activity Detection (VAD)

Für die Segmentierung wird Silero VAD als neuronales Sprachaktivitätsmodell eingesetzt. Eingehende Audiodaten werden auf Mono und 16 kHz vereinheitlicht und frameweise bewertet. Der VAD-Iterator meldet Sprachbeginn und -ende; ein rollender Vorpuffer erhält Laute unmittelbar vor dem erkannten Beginn. Erst eine abgeschlossene und durch zusätzliche Akustik- und Qualitätsgrenzen akzeptierte Äußerung wird als WAV an den Speech-to-Text-Dienst übertragen. So erhält das STT-Modell keine kontinuierliche Aufnahme, sondern nur plausible Äußerungssegmente.

VAD-/STT-Parameter	Funktion
Abtastrate und Framegröße	16 kHz, 512 Samples (32 ms). Einheitliche, fortlaufende Modellbewertung kurzer Audioblöcke.
Sprachschwelle	Ab dieser Modellwahrscheinlichkeit kann ein Frame als Sprache gewertet werden.
Ende einer Äußerung	Verhindert, dass kurze Sprechpausen unter 600 ms einen Satz vorzeitig teilen; dieses Fenster ist zugleich Teil der nutzersichtbaren Antwort-Latenz.
Vorpuffer	Nimmt frühe Silben und Konsonanten vor dem sicheren Startsignal in die Äußerung auf.
Dauergrenzen	Verwirft sehr kurze Ereignisse (unter 300ms) und begrenzt überlange Segmente (mehr als 30s).
Akustikgrenzen	Filtert zu leise Segmente, bevor kostenintensive STT-Inferenz startet.
VAD-Qualitätsgrenzen	Unterdrückt überwiegend geräuschhafte oder nur schwach als Sprache erkannte Segmente.

Text-to-Speech (TTS), Streaming und Barge-in

Text-to-Speech bezeichnet die Umwandlung der freigegebenen Textantwort in Sprache. Vor der Synthese normalisiert MARA deutsche Zahlen, Preise, Datums- und Uhrzeitangaben, Telefonnummern, URLs sowie alphanumerische Kennungen. Der TTS-Dienst liefert mono PCM mit 24 kHz; die Anwendung reicht bereits verfügbare 20-ms-Blöcke über WebSocket an den Browser weiter. Eine Antwort muss daher nicht vollständig synthetisiert sein, bevor die Wiedergabe beginnt.

Für Barge-in wird derselbe VAD-Pfad auch während laufender TTS-Ausgabe weitergeführt. Erst wenn die Audio- und VAD-Qualitätsgrenzen stabil erfüllt sind und zusätzlich 200 ms Entprellzeit verstrichen sind, wird die Unterbrechung übernommen. Generationsnummern entwerten daraufhin noch laufende Text- und Audioströme; der Browser leert seine Wiedergabewarteschlange und bestätigt, ob tatsächlich Audio unterbrochen wurde. Dieser Status wird im CallState gespeichert, damit der nächste Dialogschritt nicht voraussetzt, dass die vorherige Antwort vollständig gehört wurde.

2.2.8 Bedienoberfläche

Die Oberfläche dient als Ein- und Ausgabekanal für die vollständige Verarbeitungskette. Ein Gespräch kann gestartet und über ein Texteingabefeld oder das Mikrofon fortgeführt werden. Erkannte Sprache erscheint als Transkript im Gesprächsverlauf. Antworten werden zunächst als Text dargestellt und können zusätzlich vorgelesen werden. In diesem Fall generiert das TTS-Modell, wie beschrieben, Audio, das an das Frontend gestreamt wird.

Über Neustart lässt sich der Gesprächszustand zurücksetzen; der letzte Dialogschritt kann zurückgenommen werden. Eine Übergabe wird im Demonstrationsablauf durch die sichtbare Übergabe-Zusammenfassung dargestellt. Die Oberfläche ersetzt keine Telefonanlage, bildet aber Textdialog, Spracheingabe, Sprachausgabe und Zustandswechsel gemeinsam ab.

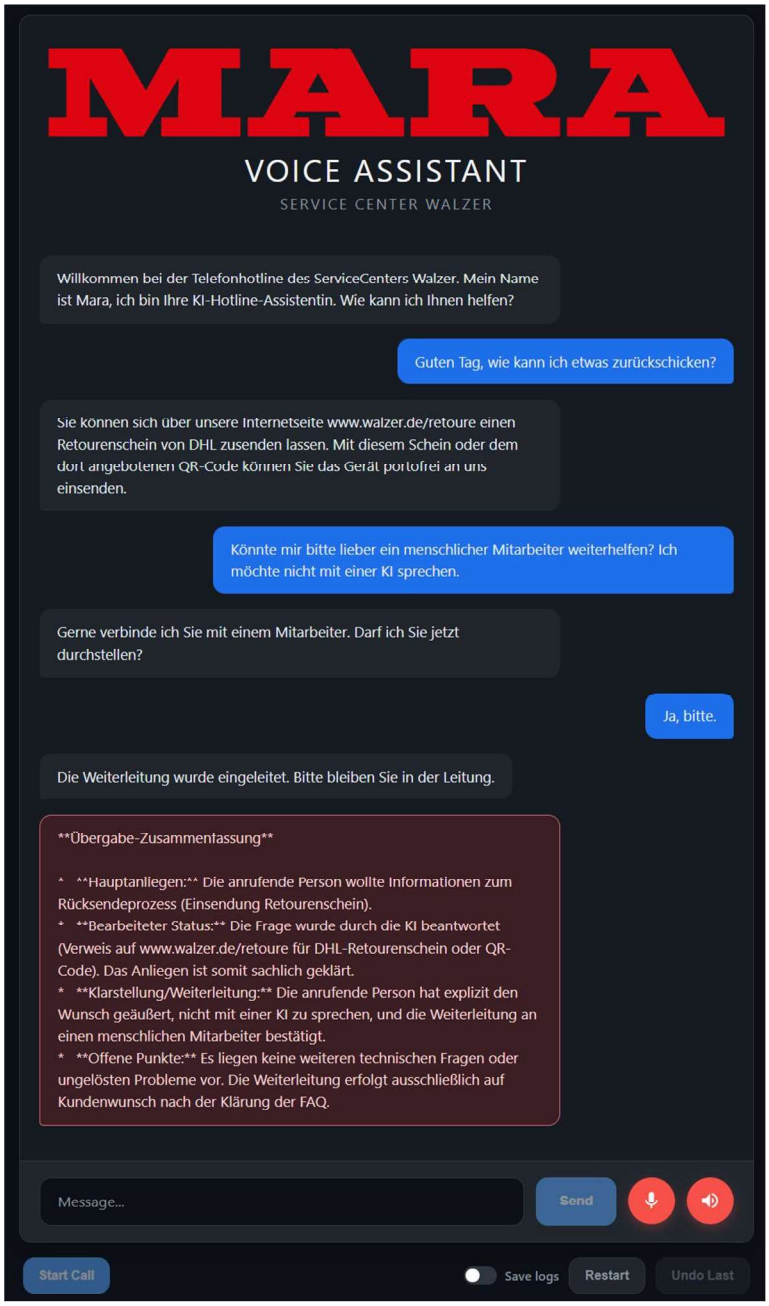


Abbildung 4: Browserbasierte Bedienoberfläche des MARA-Systems.

2.2.9 Qualitätssicherung und technische Grenzen

Die Evaluation trennt isolierte Komponentenmessungen von vollständigen Gesprächsszenarien. Dadurch lässt sich unterscheiden, ob ein Fehler aus Spracherkennung, Routing, Retrieval, Zustandsfortschreibung, Antwort oder Sprachausgabe stammt. Prozentwerte beziehen sich ausschließlich auf die jeweils dokumentierten Testdaten und sind keine allgemeine Leistungszusage.

Ergebnisse der Teil- und Systemtests

Testpfad	Umfang und Ergebnis	Technische Einordnung
STT und VAD	45 gesprochene Sätze von drei Personen; Sprechform-WER 14,75 %; kritische Token 77,08 %; alle kritischen Werte je Satz 66,67 %; 0 % Fehltranskriptionen in den zwei Negativfällen	Normale Sprache wird schnell verarbeitet, gesprochene Produkt- und Vorgangskennungen bleiben jedoch die wichtigste Fehlerquelle.
FAQ-Retrieval	701 Anfragen; FAQ-Routing 85,02 %; bei den als FAQ gerouteten Controller-Suchfragen liegt der erwartete Inhalt in 92,45 % der Fälle unter den ersten drei Produktionstreffern	Hybrid Retrieval und Reranking sind stark; Fehler entstehen zusätzlich durch Routing und Umschreiben der Suchfrage.
SQL-Servicepfad	1.727 Ein-Turn-Fragen; Retrieval-Passrate 96,41 %; exaktes Routing und Werkzeugparameter 95,77 %; Aktionspassrate 99,88 %	Freigegebene Abfragen sind zuverlässig, wenn die Kennungen im Transkript korrekt vorliegen.
Scope-Grenzen	33 Fälle; Gesamtpassrate 87,88 %; keine unerwartete SQL-Nutzung oder Kontextfreigabe; keine harten Leakage-Fehler	Die Abgrenzung schützt den Werkzeugpfad; vier unklare Anfragen wurden weich falsch geroutet.
TTS	17 Gold-Texte; alle Audiodateien technisch erzeugt und ohne TTS-Fehler; Verständlichkeitsrate 94,12 %; Median-Real-time-Factor 0,22	Die Ausgabe ist schneller als Echtzeit, verfehlte aber den gesetzten Grenzwert bis zum ersten Audioblock in allen 17 Einzelmessungen.
End-to-End	30 Szenarien mit drei Personen; subjektive Szenario-Passrate 83,33 %; Antworten 96,43 % hilfreich; TTS-, Kontextfortführungs- und Handoff-Bewertung jeweils 100 %	Die automatische Gesamtpassrate liegt bei 30,00 % und die Turn-Passrate bei 55,95 %; insbesondere Kennungserkennung, Parameter und Auth-Resume bleiben verbesserungsbedürftig.

Gemessene Latenzen der finalen Konfiguration

Messpunkt	Median
Reiner STT-Dienst	135 ms
Akustisches Sprechende bis fertiges Transkript einschließlich VAD-Stilleprüfung	745 ms
FAQ-Produktionsretrieval mit Controller-Suchfrage, ohne Controller und Antwortgenerierung	156 ms
Vollständiger SQL-Ein-Turn-Benchmark einschließlich Routing, Abfrage und Antwort	780 ms
TTS-Anfrage bis erster PCM-Audioblock	512 ms
Akustisches Sprechende bis zur ersten Audioausgabe im End-to-End-Test	1.754 ms

Die End-to-End-Latenz ist der maßgebliche Wert für den wahrgenommenen Sprachdialog. Sie enthält die 600-ms-Stilleprüfung des VAD, STT, Controllerentscheidung, gegebenenfalls Retrieval, Antwortbeginn, TTS, WebSocket-Übertragung und Browser-Scheduling. Die einzelnen Tabellenzeilen dürfen nicht addiert werden, weil sie aus unterschiedlichen Testläufen stammen und teilweise überlappende Verarbeitung enthalten. Die Messung endet beim Einplanen des ersten nicht stillen Audioanteils; zusätzliche physische Lautsprecherlatenz wurde nicht gemessen.

Grenzen des Entwickelten Systems

Bereich	Verbleibende Grenze
Telefonie	Der Web-Transport ist funktionsfähig; Provideranbindung, echte Rufweiterleitung und Rückmeldung eines Telefonieanbieters sind nicht implementiert.
Sitzungen und Skalierung	Gesprächszustände liegen im Arbeitsspeicher eines einzelnen Prozesses.
Authentifizierung	Das PoC nutzt ein sitzungsbezogenes Passwort-Gate. Web-Session-ID und Cookie sind nur Wiederverbindungshilfen, keine Identität. Für Produktion sind echte Identitätsprüfung, Autorisierung und Auditierung erforderlich.
Daten und Fachintegration	Die Geschäftsdaten sind synthetisch und die SQL-Funktionen auf das Walzer-Demoschema zugeschnitten. Ein Produktivsystem benötigt abgesicherte Fach-APIs und reale Datenqualitätsregeln.
Sprachrobustheit	Buchstabierte und lange alphanumerische Kennungen verursachen noch Fehler. Bestätigungsdialoge oder alternative sichere Übermittlung kritischer Nummern sind als Folgemaßnahme sinnvoll.
Betrieb und Sicherheit	Lokale HTTP-Endpunkte, fehlende Produktionsauthentifizierung, Quoten, zentrales Monitoring und durchgängige asynchrone Backend-Ausführung sind außerhalb des PoC-Umfangs.
Übertragbarkeit der Messwerte	Die Latenzen gelten für die dokumentierte lokale Modell- und Hardwarekonfiguration sowie die Testlast. Gleichzeitige Anrufe, Netzwerkbetrieb und Lastspitzen wurden nicht vermessen.

2.3 Verwertung

Das Ergebnis des Projekts ist mit MARA („Modularer Anruf- und Routingassistent“) ein modular aufgebauter Proof of Concept zur automatisierten Unterstützung der Kundenhotline der Walzer Elektronik Vertriebsgesellschaft mbH. Die entwickelte Architektur ist grundsätzlich auf weitere Unternehmen und Domänen übertragbar, in denen wiederkehrende Serviceanfragen bearbeitet werden und Informationen aus unterschiedlichen Wissens- und Bestandssystemen zusammengeführt werden müssen. Insbesondere die Kombination aus semantischer Wissenssuche, kontrollierten Datenabfragen, Authentifizierung, Dialogzustandsverwaltung und der Übergabe an menschliche Mitarbeitende kann auch in anderen technischen Service- und Supportprozessen eingesetzt werden.

Die Walzer Elektronik Vertriebsgesellschaft mbH kann die Projektergebnisse als Grundlage für die Weiterentwicklung eines digitalen Kundenservicekanals nutzen. Das Fraunhofer IMS kann die gewonnenen technischen und methodischen Erkenntnisse in zukünftige Transfer-, Forschungs- und Entwicklungsprojekte einbringen. Die wirtschaftliche Verwertung der Ergebnisse obliegt den Projektpartnern jeweils selbst.
Das Fraunhofer IMS wird über das Projekt unter Nennung der Walzer Elektronik Vertriebsgesellschaft mbH öffentlichkeitswirksam im Kontext von ZaKI.D berichten.
Für einen produktiven Einsatz unter realen Bedingungen sind weitere Entwicklungs- und Absicherungsschritte erforderlich. Dazu zählen insbesondere die Anbindung an die produktiven Fach-, Auftrags-, Reparatur- und Ersatzteilsysteme, die Pflege und Qualitätssicherung der Wissensbasis sowie die technische und organisatorische Absicherung von Authentifizierung, Berechtigungen und Datenschutz. Die im PoC nicht enthaltene reale Telefonieanbindung einschließlich der Weiterleitung über einen Telefonieanbieter müsste ergänzt werden.

3 Zusammenarbeit

Die Zusammenarbeit verlief flexibel und abgestimmt mit dem Projektpartner. Status-Updates seitens des Projektteams erfolgten anlassbezogen und zeitnah per E-Mail und Telefon. Rückmeldungen des Projektpartners zu Serviceabläufen, Kundenanliegen und relevanten Informationsbedarfen wurden iterativ in die Konzeption und Umsetzung des Prototyps einbezogen. Der geplante Zeitrahmen wurde dabei durchgehend eingehalten.

4 Einordnung im ZaKI.D-Kontext

Dieses Projekt war ein weiteres Umsetzungsprojekt des ZaKI.D-Projektes und das vierte durch das Fraunhofer IMS abgeschlossene Umsetzungsprojekt.
ZaKI.D hat unter anderem zum Ziel auch die Auswirkung auf die 17 Ziele zur nachhaltigen Entwicklung der Vereinten Nationen zu betrachten. Mit dem durchgeführten Projekt werden die folgenden Ziele unterstützt:

SDG 8 MENSCHENWÜRDIGE ARBEIT UND WIRTSCHAFTSWACHSTUM
Die Automatisierung wiederkehrender Serviceanfragen entlastet Mitarbeitende der Kundenhotline von Routinetätigkeiten. Dadurch bleiben mehr Kapazitäten für komplexe, wertschöpfende Aufgaben und eine menschenwürdige Arbeitsgestaltung, z. B. durch weniger repetitive Belastung.

SDG 9 INDUSTRIE, INNOVATIOn UND INFRASTRUKTUR
MARA demonstriert, wie moderne KI-Methoden (Sprachverarbeitung, semantische Suche, kontrollierte Datenabfragen) modular in bestehende Serviceinfrastrukturen integriert werden können. Die Architektur ist domänenübergreifend übertragbar und stärkt die digitale Innovationsfähigkeit von Unternehmen, insbesondere auch von KMU.

SDG 11: NACHHALTIGER KONSUM UND PRODUKTION
Durch präzise Auskünfte zu Ersatzteilen, Preisen und Kompatibilität werden durch MARA Fehlkäufe, unnötige Retouren und falsche Reparaturwege reduziert. Dies spart Material, Verpackung und Logistikressourcen – der erhöhte Strombedarf der KI-Infrastruktur wird durch die vermiedenen Material- und Prozesskosten idealerweise überkompensiert.

5 Glossar

Begriff / Abkürzung	Erläuterung
API	*Application Programming Interface*, Programmierschnittstelle. Legt fest, wie Softwarekomponenten Funktionen aufrufen und Daten miteinander austauschen.
Authentifizierung / Autorisierung	Authentifizierung prüft eine behauptete Identität. Autorisierung prüft dagegen, auf welche Daten eine Person oder Softwarekomponente zugreifen und welche Aktionen sie ausführen darf.
Barge-in	Unterbrechung einer laufenden Sprachausgabe durch eine neue Spracheingabe der nutzenden Person.
Embedding	Numerische Vektordarstellung eines Textes, mit der sich dessen inhaltliche Ähnlichkeit zu anderen Texten vergleichen lässt.
FAQ	*Frequently Asked Questions*, häufig gestellte Fragen. Im Projekt eine Wissensbasis aus vorbereiteten Fragen und Antworten.
FP8 / BF16	Gleitkomma-Zahlenformate mit 8 beziehungsweise 16 Bit. FP8 benötigt für entsprechend gespeicherte Modellwerte weniger Speicher, bietet jedoch eine geringere numerische Genauigkeit.
GPU	*Graphics Processing Unit*, Grafikprozessor. Wird zur beschleunigten Ausführung rechenintensiver KI-Modelle eingesetzt.
Hybride Suche	Kombination mehrerer Suchverfahren. MARA verbindet lexikalische Suche anhand vorkommender Wörter mit semantischer Suche anhand der inhaltlichen Bedeutung.
Inferenz	Anwendung eines bereits trainierten KI-Modells auf neue Eingaben, beispielsweise zur Transkription von Sprache oder zur Erzeugung einer Antwort.
Latenz	Zeitspanne zwischen zwei festgelegten Ereignissen, beispielsweise zwischen dem Ende einer Spracheingabe und dem Beginn der Antwortausgabe.
PoC	*Proof of Concept*, Machbarkeitsnachweis. Prototypische Umsetzung zur Untersuchung eines Lösungsansatzes; kein Nachweis der Produktivreife.
Realtime-Factor (RTF)	Verhältnis der Verarbeitungsdauer zur Audiodauer. Ein Wert unter 1 bedeutet eine Verarbeitung schneller als Echtzeit; er beschreibt nicht die Wartezeit bis zum ersten Audioblock.
Reranker / Reranking	Modell beziehungsweise Verfahren, das bereits gefundene Suchtreffer hinsichtlich ihrer Relevanz für die Suchfrage erneut bewertet und sortiert.
Retrieval	Gezieltes Auffinden und Abrufen relevanter Informationen aus einer Wissensbasis oder einer strukturierten Datenquelle.
Routing	Zuordnung einer Anfrage zu einem vorgesehenen Bearbeitungsweg, beispielsweise FAQ-Suche, Serviceauskunft oder Rückfrage.
RRF	*Reciprocal Rank Fusion*. Verfahren zur Zusammenführung mehrerer Suchtrefferlisten anhand der Rangpositionen ihrer Einträge.
Serving	Technische Bereitstellung eines KI-Modells als aufrufbarer Dienst einschließlich der Ausführung von Modellanfragen.

Begriff / Abkürzung	Erläuterung
SQL	*Structured Query Language*. Sprache zur Abfrage und Bearbeitung von Daten in relationalen Datenbanken. MARA verwendet vorab definierte, parametrisierte Abfragen statt frei erzeugter SQL-Anweisungen.
STT / ASR	*Speech-to-Text* beziehungsweise *Automatic Speech Recognition*. Automatische Erkennung gesprochener Sprache und deren Überführung in Text.
Streaming	Schrittweise Bereitstellung und Übertragung von Daten, bevor das vollständige Ergebnis vorliegt, beispielsweise bei Textantworten oder Sprachausgaben.
Systemprompt	Übergeordnete Anweisung an ein Sprachmodell, die dessen Aufgabe, Verhaltensregeln und zulässigen Informationskontext beschreibt.
Token	Verarbeitungseinheit eines Sprachmodells, beispielsweise ein Wortbestandteil oder Satzzeichen. In der Sprachauswertung bezeichnet der Begriff die jeweils bewerteten Textelemente.
TTS	*Text-to-Speech*. Automatische Umwandlung eines Textes in gesprochene Sprache.
VAD	*Voice Activity Detection*, Sprachaktivitätserkennung. Erkennt Sprachabschnitte und Sprechpausen in einem Audiosignal.
WER	*Word Error Rate*, Wortfehlerrate. Summe der ersetzten, ausgelassenen und zusätzlich eingefügten Wörter, geteilt durch die Wortanzahl des Referenztexts. Ein niedrigerer Wert bedeutet weniger Transkriptionsfehler.