JOHANNES KEPLER
UNIVERSITÄT LINZ

Praktikum Data & Knowledge Engineering, WS 2025

David Haunschmied
Institut für Wirtschaftsinformatik – Data & Knowledge Engineering

ERWARTETE VORKENNTNISSE

JOHANNES KEPLER
UNIVERSITÄT LINZ

Erwartete Vorkenntnisse lt. Studienhandbuch:

- **Vorlesung und Übung Data & Knowledge Engineering**

Darüber hinaus werden jene Studierende bevorzugt aufgenommen, die bereits folgende Lehrveranstaltung absolviert haben:

- **Software Engineering VL & UE**
- **Praktikum Software Engineering**

ZIELE (LT. STUDIENHANDBUCH)

JOHANNES KEPLER UNIVERSITÄT LINZ

Die Studierenden können im Team **praxisrelevante Aufgabenstellungen** des Data- und Knowledge Engineering lösen. Sie sind in der Lage, ein **ausgewähltes Werkzeug** zur Problemlösung einzusetzen, wie zum Beispiel ein objekt-orientiertes bzw. objektrelationales Datenbankverwaltungssystem. Sie sind in der Lage, theorie- und konzeptgeleitet Aufgaben **selbständig und eigenverantwortlich** zu bearbeiten, sowie bereit und **fähig, sich weitere Qualifikationen anzueignen, teamorientiert zu arbeiten, Gruppenprozesse zu moderieren und zu steuern,** Fachwissen in Bezug auf die Lernbedürfnisse aufzuarbeiten, zu reflektieren und zu vermitteln.

LEHRINHALTE

JOHANNES KEPLER
UNIVERSITÄT LINZ

Die Lehrinhalte (lt. Studienhandbuch) variieren entsprechend der jeweiligen Aufgabenstellung. Inkludiert sind jedenfalls der Einsatz und die Verwendung der für die Aufgabenstellung adäquaten Werkzeuge. Ein Beispiel ist die Entwicklung eines Dokumentenverwaltungssystems unter Verwendung eines objektrelationalen Datenbankverwaltungssystems mit XML-Unterstützung oder eines nativen XML-Datenbankverwaltungssystems. **Im Rahmen der Projektdurchführung werden der praktische Einsatz von Führungs-, Verhandlungs-, Konfliktlösungstechniken, Teamorganisation, Kommunikations- und Moderationstechniken sowie Techniken des selbstorganisierten Wissenserwerbs vermittelt.**

BEURTEILUNGSKRITERIEN UND LEHRMETHODEN

JOHANNES KEPLER UNIVERSITÄT LINZ

Zur Beurteilung werden **sämtliche Projektergebnisse** herangezogen, wie Zwischen- und Endpräsentationen, Dokumentation, und Systemimplementierung. Es werden **sowohl Team- als auch Einzelleistungen** beurteilt.

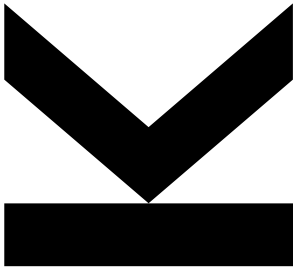
Sowohl die Aufgabenstellung als auch Lösungsidee werden vorgegeben. Die Studierenden arbeiten in Teams von 4 Personen. Jedes Team entwirft und implementiert unter Anleitung der/des LehrveranstaltungsleiterIn eine eigene Lösung. Die Studierenden präsentieren nach jeder abgeschlossenen Projektphase ihre Zwischenergebnisse und stellen diese im Plenum zur Diskussion.

ÜBER MICH

- Aus Schönau im Mühlkreis (Bez. Freistadt)
- HTL Perg für Informatik
- Studium der Wirtschaftsinformatik - JKU Linz
  - Obmann Stv. beim WIN Alumniclub
- Produkt Architekt bei Dynatrace

JOHANNES KEPLER
UNIVERSITÄT LINZ

RAG und LLM-Agenten



Von Retrieval-Augmented Generation
zu Agentic Systems

RETRIEVAL AUGMENTED GENERATION

JOHANNES KEPLER
UNIVERSITÄT LINZ

- LLMs (wie z.B. ChatGPT) werden initial anhand riesiger Datenmengen trainiert. Damit das LLM neueres Wissen lernt, muss ein erneutes Training erfolgen. Das ist teuer. Die Lösung dafür heißt RAG.

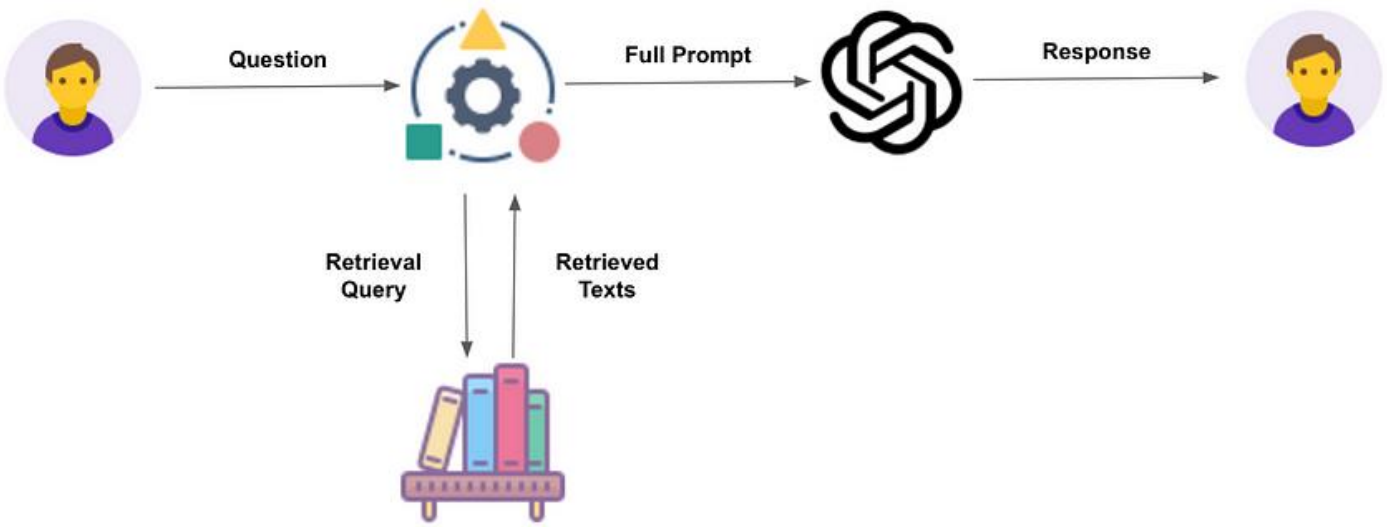
- RAG: Trainiertes LLM wird bei Abfragen mit aktuellen Daten / relevanten Kontext angereichert

- KI-Modell liefert **relevantere** und **aktuellere** Antworten

- Beispiel:
  Wie wird das Wetter am Schafberg in der kommenden Woche?
  - ☐ Prompt wird mit Wettervorhersagen am Schafberg in der kommenden Woche angereichert um die finale Antwort zu produzieren.

REFERENZARCHITEKTUR EINES RAG SYSTEMS

JOHANNES KEPLER UNIVERSITÄT LINZ



Quelle: https://pvml.com/glossary/retrieval-augmented-generation-rag/

WARUM AGENTEN?

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Früher: RAG als Pipeline („Retrieve → Generate“)

- Heute: Agenten, die **aktiv handeln** und **selbst entscheiden**, wann sie RAG oder andere Tools aufrufen
- Tools erlauben es dem Agenten mit der Umgebung zu interagieren
  - RAG, Websuche, Datenbankzugriffe, Rechner, Kalender-API, …

- Beispiel:
  Erstelle mir eine Packliste für eine Wanderung auf den Schafberg am kommenden Wochenende und ergänze falls nötig meine Einkaufsliste.
  - Tools: Wettervorhersage, Standortdaten, Präferenzen und Unverträglichkeiten des Benutzers, Einkaufsliste, (… und viele andere Tools die irrelevant für diese Frage sind)

AGENT REACT FRAMEWORK

JOHANNES KEPLER UNIVERSITÄT LINZ

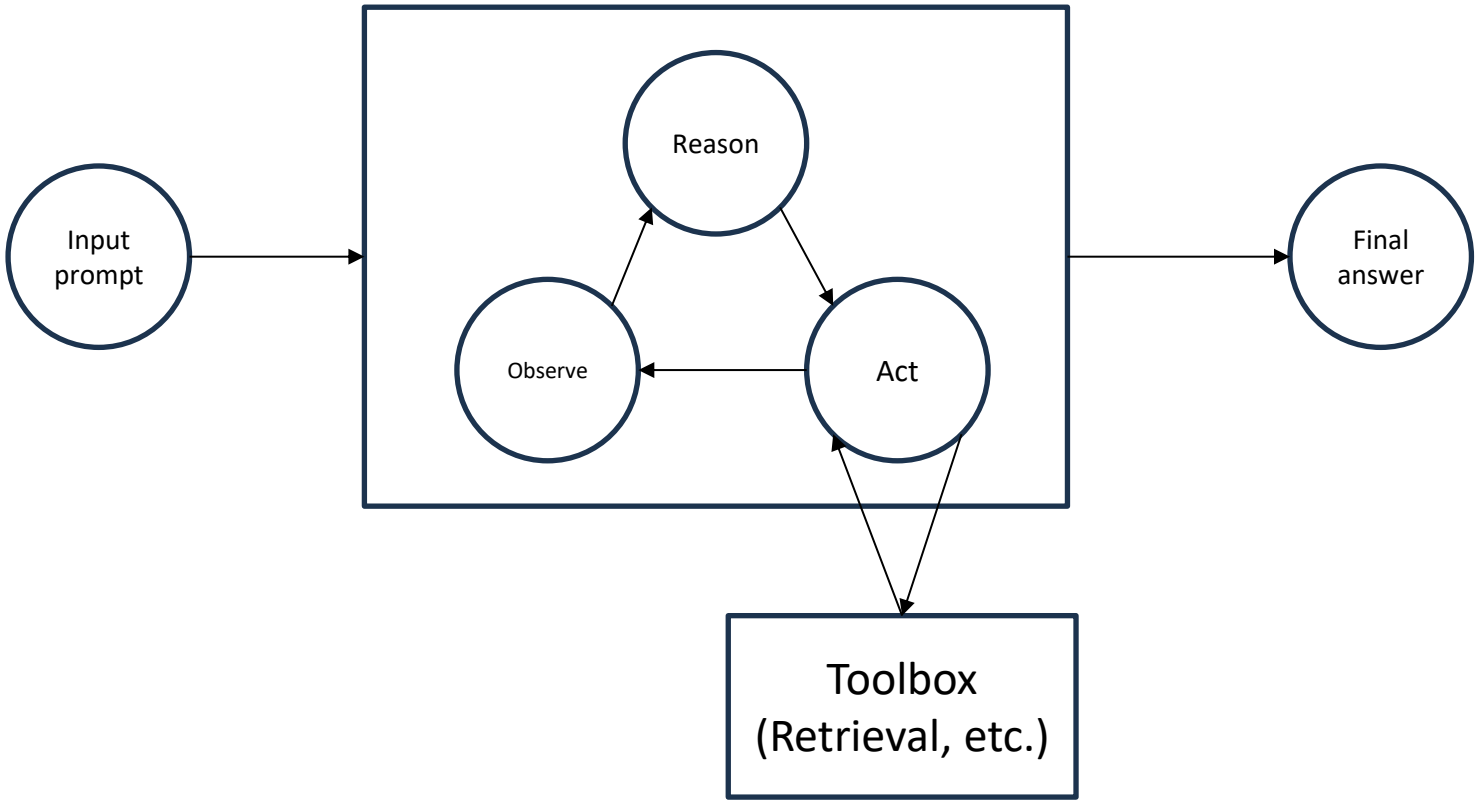
- Aktuelles Paradigma: *Reason* + *Act* (Yao et al., 2023) aka „React“

- Schritte in einer Schleife
  1. **Thought (Denkschritt):** „Was muss ich tun?“
  2. **Action (Aktion):** Toolaufruf (z. B. RAG, Web-Search, Calculator)
  3. **Observation (Beobachtung):** Tool-Ergebnis im Memory
  4. **Reflection (Reflexion):** Erkennt, ob Ziel erreicht → Schleife

- https://huggingface.co/docs/smolagents/conceptual_guides/react

REFERENZARCHITEKTUR EINES REACT-AGENTEN SYSTEMS

JOHANNES KEPLER UNIVERSITÄT LINZ



AUFGABENSTELLUNG

JOHANNES KEPLER
UNIVERSITÄT LINZ

- **Planung**, **Implementierung** und **Evaluierung**
  eines Retrieval Augmented Generation (RAG) Systems **oder**
  eines LLM-Agenten Systems

- Abhängig von Präferenz, Erfahrung und Anwendungsfall

KURSABLAUF

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Teambildung + Themenfindung
  - Vorschläge übernehmen oder selber eine Problemstellung überlegen
- Konzeptpräsentation
- Technischer Durchstich
- Zwischenpräsentation
- Endpräsentation & Einzelgespräche

BEURTEILUNG

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Projektnote / Teamleistung (50%)
  - Lauffähigkeit des Prototyps (**Minimalanforderung!**)
  - Funktionsumfang inkl. Evaluierung der Problemlösung
  - Qualität/Schwierigkeitsgrad der Implementierung

- Individuelle Performance (50%)
  - Beitrag zur Implementierung (**Minimalanforderung!**)
  - Einzelgespräch (**Minimalanforderung!**)
  - Präsentation(en)
  - Außerordentliche Leistungen (Teamleitung, hervorragende technische Lösungen, …)

TERMINPLAN

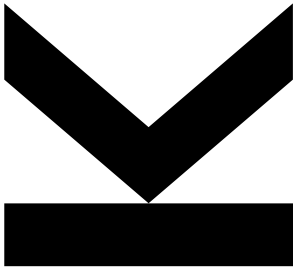
JOHANNES KEPLER
UNIVERSITÄT LINZ

Vorbesprechung, Aufnahme, Teambildung und Themenvorstellung	Di	07.10.2025	15:30 – 18:45, S3 134 (Anwesenheitspflicht)
Zwischenpräsentation – Konzeptpräsentation	Di	21.10.2025	15:30 – 18:45, S3 134 (Anwesenheitspflicht)
Zwischenpräsentation – Technischer Durchstich	Di	04.11.2025	15:30 – 18:45, S3 134 (Anwesenheitspflicht)
Gruppenbesprechung	Di	02.12.2025	vor Ort oder via Zoom
Zwischenpräsentation - Projektfortschritt	Di	16.12.2025	15:30 – 18:45, S3 134 (Anwesenheitspflicht)
Abschlusspräsentation	Di	20.01.2026	15:30 – 18:45, S3 134 (Anwesenheitspflicht)
Einzelgespräche	Di	26.01. - 30.01.2026	Termin nach individueller Vereinbarung

Für die restlichen Termine gilt: Gruppenbesprechung bei Bedarf (Anmeldung per E-Mail) vor Ort oder via Zoom

JOHANNES KEPLER
UNIVERSITÄT LINZ

Teambildung + Themenfindung



THEMENVORSCHLÄGE

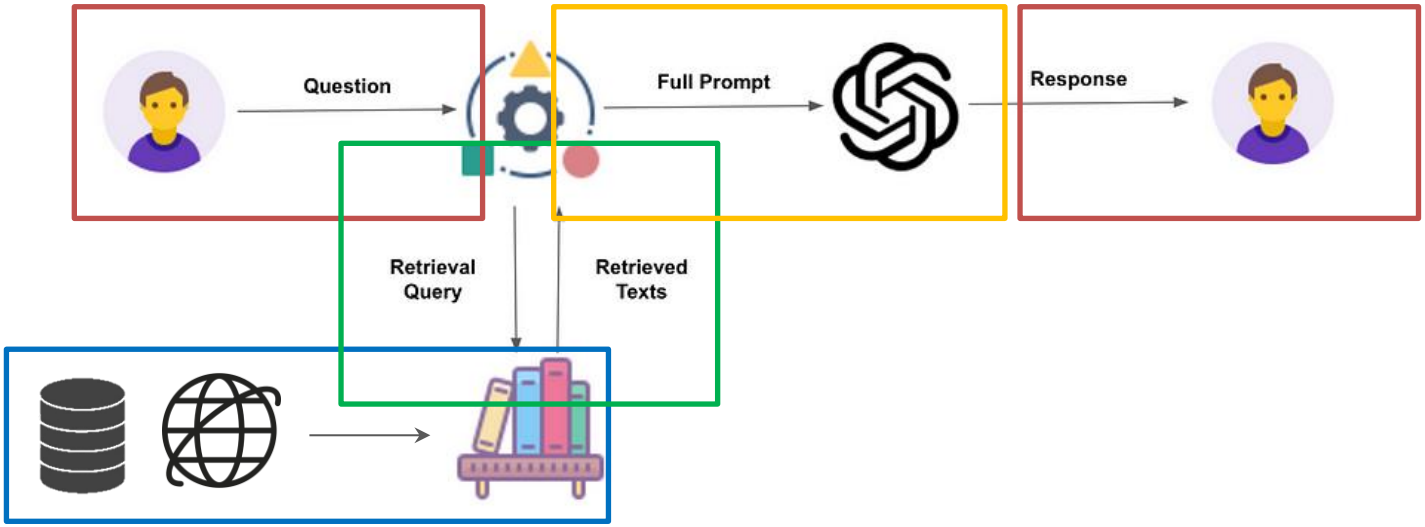
JOHANNES KEPLER
UNIVERSITÄT LINZ

- **News Bot (RAG)**
  - ☐ Beantwortet Fragen zu aktuellen Geschehnissen in meiner Umgebung stellen.
  - ☐ Kontext: Nachrichtenseiten, Suchmaschinen, …
- **JKU Study Assistant (RAG)**
  - ☐ Beantwortet Fragen zur JKU, Studienrichtungen oder Kursen beantworten.
  - ☐ Kontext: JKU-Webseite, Studienhandbuch, Curricula, …
- **Kitchen Assistant (Agent)**
  - ☐ Aufgabe: Plant Mahlzeiten & Einkäufe.
  - ☐ Tools: RAG (Rezepte-Retrieval), Inventar-DB, Nährwert-/Wochenplan-Tool, …
- **Personal Secretary (Agent)**
  - ☐ Aufgabe: Verwaltet Meetings, Aufgaben & Notizen.
  - ☐ Tools: RAG (Notiz-Retrieval), Kalender API, To-Do Datenbank

AUFTEILUNG IM TEAM (RAG)

JOHANNES KEPLER UNIVERSITÄT LINZ

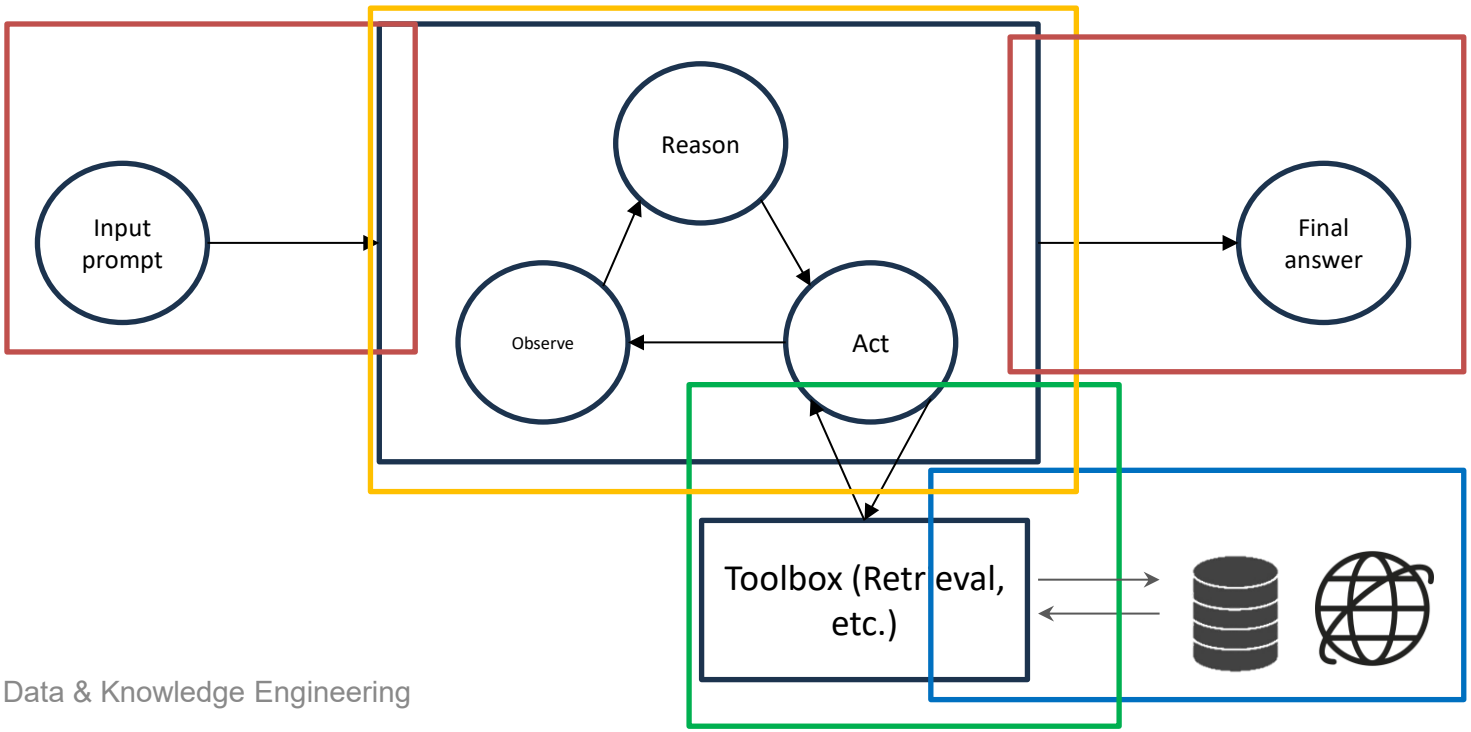
1. **User Interface** (z.B. Für Webinterface mit Chateingabe)
2. **ETL Komponente** (z.B. Studienhandbuch parsen und für Retrieval aufbereiten)
3. **Retrieval Komponente** (z.B. Relevante Nachrichten anhand der Benutzerfrage finden)
4. **LLM Komponente** (z.B. Aktuellen Chat merken; Fragen + relevante Nachrichten als Prompt an das LLM senden und Chatverlauf erweitern)



AUFTEILUNG IM TEAM (AGENT)

JOHANNES KEPLER UNIVERSITÄT LINZ

1. **User Interface** (z.B. Webinterface mit Agent-Chat + Dashboard)
2. **ETL Komponente** (z.B. Rezepte abrufen und für Tool bereitstellen)
3. **Toolbox Komponente** (z.B. Funktionen für RAG und CRUD Operationen bereitstellen)
4. **LLM Komponente** (ReAct Prozess abbilden)



UI + WEBSERVER

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Webinterface
- Usereingabe durch „Prompt“ (zb Chatfenster)
- Mindestens eine Art von User-Kontext (gilt nur für RAG)
  - Präferenzen
  - Demographische Merkmale
  - Zeit, Ort
  - …
- Feedback durch Benutzer

ETL KOMPONENTE

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Datenbeschaffung
- Bereinigung
- Abspeicherung in Datenbank
- Metadaten und Vektor-Embeddings für semantische Suche
- Embedding für große Texte
  - Mit LLM zusammenfassen lassen
  - In Chunks aufteilen und Durchschnittsvektor berechnen
  - Mehrfach speichern mit Vektor pro Chunk

RETRIEVAL KOMPONENTE (RAG)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Relevante Dokumente finden
- Basis ist der User Prompt und Kontext
- Retrieval Pipeline
  - Maximale Distanz
  - Maximale Anzahl an zurückgegeben Dokumenten
  - Erweiterte Suche (Ähnliche Dokumente gefundender Dokumente)
  - Reihung der Dokumente (zB. Nach Neuigkeit bei Nachrichten)

LLM KOMPONENTE (RAG)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Aus Prompt, Kontext, Feedback und Dokumenten eine präzise, relevante und benutzerfreundliche Antwort erstellen

- Kontextmanagement (Chatverlauf merken und erweitern)

- Optimierungen
  - ☐ (System) Prompt Engineering
  - ☐ Modell-Parameter ändern
  - ☐ Ausgabeformat (zB JSON)

TOOLBOX KOMPONENTE (AGENT)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Tools definieren und dem Agent zur Verfügung stellen
- Komplexität ist abhängig von den gewählten Tools (mindestens 3)
  - ☐ Datenbank (CRUD)
  - ☐ Web API
  - ☐ Semantisches Retrieval (RAG)
  - ☐ Lokales Filesystem
  - ☐ Kalender API
  - ☐ …
- Name und Beschreibung optimieren

LLM-KOMPONENTE (AGENT)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Den „Agent-Kern“ also den ReAct Algorithmus implementieren

- Kontextmanagement (Memory)

- Optimierungen
  - ☐ (System) Prompt Engineering
  - ☐ Modell-Parameter ändern
  - ☐ Ausgabeformat (zB JSON)

EVALUIERUNG

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Eingaben und erwartete Ausgaben definieren
- Manuell mit tatsächlichen Ausgaben vergleichen
- Mindestens 15 Fragen (RAG) oder 15 Aufgaben (Agenten)
  - ☐ Retrieval / Toolbox Komponenten separat evaluieren
  - ☐ Agenten: Iterationen evaluieren
- Optional: Automatisierte Evaluierung

TEAMBILDUNG

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Jeweils 4 Studierende pro Team

- Pro Team muss ein:e Teamleiter:in festgelegt werden
  - Projektkoordination
  - Hauptansprechpartner:in für den Kursleiter

- Jede:r muss einen Teil implementieren
  - Gemeinsames GIT Repository
  - Kontrolle durch den Kursleiter anhand der Commits

- Abgaben via Moodle
  - Konzeptdokument / Präsentation
  - Evaluierungsdokumente

JOHANNES KEPLER
UNIVERSITÄT LINZ

Konzeptpräsentation

KONZEPT (1)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- **Problem (WHY)**
  - Welches Problem wollen wir lösen? (Problemstellung)
  - Wie wollen wir es lösen? Warum ist RAG bzw. ein Agent dafür gut geeignet?
  - **Tipp**: Ein kleines Problem ganz zu lösen ist besser als ein großes Problem halb zu lösen

- **Funktionsumfang (WHAT)**
  - Welche Funktionen gibt es?
  - Wie interagiert der Benutzer mit dem System? (UI)
  - Minimalziel für jede Komponente (Must-haves, Nice-to-haves)
  - UI-Mockups, Beispieleingaben/-ausgaben

KONZEPT (2)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- **Systemarchitektur & Technologieauswahl (HOW)**
  - Wie entwickeln wir die jeweiligen Teile der Referenzarchitektur? Wie arbeiten/kommunizieren die Teile miteinander? Welche Schnittstellen gibt es? (Client/Server Verbindungen, Interfaces im Backend, Logisches Datenbank Design, …)
  - Beispiel: Python für das Backend, Datenbankzugriffe & ETL Prozess, Angular Frontend, PostgreSQL Datenbank mit Vektor-Erweiterung

- **Evaluierungsmethodik**
  - Welche Antworten erwarten wir uns auf welche Eingaben? Wie können wir einzelne Teile evaluieren? Manuell vs. automatische Evaluierung.
  - Beispiele: Manuel basierend auf der Qualität der Antworten bzw. der Trefferquote relevanter Daten der Retrieval Komponente; GPT-Similarity,
  - Minimalanforderung: 15 Fragen / Aufgaben und Antworten manuell evaluieren

NÜTZLICHE LINKS

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Getting started (RAG)
  - PostgreSQL as a Vector Database: A pgvector Tutorial
  - The Beginner's Guide to Building a RAG System with Qdrant
  - LangChain RAG tutorial
- Getting started with AI agents
- Beispiele: https://github.com/davidHaunschmied/dke-pr-examples
- Free LLM APIs: https://github.com/cheahjs/free-llm-api-resources
  - Recommended: Google Gemini
  - OpenAI (Kostet wenig, ist aber nicht gratis!)
- Datenbanken als Service
  - Simple Vector DB: https://qdrant.tech/
  - Relational DB: https://supabase.com/
  - NoSQL DB: https://www.mongodb.com/

ANSPRECHPARTNER

JOHANNES KEPLER
UNIVERSITÄT LINZ

- **David Haunschmied**
  E-Mail: david.haunschmied@hotmail.com, Spr.std.: nach Vereinbarung, Raum S3 103, Sekretariat: Raum S3 116

JOHANNES KEPLER
UNIVERSITÄT LINZ

Beispiel: NewsBot

BEISPIEL: NEWSBOT (RAG)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Problemstellung(en)
  - Nachrichtenflut führt zu gemindertem Interesse
  - Fehlender Kontext bei einem einzelnen Artikel
  - Ich will mehr über ein Thema wissen oder habe Fragen

- Lösungsansätze
  - Zusammenfassung aktueller Nachrichten
  - Ähnliche Artikel anzeigen
  - Fragen zu aktuellen Nachrichten in natürlicher Sprache

BEISPIEL: NEWSBOT

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Warum RAG?
  - Generative KI-Modelle verstehen Fragen und können Antworten in verschiedenen Varianten (z.B. Zusammenfassungen) liefern, kennen aber keine aktuellen Nachrichten
  - Der Originaltextservice bietet einen API an, um tagesaktuelle österreichische Nachrichten abzufragen
  - Kombiniert können beliebige Zusammenfassungen der Nachrichten generiert und Fragen dazu beantwortet werden

RAG-ARCHITEKTUR

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Retrieval Komponente
  - User Abfrage in Vektorrepräsentation umwandeln und in Datenbank nach Nachrichten mit ähnlichen Vektoren suchen. Aktuellere Artikel werden dabei als relevanter eingestuft.
  - Im Hintergrund werden periodisch aktuelle Nachrichten von https://www.ots.at/ abgefragt, Vektorrepräsentationen generiert und in der Datenbank gespeichert.
- Generative Komponente
  - User Abfrage wird mit 5 relevantesten Nachrichten angereichert und ergibt gemeinsam den Prompt an das LLM
- User-Interface
  - Web Interface mit Freitext-Eingabefeld und Bereich für die Antwort. Zudem werden Links zu den relevanten Nachrichten bereitgestellt, um die Originaltexte nachlesen zu können.

EVALUIERUNG

JOHANNES KEPLER UNIVERSITÄT LINZ

Requirement	Test Case	Test Title	Test Steps (Prompt)	Test Data (Retrieved Recipes)	Expected Result	Actual Result	Status	Notes to Status
						Suppe zum Kochen bringen und regelmäßig		
Frontend/ Gesamtsystem	3.3.2	Mehrbenutzermodus/Generativ	Nutzer 1: Bitte erstelle ein neues Rezept mit Hühnerfleisch Nutzer 2: Bitte erstelle ein neues vegetarisches Rezept Nutzer 3: Bitte erstelle ein neues süßes Rezept	**Nutzer 1:** wb-37285: Afrikanisches Huhn wb-37285: Afrikanisches Huhn wb-68176: Hühnersuppe festlich **Nutzer 2:** wb-38772: Linseneintopf wb-108459: Nasi Goreng **Nutzer 3:** wb-93869: Mispel-Parfait wb-36015: Bräntis wb-346: Mousse au chocolat	Drei verschiedene generierte Rezepte, welche keinerlei Ähnlichkeit oder Gemeinsamkeiten aufweisen	**Nutzer 1:** ID: N/A Name: Kokos-Curry-Hühnchen Zubereitungszeit: 30 Minuten Zutaten: 500 g Hühnerbrustfilets 200 ml Kokosmilch 1 EL rote Currypaste 1 EL Pflanzenöl 1 Zwiebel 2 Knoblauchzehen 1 rote Paprika 100 g Erbsen 1 TL Zucker 2 EL Sojasauce 1 Limette Frische Koriander blätter Zubereitungsschritte: Hühnerbrustfilets in Streifen schneiden. Zwiebel und Knoblauch schälen und fein hacken. Paprika in Streifen schneiden. Pflanzenöl in einer Pfanne erhitzen und die Zwiebel sowie den Knoblauch darin anbraten. Hühnerstreifen hinzufügen und anbraten, bis sie goldbraun sind. Rote Currypaste einrühren und kurz mitbraten. Kokosmilch, Zucker und Sojasauce hinzufügen und gut umrühren. Paprika und Erbsen in die Pfanne geben und	Passed	

EVALUIERUNG (RETRIEVAL)

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Eingabe: 10 – 15 definierte Fragen
  - „Wie hoch ist die Inflation aktuell?“
  - „Welches Wirtschaftswachstum können wir heuer erwarten?“
  - „Wie erfolgreich war Österreich bei den letzten olympischen Spielen?“
  - „Warum ist Klaus Luger nicht mehr im Amt?“

- Ausgabe: Top 5 relevante Nachrichten

- Evaluierung
  - Manuelle Evaluierung der Relevanz ausgewählten Nachrichten
  - Semi-automatisch: Zuweisung von Fragen zu Nachrichten und anschließender Abgleich ob diese zurückgegeben wurden.

EVALUIERUNG DES GESAMTSYSTEMS

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Eingabe: 10 – 15 definierte Fragen
  - „Wie hoch ist die Inflation aktuell?“
  - „Welches Wirtschaftswachstum können wir heuer erwarten?“
  - „Wie erfolgreich war Österreich bei den letzten olympischen Spielen?“
  - „Warum ist Klaus Luger nicht mehr im Amt?“

- Ausgabe: Antwort in natürlicher Sprache (Text)

- Evaluierung:
  - Manuelle Evaluierung nach Grad der Nützlichkeit der Antwort und der Relevanz der verlinkten Artikel (z.B. 1-5, gut/schlecht)

JOHANNES KEPLER
UNIVERSITÄT LINZ

Technischer Durchstich

TECHNISCHER DURCHSTICH: GENERELL

JOHANNES KEPLER
UNIVERSITÄT LINZ

- **Technische Machbarkeit prüfen**: Herausfinden, ob bestimmte Technologien, Frameworks oder Ansätze für das Projekt geeignet sind.
- **Risiken identifizieren**: Technische oder konzeptionelle Risiken frühzeitig erkennen
- **Unklarheiten beseitigen**: Offene Fragen hinsichtlich Implementierungsmöglichkeiten oder Integration klären
- **Entscheidungsgrundlage schaffen:** Der Durchstich liefert konkrete Ergebnisse, die als Grundlage für weitere Architektur- und Entwicklungsentscheidungen dienen
- **Ziel: Die Tauglichkeit eures Konzeptes nachzuweisen**

TECHNISCHER DURCHSTICH: KONKRET

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Fokus auf die prototypische Implementierung kritischer Komponenten. Beispiele:
  - ☐ Können wir Website XY parsen so wie wir uns das vorgestellt haben?
  - ☐ Ist Llama 3.1. lokal schnell genug lauffähig für unser Projekt?
  - ☐ Ist das LLM in der Lage die erwarteten Antworten zu generieren?

- **Das Ergebnis soll eine belastbare Grundlage für die weitere Implementierung sein.** Beispiele:
  - ☐ Verwendete Technologien stehen fest
  - ☐ Schnittstellen zwischen den Komponenten sind vereinbart und validiert
  - ☐ Bekannte Unsicherheiten sind beseitigt (zb. Externe Datenquelle, LLM API, …)

TECHNISCHER DURCHSTICH: BIS ZUM NÄCHSTEN TERMIN

JOHANNES KEPLER UNIVERSITÄT LINZ

- GIT Repository mit Zugriff für alle Teamitglieder + david.haunschmied@hotmail.com
  - GitHub: davidHaunschmied
  - Falls kein GIT Repository verwendet wird, bitte rechtzeitig bescheid geben!

- Präsentation eures technischen Durchstichs
  - Demo eures Prototyps / eurer Prototypen
  - Eventuelle Rechercheergebnisse

JOHANNES KEPLER
UNIVERSITÄT LINZ

Zwischenpräsentation

ZWISCHENPRÄSENTATION: BIS ZUM NÄCHSTEN TERMIN

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Aktueller Stand eures Produkts
  - ☐ Demos der Komponenten erwünscht
  - ☐ Code

- Im besten Fall sind alle Must-Haves umgesetzt und das System lauffähig, dann spart ihr euch einen stressigen Semesterabschluss

- Verbleibende Tätigkeiten bis zur Endpräsentation

JOHANNES KEPLER
UNIVERSITÄT LINZ

Endpräsentation

ENDPRÄSENTATION

JOHANNES KEPLER
UNIVERSITÄT LINZ

- Finale Demo aller Ende-zu-Ende Funktionalitäten
  - Live-Demo
  - Code
- Abgrenzung was nicht umgesetzt wurde
  - Must-haves (Erklärung warum)
  - Nice-to-haves
- Evaluierungsergebnisse