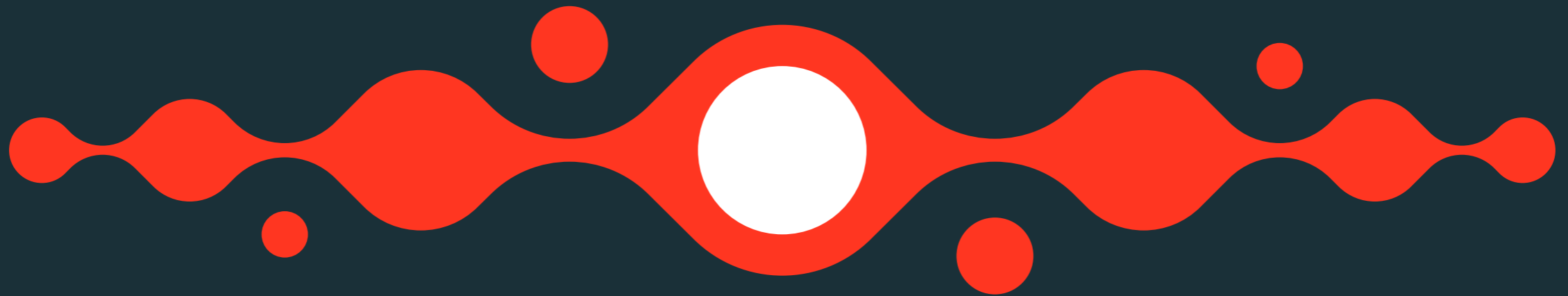


Kompaktleitfaden

# GenAI



# INHALT



- Einführung .....3**
- Kapitel 1: Grundlagen der generativen KI .....5**
  - Was ist generative KI?.....5
  - Grundzüge von Large Language Models ..... 7
  - Open-Source oder proprietäre Modelle: LLM-Benchmarking und -Auswahl .....9
  - Entwicklung und Anpassung von LLMs für konkrete Anwendungsfälle ..... 11
  - Wirtschaftlichkeit und langfristige Strategie ..... 14
- Kapitel 2: Die Kunst des Prompt-Engineerings..... 15**
  - Was ist Prompt-Engineering? ..... 15
  - Die Rolle des Prompt-Engineerings beim Steigern der Modellausgabequalität..... 16
  - Prompt-Engineering im Kontext der Konzipierung von KI-Agents ..... 18
- Kapitel 3: Entwicklung von KI-Agent-Systemen ..... 19**
  - Was ist ein KI-Agent?..... 19
  - Was ist ein KI-Agent-System? .....20
  - Kernphasen der Entwicklung eines KI-Agent-Systems .....20
  - Zentrale Komponenten eines KI-Agent-Systems .....21
  - Anwendungsfälle für KI-Agent-Systeme..... 23
- Kapitel 4: Fusion von LLMs und Wissensdatenbanken .....26**
  - Nutzen von RAG für den Zugriff auf externe Informationen ..... 27
  - Unstrukturierte Daten und Vektordatenbanken .....29
  - RAG-Qualität ..... 31
  - Fortlaufende Optimierung von RAG-Systemen.....35
- Kapitel 5: GenAI-Leistung und -Evaluierung.....36**
  - Mosaic AI Agent Evaluation.....36
  - Qualitätsmessung bei der Evaluierung einer KI-Anwendung ..... 41
  - LLMs als Prüfer .....43
  - Tracing für LLM-Transparenz ..... 47
- Kapitel 6: Governance für GenAI ..... 49**
  - Einheitliche Modellbereitstellung, Governance und Überwachung.....49
  - Mosaic AI Gateway: Eine umfassende Lösung für die GenAI-Governance .....50
  - Hauptmerkmale von Mosaic AI Gateway ..... 52
  - Umfassende Leitlinien für den sicheren KI-Einsatz ..... 53
- Ressourcen .....55**
- Fazit 57**
- Über Databricks .....58**

## Einführung



### **GenAI in Produktionsqualität erfordert neue Tools und Skills**

Generative KI hat Unternehmen neue Möglichkeiten erschlossen und grundlegend verändert, wie sie arbeiten, Produkte entwickeln und mit Kunden interagieren. Laut dem aktuellen Bericht „**Unlocking Enterprise AI**“ nutzen bereits 85 % der Unternehmen weltweit GenAI und bis 2027 sollen es 99 % sein. Allerdings tun sich noch viele Unternehmen schwer damit, diese Projekte erfolgreich zu skalieren. Der Bericht stellt auch fest, dass nur 22 % der Unternehmen der Überzeugung sind, ihre Infrastruktur sei KI-tauglich – und lediglich 37 % halten ihre GenAI-Modelle für wirklich produktionsreif. Viele Unternehmen mussten feststellen, wie schwierig es ist, solche Anwendungen in Produktionsqualität umzusetzen. Damit sich KI-Output für kundennahe Anwendungen eignet, muss er präzise, reguliert und sicher sein.

### **Mosaic AI: Vereinheitlichung des KI-Lebenszyklus von den Daten zur Bereitstellung**

Mosaic AI bietet eine umfassende Tool-Sammlung, mit der sich die Herausforderungen bei der Implementierung produktionsreifer GenAI-Anwendungen bewältigen lassen. Mit Mosaic AI können Unternehmen den gesamten KI-Lebenszyklus nahtlos verwalten – von der Datenerfassung und -aufbereitung über die Modellentwicklung bis hin zu Bereitstellung, Monitoring und Governance. Stand Juni 2024 hatten Mosaic AI-Kunden innerhalb eines Jahres über 200.000 maßgeschneiderte KI-Modelle entwickelt – auf der gleichen Infrastruktur und Technologie, die auch die hochmodernen Modelle von Databricks antreibt.

### **Die Dateninfrastruktur muss fit für GenAI-Anwendungen werden**

Der Sprung zur generativen KI geht deutlich über die Einführung simpler Chatbots hinaus: Er erfordert einen Umbau grundlegender Datenverwaltungspraktiken. Im Kern dieser Transformation steht das Data Lakehouses als moderner Datenstack, der Unternehmen schnellere und kostengünstigere Lösungen für die Datenverwaltung eröffnet. Dank dieser modernen Datenarchitekturen können Unternehmen das volle Potenzial von GenAI ausschöpfen, indem sie Daten auf einem einheitlichen System integrieren, skalieren und steuern.

Je stärker Unternehmen auf GenAI-Anwendungen setzen, um sich Wettbewerbsvorteile zu sichern, desto wichtiger wird eine zugrunde liegende Dateninfrastruktur, die diese modernen Technologien sicher und skalierbar unterstützt.

## **Ganz gleich, wo auf Ihrem Weg zur Implementierung von GenAI-Anwendungen Sie stehen: Es zählt die Qualität der Daten.**

Um mit GenAI Ergebnisse in Produktionsqualität zu erzielen, benötigen Unternehmen robuste Tools, mit denen sie die Qualität ihrer Daten und die von den Modellen generierten Resultate beurteilen können. Das umfasst das Zusammenführen und Optimieren aller Aspekte des GenAI-Prozesses: Datenaufbereitung, Modelltraining (unter Verwendung von Retrieval-Modellen, Sprachmodellen, Ranking- und Nachbearbeitungspipelines), Prompt-Engineering und das Anpassen von Modellen mithilfe von Unternehmensdaten.

Mosaic AI bietet eine einheitliche Plattform, auf der Data Scientists, Engineers und Fachbereiche gemeinsam daran arbeiten können, die Herkunft ihrer Daten und Modelle von den Rohdaten bis zur produktiven Bereitstellung nachvollziehbar zu machen. Durch die Integration mit Unity Catalog, Lakehouse Monitoring und Model Serving können Unternehmen jede Phase ihres GenAI-Workflows gezielt steuern und optimieren.

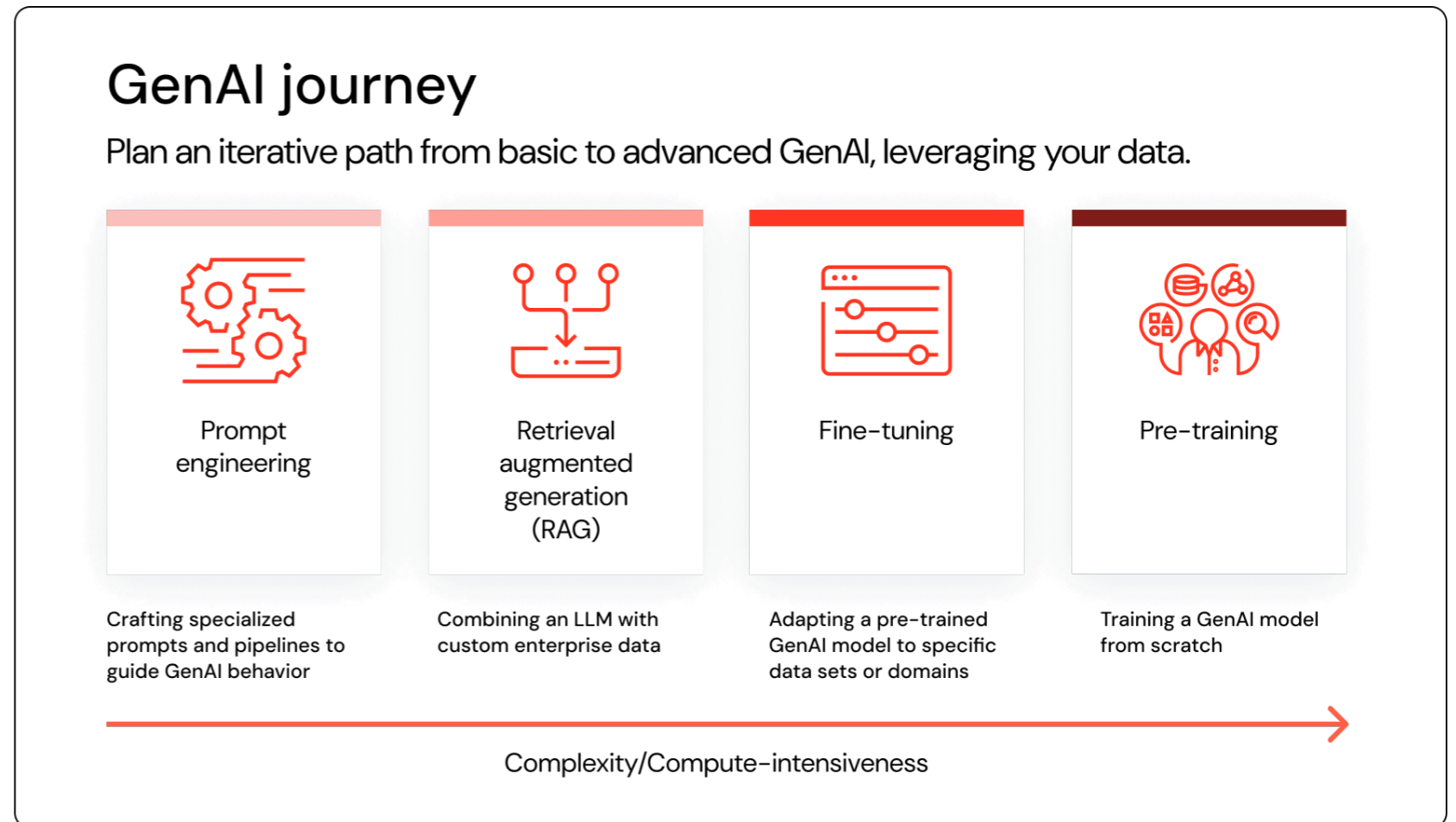
### **In diesem Leitfaden erfahren Sie Folgendes:**

- Wie man wirkungsvolle Prompts erstellt und fortschrittliche Verfahren für die Verwendung von Large Language Models (LLMs) in GenAI-Anwendungen einsetzt
- Grundlagen zum Aufbau aufgabenorientierter KI-Agent-Systeme und zur Integration von Retrieval Augmented Generation (RAG) für intelligentere Ergebnisse
- Wie die Performance von GenAI-Anwendungen bewertet und verbessert werden kann, inklusive Finetuning und Pretraining von Modellen
- Wie man maßgeschneiderte LLMs mit einer einheitlichen Plattform produktiv einsetzt und überwacht
- Best Practices für die Skalierung und Optimierung Ihrer GenAI-Anwendungen im Hinblick auf Performance, Qualität und Kosten

### **Behandelte Anwendungsfälle für GenAI:**

- Wie man LLMs nutzt, um aus Produktbewertungen umsetzbare Erkenntnisse zu gewinnen
- Wie sich die Qualität von Chatbot-Ausgaben mit RAG verbessern lässt
- Wie man eigene GenAI-Modelle kostengünstig trainiert
- Wie man implementierte GenAI-Modelle und ihre Performance überwacht und evaluiert

## Kapitel 1: Grundlagen der generativen KI



### Was ist generative KI?

Generative KI ist ein Bereich der künstlichen Intelligenz, der sich der Erstellung von Modellen widmet, die neue Inhalte generieren können – z. B. Texte, Bilder, Code oder synthetische Daten. Im Gegensatz zu klassischen KI-Modellen, die auf festen Eingabe-Ausgabe-Mustern basieren, analysieren oder Vorhersagen treffen, zielt generative KI darauf ab, Ergebnisse zu erzeugen, die bestehenden Inhalten ähneln oder diese erweitern. Der entscheidende Unterschied: Generative KI-Modelle können auf Basis erlernter Muster aus großen Datenmengen völlig neue, einzigartige Kombinationen erzeugen.

Diese Modelle lassen sich in der Regel in zwei Kategorien einteilen: Large Language Models (LLMs) und Foundation Models (Basismodelle). LLMs sind eine Unterklasse generativer Modelle, die auf Sprachaufgaben spezialisiert sind. Foundation Models hingegen sind umfangreiche, vortrainierte Modelle, die als Grundlage für die weitere Finetuning oder anwendungsspezifische Anpassung dienen. Nach erfolgtem Training generieren diese Modelle Ausgaben, die die Struktur, den Tonfall und die Intention natürlicher Sprache nachahmen – etwa zur Textgenerierung, Übersetzung, Zusammenfassung oder Beantwortung von Fragen.

Generative KI nutzt außerdem fortschrittliche Verfahren wie **Prompt-Engineering** und **Retrieval Augmented Generation (RAG)**. Prompt Engineering bezeichnet das gezielte Formulieren von Prompts, mit denen bei einem Modell das gewünschte Verhalten hervorgerufen werden soll. RAG hingegen kombiniert ein LLM mit Funktionen zum Abrufen externer Informationen. So kann das Modell die generierten Inhalte um relevante und aktuelle Details ergänzen.

Die Flexibilität und Leistungsfähigkeit generativer KI zeigen sich in einer Vielzahl von Anwendungsbereichen, unter anderem in:

- **Dokumentenerstellung:** Automatisches Erstellen von Berichten, Verträgen oder Angeboten, die auf spezifische Anforderungen zugeschnitten sind, wodurch die Produktivität gesteigert und der manuelle Aufwand verringert wird
- **Bildgenerierung:** Erzeugung neuer Bilder auf Grundlage einer Eingabe oder einer Stiltransformation
- **Sprachverarbeitung:** Etwa Transkription, Übersetzung und das Analysieren von Intentionen mittels Natural Language Processing
- **Agents:** Autonome Systeme, die komplexe Aufgaben durch Interaktion mit Benutzern oder anderen Systemen ausführen können, häufig auf Basis generativer Modelle
- **Finetuning:** Anpassung vortrainierter Modelle für spezifischere, fachbezogene Aufgaben durch Anpassung mit passgenauen Datensätzen

Obwohl viele GenAI-Modelle über integrierte Sicherheitsvorkehrungen verfügen, können sie dennoch unrichtige oder sogar schädliche Ergebnisse ausgeben. Daher ist bei ihrer Verwendung Vorsicht geboten.

## Grundzüge von Large Language Models

Large Language Models sind eine Variante der Deep-Learning-Modelle, die sich besonders für sprachbezogene Aufgaben wie das Verstehen und Generieren natürlicher Sprache eignen. Diese Modelle basieren auf riesigen Datenmengen, die häufig Texte aus vielen unterschiedlichen Quellen umfassen, darunter Bücher, Websites und Social-Media-Plattformen. Beim Training dieser Modelle werden riesige Textmengen eingespeist, damit sie Muster und Zusammenhänge erkennen – und so vorhersagen können, welches Wort oder welche Wortgruppe in einem Satz als Nächstes folgt.

Wenn ein Modell trainiert wurde, kann es mithilfe der erlernten Muster logisch aufgebaute, im jeweiligen Kontext relevante Sätze generieren. Diese Modelle basieren auf statistischen Wahrscheinlichkeiten, die aus den Trainingsdaten abgeleitet werden. Daher können sie menschliche Texte mit erstaunlicher Präzision nachahmen. LLMs können verschiedene sprachbezogene Aufgaben ausführen, etwa Textgenerierung, Sprachübersetzung und Zusammenfassung.

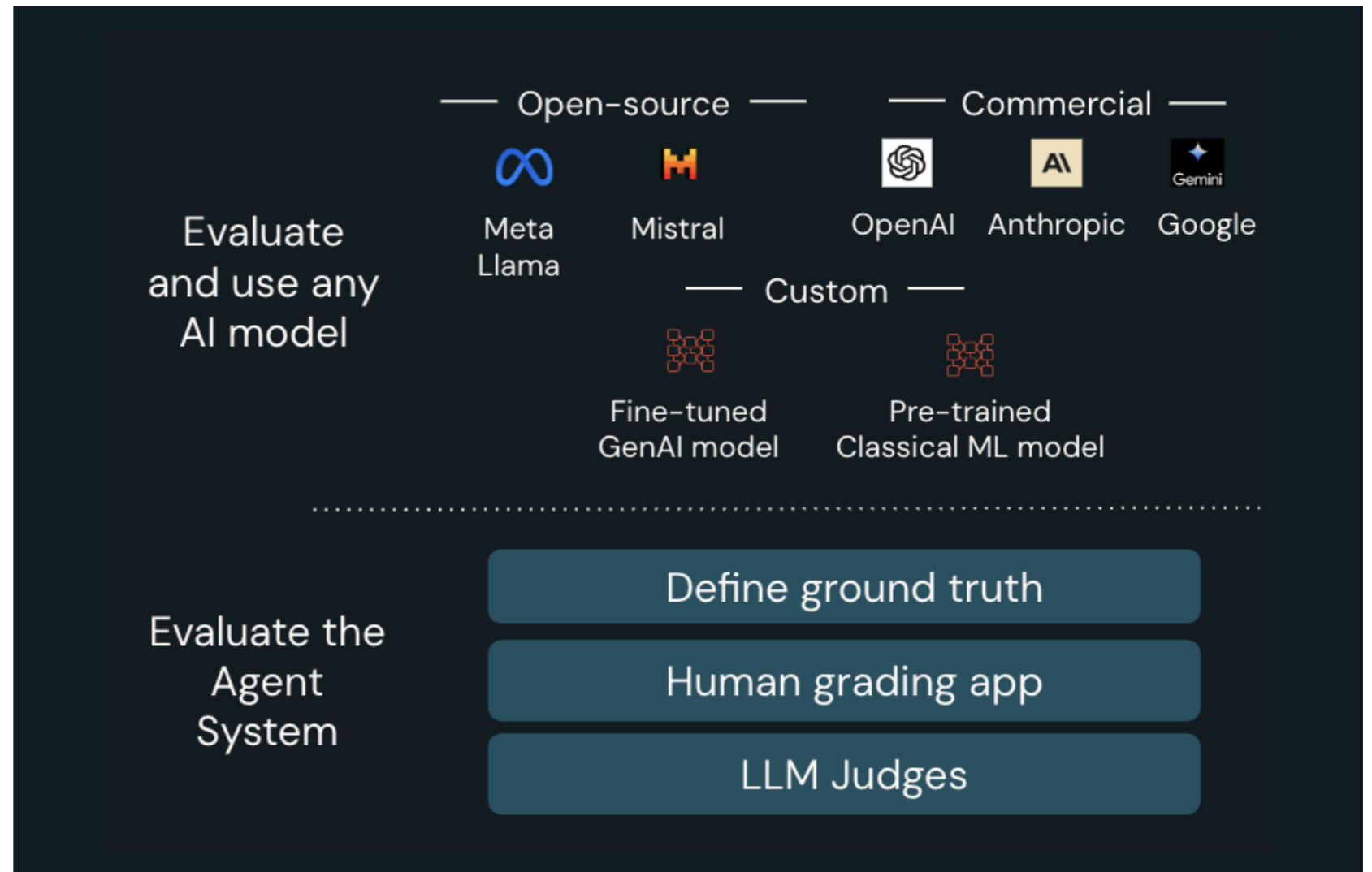
Training und Anpassung:

- 1. Pretraining:** Das Pretraining von LLMs beginnt mit dem Einlesen umfangreicher Textdaten, um so eine grundlegende Sprachkompetenz aufzubauen. In dieser Phase erlernt das Modell einfache Sprachstrukturen und Wortassoziationen. Das Pretraining erfolgt häufig mithilfe von Techniken des unbeaufsichtigten Lernens, bei denen das Modell fehlende Wörter in Sätzen voraussagt oder lernt, Sprachmuster von Grund auf neu zu modellieren.
- 2. Finetuning:** Nach dem Pretraining erfolgt normalerweise ein Finetuning des LLM mit fachspezifischen Datensätzen. In diesem Schritt wird das universelle Modell an die Anforderungen bestimmter Branchen oder Anwendungen angepasst, beispielsweise das Gesundheits- oder Finanzwesen oder Kundensupport. Beim Finetuning werden die Modellparameter angepasst, um die Performance bei bestimmten Aufgaben oder Datensätzen zu optimieren. Hierzu werden häufig kleinere, kuratierte Datensätze genutzt.

LLMs erhalten eine zusätzliche Dimension, wenn sie mit **RAG (Retrieval Augmented Generation)** kombiniert werden. Sie greifen während der Textgenerierung auf relevante Informationen aus externen Datenquellen oder Wissensdatenbanken zu und erweitern so ihre Fähigkeiten. So bleibt das Modell thematisch aktuell und liefert dank der Verbindung von generativer Leistungsfähigkeit und Faktengenauigkeit präzisere Ergebnisse.

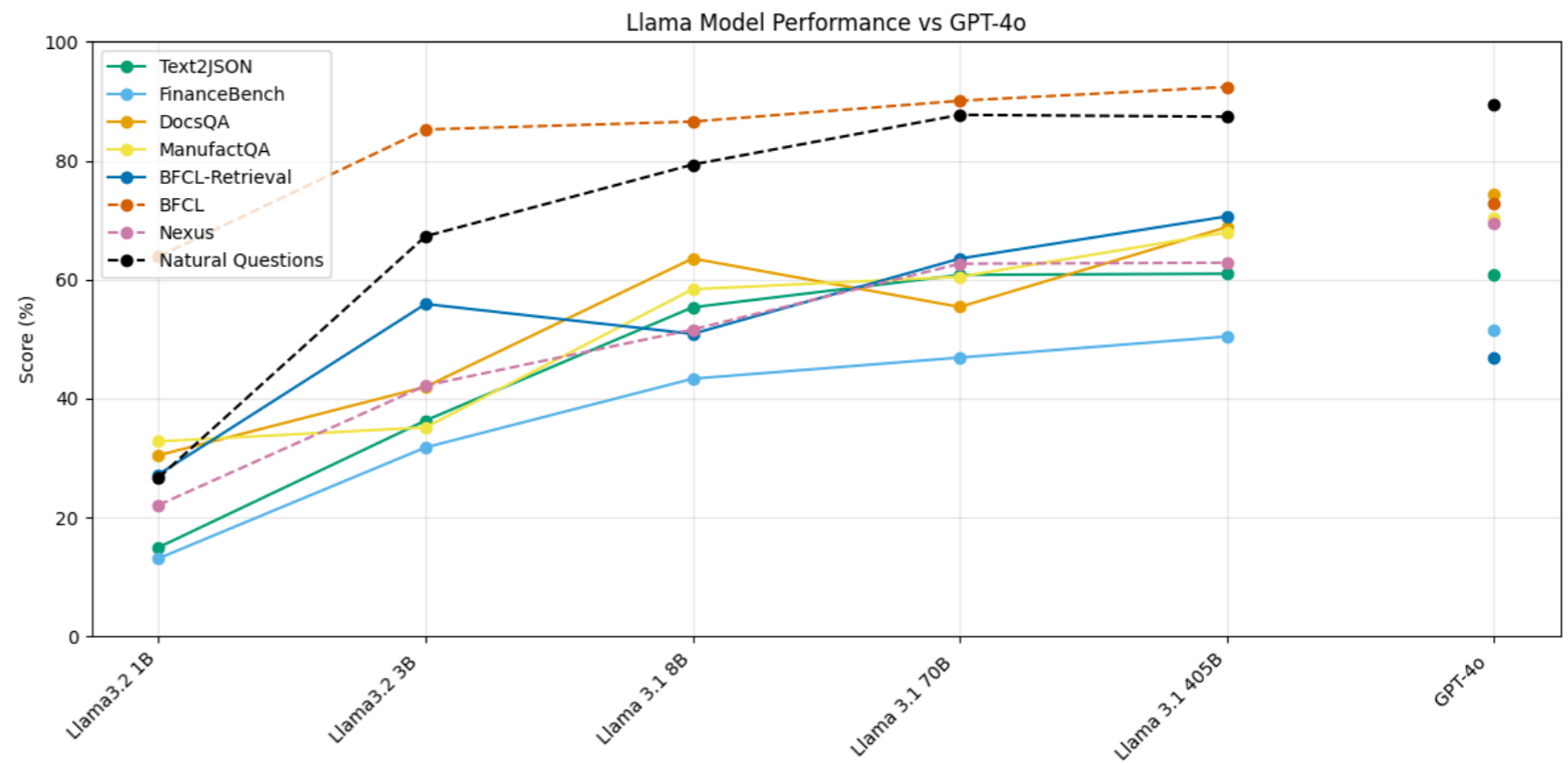
Darüber hinaus sorgt der Einsatz von **Agents** LLMs eine weitere Funktionsebene. Agents sind Systeme, die autonom mit Benutzern interagieren und Entscheidungen treffen. LLM-basierte Agents können komplexe Aufgaben übernehmen, wie beispielsweise das Ausführen von Abfragen, das Steuern von Workflows oder die Beteiligung an Dialogen. Durch Finetuning können sie weiter optimiert werden, um Aufgaben noch effizienter auszuführen.

Durch Verwendung dieser Modelle als Bausteine kann generative KI für eine Vielzahl von Anwendungsfällen angepasst werden – von Chatbots und virtuellen Assistenten bis hin zu stärker spezialisierten Anwendungen, die synthetisiertes Wissen aus verschiedenen Bereichen erfordern.



### Open-Source oder proprietäre Modelle: LLM-Benchmarking und -Auswahl

Die Auswahl des richtigen LLM für Ihre Anwendung ist kritisch, da sie sich direkt auf Leistung, Kosten und Skalierbarkeit auswirkt. Oft beinhaltet sie eine Entscheidung zwischen Open-Source- und proprietären Modellen mit jeweils eigenen Vor- und Nachteilen. Zu den wichtigsten Faktoren gehören die Performance, die Ausrichtung auf den Anwendungsfall, die Kosten und die Sicherheit.



**Abbildung 1:** Llama 3.1 405B zeigt bei vielen Aufgaben eine ähnliche Leistung wie GPT-4o. Wir beobachten, dass die Leistungsqualität mit abnehmender Modellgröße geringfügig abnimmt, auch wenn Llama 3.2 3B bei einigen der von uns bewerteten akademischen Benchmarks überdurchschnittlich gut abschneidet.

## LEISTUNGS-BENCHMARKS

Das Benchmarking unterstützt die Evaluierung von Modellen anhand von Kennzahlen wie Korrektheit, Latenz, Durchsatz und Kosteneffizienz. Die folgenden Metriken sollten sich an den Anforderungen Ihrer Anwendung orientieren:

- **Echtzeitanwendungen** legen den Schwerpunkt auf niedrige Latenzzeiten.
- **Umgebungen mit hohem Transaktionsaufkommen** erfordern Skalierbarkeit und Durchsatz.
- **Regulierte Branchen** benötigen Modelle, die strenge Sicherheits- und Compliance-Standards einhalten.

Proprietäre Modelle bieten häufig integrierte Sicherheits- und Compliance-Funktionen, während die Verwaltung dieser Aspekte bei Open-Source-Modellen in der Regel den Unternehmen obliegt, was Fachwissen und Ressourcen erfordert.

## OPEN-SOURCE-MODELLE

Open-Source-Modelle wie Llama von Meta sind kostengünstig und hochgradig anpassungsfähig. Dank ihrer Transparenz und Flexibilität eignen sie sich ideal für Organisationen, die die Kontrolle über die Architektur und das Verhalten nicht abgeben wollen oder dürfen (das gilt auch und gerade für regulierte Branchen). Solche Modelle werden von aktiven Communitys unterstützt, die Innovation und Aktualität vorantreiben.

Allerdings mangelt es ihnen häufig an aufgabenspezifischen Optimierungsoptionen, und sie erfordern einen erheblichen Wartungsaufwand bei Einsatz in der Produktion. Sicherheit und Skalierbarkeit sind weitere Herausforderungen, denen sich Unternehmen bei der Einführung von Open-Source-Modellen stellen müssen.

## PROPRIETÄRE MODELLE

Proprietäre Modelle von Anbietern wie OpenAI und Google sind häufig für bestimmte Aufgaben optimiert und bieten von Haus aus fortschrittliche Funktionen. Diese Modelle umfassen Support auf Unternehmensniveau und eine vollständige Dokumentation und sind konform zu Standards wie HIPAA oder SOC 2. Daher eignen sie sich für Unternehmen, die strengen regulatorischen Anforderungen unterliegen.

Der Nachteil sind die Kosten: Lizenz- und Nutzungsgebühren können beträchtlich sein. Außerdem haben die Unternehmen nur begrenzte Kontrolle über diese Modelle, was die Anpassung beschränkt und Datenschutzbedenken aufwerfen kann.

## Entwicklung und Anpassung von LLMs für konkrete Anwendungsfälle

Die Entwicklung und Anpassung von Large Language Models umfasst in der Regel drei Hauptphasen: **Pretraining**, **Finetuning** und **Continued Pretraining (CPT)**. Jede dieser Phasen erfüllt eine spezifische Funktion bei der Entwicklung von Modellen, die sowohl vielseitig einsetzbar als auch auf konkrete Anwendungsfälle oder Fachbereiche anpassbar sind. Gemeinsam bilden sie ein robustes Framework für die Entwicklung leistungsstarker, fachspezifischer KI-Lösungen.

### PRETRAINING

Das Pretraining ist der grundlegende Prozess, bei dem ein Modell mit riesigen Datensätzen konfrontiert wird, die Milliarden von Token aus verschiedenen Quellen wie Büchern, Artikeln und Websites enthalten. In dieser Phase erlernt das Modell allgemeine Sprachmuster, Grammatik und Fakten und wird so in die Lage versetzt, stimmige Texte zu generieren und eine Vielzahl sprachbezogener Aufgaben zu erledigen. Mit dem Universalwissen aus dem Pretraining können Modelle in verschiedenen Anwendungen hervorragende Ergebnisse erzielen, ohne für jede Aufgabe speziell trainiert werden zu müssen.

Allerdings erhalten Modelle durch das Pretraining zwar ein umfassendes Sprachverständnis, aber kein fachbezogenes Wissen. Daher können vortrainierte Modelle ohne weitere Anpassung Probleme mit Nischenaufgaben haben, beispielsweise mit der Interpretation von Rechtsdokumenten oder der Analyse klinischer Daten. Zudem ist das Pretraining ausgesprochen ressourcenintensiv und erfordert beträchtliche Rechenleistung und große Datensätze. Daher wird diese Phase meist vorrangig von Organisationen mit entsprechend umfangreichen Ressourcen durchgeführt.

## FINETUNING

Das Finetuning setzt auf dem vortrainierten Modell auf und trainiert dieses mithilfe kleinerer, auf die jeweilige Aufgabe zugeschnittener Datensätze. In dieser Phase können Unternehmen Modelle für konkrete Anwendungen anpassen, beispielsweise für die Kundendienstautomatisierung, die Inhaltszusammenfassung oder medizinische Diagnosen. Das Finetuning ist deutlich weniger ressourcenintensiv als das Pretraining und ermöglicht eine schnellere Anpassung an neue Anwendungsfälle.

Folgende Aspekte sind im Hinblick auf die Feinabstimmung wichtig:

- **Datenqualität:** Die Qualität des Datensatzes ist für das Finetuning entscheidend. Hochwertige und gut klassifizierte Daten gewährleisten, dass das Modell die für die Aufgabe erforderlichen Nuancen erlernen kann.
- **Iterativer Ansatz:** Das Finetuning erfolgt häufig iterativ, wobei zunächst Experimente mit kleinen Datensätzen durchgeführt werden, um die Modelleistung zu beurteilen. Nach der erfolgreichen Validierung können größere Datensätze verwendet werden, um die Genauigkeit weiter zu verbessern.
- **Synthetische Daten:** In Fällen, in denen reale Daten nur spärlich verfügbar sind, können synthetische Daten generiert werden, um den Trainingsdatensatz zu ergänzen. Dieser Ansatz trägt zur Generalisierung bei, ohne dass umfangreiche Daten erhoben werden müssten.

Durch das Finetuning können Organisationen Modelle entwickeln, die hochspezialisiert und auf ihre jeweiligen Fachbereiche zugeschnitten sind und sowohl präzisere als auch relevantere Ergebnisse liefern.

## CONTINUED PRETRAINING (CPT)

Continued Pretraining – auch als „Domain-Adaptive Pretraining“ (d. h., an das jeweilige Fachgebiet anpassendes Pretraining) bekannt – ist ein Zwischenschritt zwischen Pretraining und Finetuning. Im Gegensatz zum Finetuning, das auf die Anpassung an eine konkrete Aufgabe abzielt, beinhaltet CPT ein weiterführendes Training des vortrainierten Modells mit einem großen Datensatz, der für ein bestimmtes Fachgebiet oder eine Sprache spezifisch ist. Dieser Vorgang trägt dazu bei, dass das Modell fundierte Kenntnisse über einen bestimmten Bereich erlangt, bevor es nachfolgend für konkrete Aufgaben optimiert wird.

Häufige Anwendungsszenarien für CPT umfassen:

- **Fachspezifisches Wissen:** Unternehmen, die in hochspezialisierten Fachgebieten wie Gesundheitswesen, Finanzen oder Recht tätig sind, können CPT einsetzen, um das Verständnis des Modells für ihr jeweiliges Gebiet zu verbessern.
- **Sprachspezialisierung:** Allgemeine Modelle liefern in weniger verbreiteten Sprachen möglicherweise minderwertige Ergebnisse. CPT kann verwendet werden, um die Kompetenz des Modells in einer bestimmten Sprache zu verbessern, indem hierfür große Datensätze in der betreffenden Sprache genutzt werden.
- **Programmiersprachen:** CPT kann auch eingesetzt werden, um die Entwicklungsfähigkeiten eines Modells in bestimmten Programmiersprachen zu verbessern, die während der primären Pretraining-Phase möglicherweise nicht ausreichend berücksichtigt wurden.

CPT erfordert in der Regel weniger Ressourcen als das Pretraining, jedoch mehr als das Finetuning. Die benötigte Datenmenge variiert je nach Fachgebiet und Modellgröße. Die Faustregel besagt: Für CPT ist ein Training mit Milliarden von Token erforderlich ist – typischerweise 1 % der Ursprungsgröße des Pretraining-Datensatzes.

Durch die Kombination von CPT und Finetuning können Unternehmen hochspezialisierte Modelle entwickeln, die ein breites Sprachverständnis mit fundiertem Fachwissen verbinden. Diese Kombination eignet sich insbesondere für Anwendungen, die hohe Genauigkeit, einen fachspezifischen Kontext und Zuverlässigkeit erfordern.

## Wirtschaftlichkeit und langfristige Strategie

Die Kosten sind ein weiterer entscheidender Faktor bei der Modellauswahl. Open-Source-Modelle sind in der Anschaffung oft günstiger, da keine Lizenzgebühren anfallen, wohl jedoch Kosten für Wartung, Finetuning und Skalierung. Proprietäre Modelle sind dagegen zwar teurer, bieten jedoch insbesondere für komplexe Anwendungsfälle einen höheren **Mehrwert** im Hinblick auf vorkonfigurierte Optimierung, Sicherheit und Skalierbarkeit.

Letztendlich hängt die Entscheidung zwischen Open-Source- und proprietären Modellen von mehreren Faktoren ab: Leistungsanforderungen, Anpassungsbedarf, Sicherheitsaspekte, Budgetbeschränkungen und langfristiger Support. Mit einem gründlichen Benchmarking können Unternehmen sicherstellen, dass sie das für ihre spezifischen Ziele am besten geeignete LLM auswählen und gleichzeitig für ein ausgewogenes Verhältnis von Kosten, Sicherheit und Leistung sorgen.

Debugging erfordert in den meisten Fällen Entwicklungen und Anpassungen in den folgenden Bereichen:

- **Infrastruktur:** **Mosaic AI Model Training** eliminiert die meisten Infrastrukturprobleme. Beispielsweise wird das Training automatisch unterbrochen, falls GPUs, Netzwerke oder andere Infrastrukturkomponenten ausfallen, und nach erfolgter Behebung des Problems fortgesetzt. Es ist jedoch sinnvoll, die Auslastung zu überwachen, insbesondere bei Verwendung unüblicher Konfigurationen.
- **Lernfortschritt:** Loss-Werte und andere Metriken zu Trainings- und Evaluierungsdaten sollten überwacht werden, damit Daten- und Konfigurationsprobleme früh erkannt werden. Die häufigsten Warnsignalen, auf die zu achten ist, gehören Spitzenwerte bei Datenverlust und Divergenzen. Wir empfehlen, beim Mosaic AI-Training standardmäßig die Option „**Protokollierung für MLflow-Experimente**“ zu aktivieren, um sowohl eine Live-Beobachtung als auch eine nachträgliche Kontrolle zu ermöglichen.
- **Konfigurationen:** Wenn Ihre Konfigurationen nicht einwandfrei sind, treten diese Probleme beim Training bereits frühzeitig auf. Die Lernrate ist die häufigste Konfiguration, die angepasst werden muss.
- **Daten:** Ein häufiges Problem beim Training sind beispielsweise Spitzen beim Datenverlust aufgrund eines unsachgemäßen Shufflings von Datensätzen. Mosaic AI Training vereinfacht das Shuffling mithilfe der **Mosaic Streaming-Bibliothek**. Da dies jedoch mit Kosten verbunden ist, unterstützt Mosaic Streaming **verschiedene Shuffling-Einstellungen**, um geeignete Kompromisse zwischen Qualität und Kosten möglich zu machen. Hohe Loss-Werte können möglicherweise durch striktere Einstellungen für das Shuffling in Mosaic Streaming abgestellt werden. Wenn Ihre Daten beispielsweise aus verschiedenen Buckets (Fachgebieten, Sprachen usw.) stammen und nicht ordnungsgemäß durchmischt („geschuffelt“) sind, wird das Auftreten solcher Loss-Spitzen wahrscheinlicher.



## Kapitel 2: Die Kunst des Prompt-Engineerings

Generative KI entwickelt sich stetig weiter und revolutioniert ganze Branchen. Daher ist das Prompt-Engineering einer der Schlüsselbereiche, denen die Unternehmen verstärkt Aufmerksamkeit widmen sollten. Diese Schlüsselkompetenz sorgt dafür, dass KI-Modelle wie Large Language Models menschliche Absichten verstehen und korrekte, sinnvolle und kreative Ergebnisse liefern. Ganz gleich, ob Sie gerade erst in das Thema KI einsteigen oder komplexe KI-Systeme weiter optimieren möchten: Wenn Sie das volle Potenzial von GenAI-Technologien ausschöpfen wollen, ist Prompt-Engineering ein wichtiger Erfolgsfaktor.

In diesem Kapitel werden wir ausführlich betrachten, was Prompt-Engineering ist, warum es so wichtig ist und wie Sie es gezielt einsetzen, um die optimale Performance Ihrer KI-Anwendungen sicherzustellen.

### Was ist Prompt-Engineering?

Prompt-Engineering ist ein Spezialgebiet, dessen Schwerpunkt auf der Erstellung, Verfeinerung und Optimierung von Eingaben in natürlicher Sprache (so genannte „Prompts“) liegt, die KI-Modellen sagen, wie sie welche Aufgaben ausführen sollen. Die Bandbreite dieser Aufgaben reicht von der Textgenerierung über das Schreiben von Code bis hin zur Bilderstellung. Im Kern geht es beim Prompt-Engineering darum, menschliche Absichten in eine Form zu übersetzen, die KI verstehen und korrekt verarbeiten kann.

Mit dem Aufkommen generativer KI ist das Prompt-Engineering zu einer unverzichtbaren Kompetenz geworden – gerade angesichts der rasanten Entwicklung leistungsstarker Modelle wie ChatGPT von OpenAI, Llama von Meta und Gemini von Google. Diese Modelle sind durchaus in der Lage, Texte in einer Qualität zu erstellen, die menschlich wirkt, aber ihr Erfolg hängt maßgeblich von der Güte der zugeführten Prompts ab. Schlecht formulierte Prompts können zu irrelevanten oder falschen Ergebnissen führen, während sorgfältig gestaltete Prompts Treffsicherheit, Kreativität und Relevanz verbessern.

Ein Beispiel: Ein E-Commerce-Unternehmen implementiert einen Chatbot, der auf einem GenAI-Modell basiert. Dieser soll eine Vielzahl von Kundenanfragen bearbeiten können – vom Bestellstatus bis hin zu Rückgabebedingungen. In diesem Fall formulieren Prompt-Engineers gezielte Anweisungen, damit der Chatbot Kundenanfragen effizient bearbeiten, erforderliche Informationen (wie Artikelnummern) abfragen, hilfreiche Antworten geben und komplexe Anliegen bei Bedarf an menschliche Fachkräfte weiterleiten kann. Je besser die Prompts, desto effektiver arbeitet der Chatbot. Das freut die Kundschaft und entlastet das operative Team.

Folgende Ansätze werden beim Prompt-Engineering unterschieden:

- **Zero-Shot-Prompting:** Fordert das LLM auf, eine Aufgabe ohne Beispiele auszuführen – allein basierend auf seinem vorhandenen Wissen.
- **Few-Shot-Prompting:** Gibt ein oder mehrere Beispiele im Prompt vor, um die Antwort des Modells gezielt zu steuern.
- **Chain-of-Thought-Prompting:** Hierbei werden komplexe Probleme in logische Schritte zerlegt, sodass das Modell die Aufgaben schrittweise durchdenken kann.
- **Self-Refine-Prompting:** Das Modell löst zunächst ein Problem, bewertet anschließend seine eigene Antwort und verfeinert sie basierend auf Feedback.
- **Directional-Stimulus-Prompting:** Verwendet Hinweise oder Schlagwörter, um die Antwort des Modells in eine bestimmte Richtung zu lenken.
- **Iteratives Prompting:** Baut auf vorherigen Antworten auf und stellt Folgefragen zur Vertiefung.
- **Kontextinjektion:** Stellt zusätzliche Kontextinformationen bereit, um die Relevanz der Antworten zu verbessern.
- **Role-Playing-Prompting:** Beschreibt im Prompt eine spezifische Persona, um die Ergebnisse des LLM in eine bestimmte Richtung zu lenken.

## Wie Prompt-Engineering die Qualität von Modellausgaben steigert

Effektives Prompt-Engineering bedeutet nicht nur, Eingaben für die KI zu erstellen, sondern diese Eingaben auch zu optimieren, um eine möglichst hohe Qualität der KI-Ergebnisse zu erzielen. Dazu lassen sich verschiedene Strategien einsetzen, die das Verhalten der KI steuern, ihre Kreativität fördern und potenzielle Fehler minimieren.

### 1. ANGABE VON KONTEXT UND SPEZIFITÄT

Gut formulierte Prompts vermitteln der KI den Kontext, der notwendig ist, um die Aufgabe besser zu verstehen. Ganz gleich, ob eine Frage beantwortet, kreative Inhalte erstellt oder Probleme gelöst werden sollen: Je mehr Kontext Sie angeben, desto relevanter wird das Ergebnis sein. Anstatt beispielsweise einer KI die Anweisung „Gib mir Informationen zum Markt“ zu geben, könnten Sie präziser formulieren: „Fasse die wichtigsten Trends auf dem globalen Technologiemarkt im Jahr 2025 zusammen.“ Diese zusätzliche Spezifizierung hilft der KI, ihre Antwort einzugrenzen, und generiert spezifischere und nützlichere Informationen.

## 2. GEFÜHRTE GEDANKENGÄNGE FÜR KOMPLEXERE AUFGABEN

Bei anspruchsvollen oder mehrstufigen Aufgaben, können fortgeschrittene Prompt-Engineering-Techniken wie Chain-of-Thought-Prompting eingesetzt werden, um das Problem in kleinere, überschaubare Schritte zu zerlegen. Mithilfe dieser Methodik leiten Sie die KI dazu an, das Problem logisch zu durchdenken, was zu verständlicheren und treffenderen Antworten führt. Ein Beispiel: Soll die KI eine Rechenaufgabe lösen, hilft ein Chain-of-Thought-Prompt dabei, den Lösungsweg Schritt für Schritt aufzubauen – bis hin zum Ergebnis.

## 3. ELIMINIERUNG VON MEHRDEUTIGKEITEN UND VERZERRUNGEN

Eindeutigkeit ist beim Prompt-Engineering der Schlüssel zum Erfolg. Ambivalente Prompts können die KI verwirren und irrelevante Antworten zur Folge haben. Zudem können unzulänglich formulierte Prompts unbeabsichtigt Vorurteile festigen, sei es zu Geschlecht, Ethnie oder sonstigen sensiblen Themen. Indem Sie die gewünschte Ausgabe unmissverständlich angeben und eine vielfältige und inklusive Sprache in Ihre Prompts aufnehmen, können Sie solche Risiken reduzieren und die Fairness der KI-Antworten verbessern.

## 4. VERBESSERUNG DER KREATIVITÄT UND DER ERGEBNISBANDBREITE

Bei kreativen Anwendungen lassen sich KI-Modelle mit Prompts dazu bewegen, vielgestaltige und innovative Ergebnisse zu erzeugen. Beispielsweise können Prompts für kreatives Schreiben ein Genre, einen Tonfall und bestimmte Elemente vorgeben, die in der Geschichte enthalten sein sollen, und so Stil und Atmosphäre der KI-Kreation beeinflussen. Mit gezielten Prompts können Unternehmen kreative Inhalte generieren, die sowohl individuell als auch markenkonform sind.

## 5. MINIMIEREN VON HALLUZINATIONEN UND FEHLERN

Generative KI-Modelle mögen zwar leistungsstark sein, sind aber nicht unfehlbar. Schlecht formulierte Prompts können zu Halluzinationen führen: Fälle, in denen die KI fehlerhafte oder erfundene Informationen generiert. Wirkungsvolles Prompt-Engineering minimiert solche Risiken durch präzise Aufgabenstellungen, klare Anweisungen und ein realistisches Verständnis der Stärken und Grenzen der KI. Darüber hinaus sollten KI-Ergebnisse auf ihre Richtigkeit abgeklöpft werden, insbesondere in Anwendungen, bei denen sachliche Korrektheit unverzichtbar ist.

## Prompt-Engineering im Kontext der Entwicklung von KI-Agenten

Mit dem Aufkommen fortschrittlicher KI-Tools wie Databricks Mosaic AI ist das Prompt-Engineering zu einem noch zentraleren Faktor bei der Entwicklung leistungsstarker KI-Agenten für den Geschäftseinsatz geworden. Databricks Mosaic AI bietet eine umfassende Plattform zur Entwicklung von KI-Anwendungen, zur Integration von ML-Modellen und zur skalierbaren Verwaltung von KI-Workflows. Beim Prompt-Engineering im Mosaic AI-Ökosystem steht die Optimierung der Eingaben in die Modelle von Databricks im Mittelpunkt – so entstehen KI-Agenten, die präzise, effizient und kreativ arbeiten.

Databricks Mosaic AI gestattet eine einfache Integration in LLMs und andere fortschrittliche KI-Modelle und bietet robuste Tools für die Entwicklung intelligenter Agents, die eine Vielzahl von Aufgaben ausführen können. Durch gezieltes Prompt Engineering für den konkreten Zweck des jeweiligen Agents – von der Automatisierung des Kundensupports bis hin zur Generierung von Marktübersichten – können Unternehmen das Potenzial ihrer KI-Systeme voll ausschöpfen.

### OPTIMIEREN VON KI-AGENTS FÜR AUFGABENSPEZIFISCHE ANFORDERUNGEN

Beim Erstellen von KI-Agenten in Databricks Mosaic AI können Sie die Agents durch geschicktes Prompt-Engineering auf bestimmte Geschäftsaufgaben und Kundenbedürfnisse zuschneiden. Ein Prompt kann den KI-Agent beispielsweise anleiten, personalisierte Empfehlungen auf Basis von Nutzerpräferenzen zu geben oder die Verlaufsdaten eines Kunden auszuwerten, um maßgeschneiderten Support zu leisten. Mithilfe gut formulierter Prompts kann der KI-Agent präzise und kontextbezogene Antworten und trägt so zu einem deutlich besseren Benutzererlebnis und einer höheren betrieblichen Effizienz bei.

### VERWENDEN VON BENUTZERFEEDBACK ZUR FORTLAUFENDEN VERBESSERUNG

Einer der Vorteile des Einsatzes von Databricks Mosaic AI ist die Möglichkeit, die Performance Ihrer KI-Agenten stetig zu beobachten, zu analysieren und zu verbessern. Durch das Testen verschiedener Prompt-Varianten, das Erfassen von Benutzerfeedback und die Überwachung der Agent-Performance können Sie über Prompts iterieren und nachjustieren, um die Ergebnisse schrittweise zu optimieren. Diese kontinuierliche Prompt-Verfeinerung stellt sicher, dass Ihre KI-Agenten präzise, relevant und effektiv bleiben – auch bei sich verändernden Geschäftszielen.

Zusammenfassend lässt sich sagen, dass effektives Prompt-Engineering eine wichtige Grundlage für die Entwicklung robuster und zuverlässiger KI-Agenten mit Databricks Mosaic AI ist. Es trägt nicht nur zur Verbesserung der Performance von KI-Modellen bei, sondern hilft Unternehmen auch, intelligentere, flexiblere und skalierbare Systeme zu entwickeln, um den Herausforderungen einer dynamischen digitalen Landschaft gerecht zu werden.



## Kapitel 3: Entwicklung von KI-Agent-Systemen

KI-Agent-Systeme kombinieren verschiedene KI-Technologien, um komplexe Aufgaben autonom zu erledigen, Entscheidungen zu verbessern und Abläufe effizienter zu gestalten. Die Systeme verbinden Komponenten wie Large Language Models, klassische Machine-Learning-Modelle und Unternehmensdaten zu intelligenten, kontextsensitiven Lösungen. KI-Agent-Systeme decken ein breites Spektrum an Anwendungsfällen ab – von automatisiertem Kundensupport über personalisierte Empfehlungen bis hin zu Betrugserkennung und Trendprognosen. Durch die Verbindung von Datenanalyse und Sprachverarbeitung liefern sie präzise, datengesteuerte Erkenntnisse und bleiben flexibel im Umgang mit unstrukturierten Informationen. So werden sie branchenübergreifend unverzichtbar: Sie optimieren Abläufe, verbessern die Nutzerinteraktion und ermöglichen skalierbare KI-Lösungen.

Databricks Mosaic AI bietet eine einheitliche Plattform für Integration und Governance aller KI-Bausteine und schafft so die Basis für leistungsstarke KI-Agent-Systeme. Dank Lakehouse-Architektur ermöglicht Mosaic AI einen sicheren Zugriff auf Unternehmensdaten – für präzise, fachspezifisch angepasste Agent-Systeme. Dank Unterstützung von Open-Source- und proprietären Modellen können Unternehmen mit Mosaic AI die für ihre Anforderungen am besten geeigneten Lösungen auswählen und kombinieren. Automatisierte Tools für Bewertung, Finetuning und Optimierung ermöglichen schnelle Iterationen und kontinuierliche Verbesserungen. Mit einem robusten Governance-Framework sorgt Mosaic AI für leistungsfähige, regelkonforme und auf die Geschäftsziele ausgerichtete KI-Agent-Systeme, und bietet volle Transparenz und Kontrolle von der Datenerfassung bis zur Bereitstellung.

### Was ist ein KI-Agent?

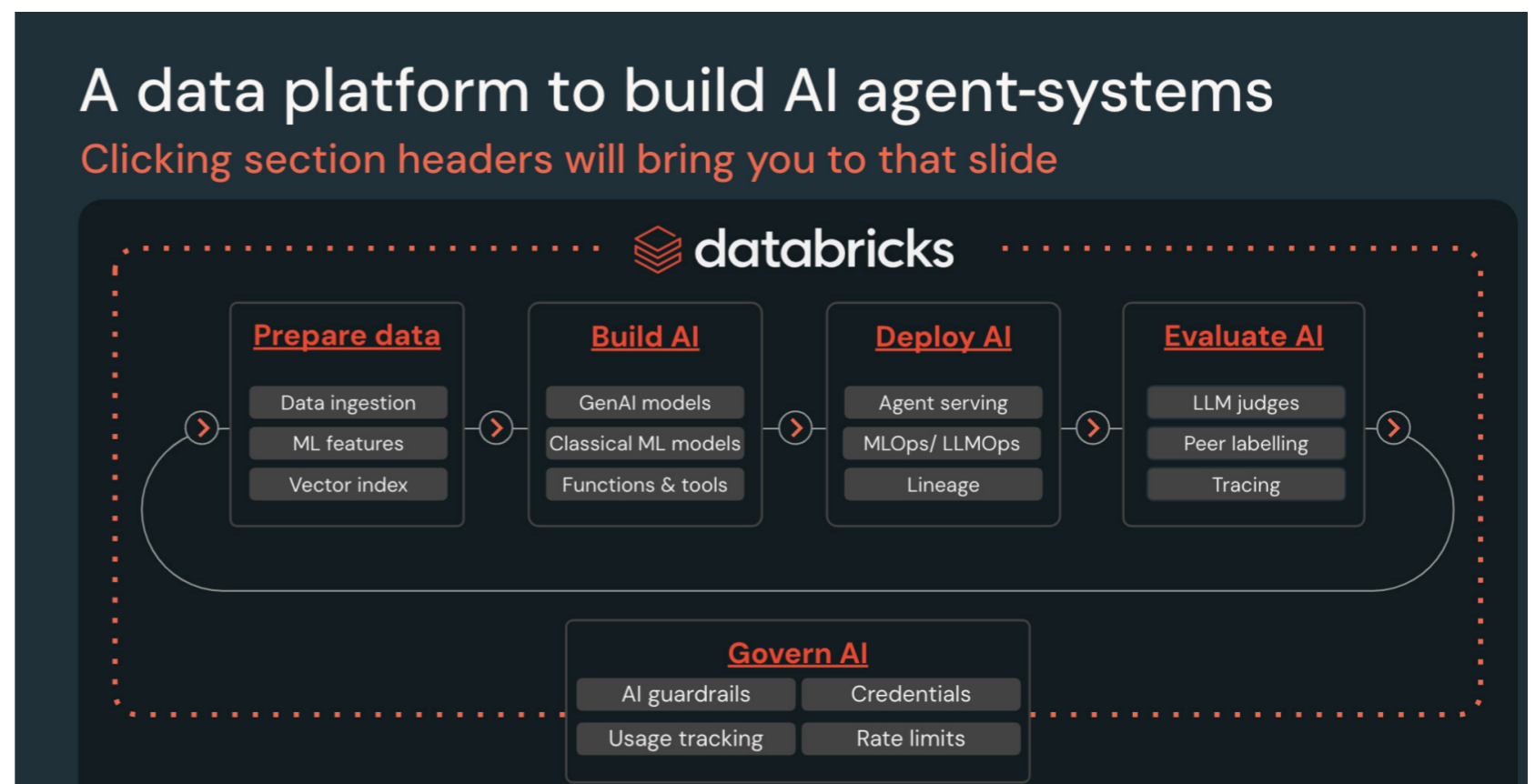
Wer einen effektiven KI-Agents entwickeln möchte, kommt um ein KI-Agent-System nicht herum – ganz gleich, ob es sich um einen Einzel-Agent oder mehrere interagierende Agents handelt. KI-Agents sind intelligente Anwendungen zur Automatisierung von Aufgaben und Produktivitätssteigerung. Sie analysieren Informationen, treffen Entscheidungen und handeln zielgerichtet, wodurch Zeit und Ressourcen für strategische Initiativen frei werden. Ein paar Beispiele:

- **Kundendienst-Agent:** Interagiert mit Kunden, um ihre Anfragen zu verstehen und relevante Antworten zu geben. Das ermöglicht eine effiziente Problemlösung und eine höhere Kundenzufriedenheit.
- **Agent zur Kampagnenerstellung:** Analysiert Daten, erkennt Zielgruppen und erstellt personalisierte Kampagnen – etwa zur Steigerung der Markenbekanntheit oder zur Umsatzförderung.
- **Agent zur Codegenerierung:** Unterstützt Entwickler beim Schreiben, Debuggen und Optimieren von Code auf Basis vorgegebener Anforderungen und vereinfacht den Softwareentwicklungsprozess.

## Was ist ein KI-Agent-System?

Mit einem KI-Agent-System können Unternehmen intelligente Agents entwickeln und einsetzen, die komplexe Aufgaben ausführen können. Anders als eigenständige Modelle integrieren solche Systeme verschiedene Komponenten wie Large Language Models, klassische Machine-Learning-Modelle, Unternehmensdaten und externe Tools, um bestimmte Ziele effizient zu erreichen. Außerdem verfügen KI-Agent-Systeme über integrierte Evaluationsmethoden und Governance-Mechanismen, um Qualität, Nachvollziehbarkeit und die Einhaltung organisatorischer Standards zu gewährleisten.

## Kernphasen der Entwicklung eines KI-Agent-Systems



- **Daten aufbereiten:** Daten werden organisiert und vorverarbeitet, um sicherzustellen, dass sie bereinigt, zugänglich und relevant für die Entscheidungsfindung und Agent-Interaktionen sind.
- **Agents erstellen:** GenAI-Modelle, klassische ML-Modelle und Tools werden kombiniert, um Agents zu erstellen, die auf bestimmte Aufgaben zugeschnitten sind.
- **Agents implementieren:** Agents werden so bereitgestellt, dass sie von Nutzern und Systemen sicher, effizient und performant genutzt werden können.
- **Performance bewerten:** Die Ausgaben von Agents werden gemessen und auf Erfüllung der Ziele geprüft. Ausgehend von Feedback werden iterative Verbesserungen vorgenommen.
- **Abläufe regulieren:** Sicherheit, Compliance und ethische Standards bleiben gewahrt, während Agent-Aktivitäten transparent und nachvollziehbar protokolliert werden.

## Zentrale Komponenten eines KI-Agent-Systems

KI-Agent-Systeme bestehen aus zentralen Komponenten, die zusammen komplexe Funktionen ermöglichen.

### LLM-/ZENTRALER AGENT

Die zentrale Komponente eines KI-Agent-Systems ist ein vortrainiertes Universal-LLM, das Daten verarbeitet und versteht. Solche Modelle werden durch sorgfältig ausgearbeitete Prompts aktiviert, die wichtigen Kontext liefern, die Interaktion steuern und Ziele sowie genutzte Tools vorgeben.

Agent-Frameworks ermöglichen Anpassungen, sodass das Modell eine eindeutige Identität und spezifisches Fachwissen erlangt. Diese Flexibilität sorgt dafür, dass der LLM-Agent vielfältige Aufgaben präzise ausführen kann, was ihn zu einem wertvollen Tool für Unternehmensanwendungen macht.

## PLANER

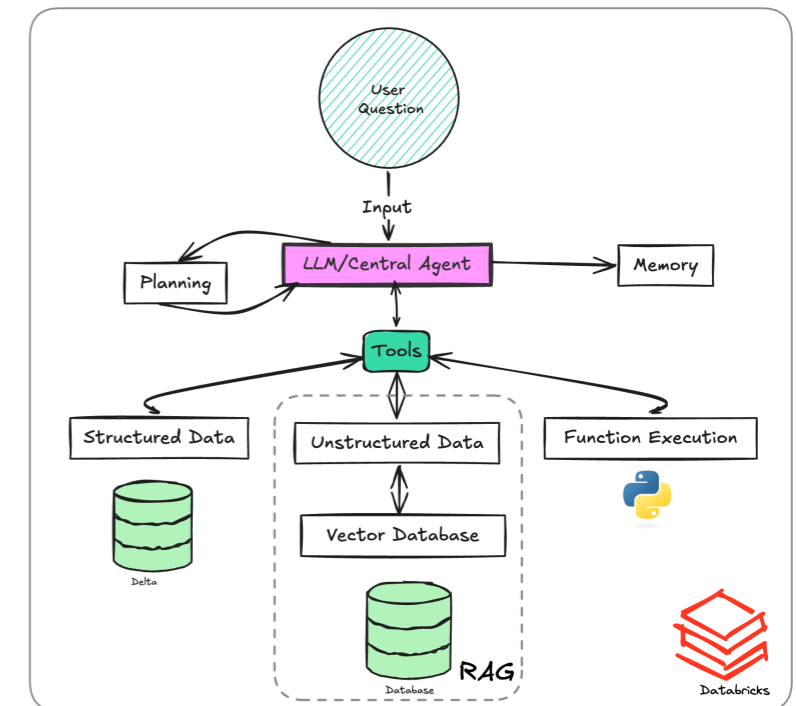
Die Planungskomponente unterteilt komplexe Aufgaben in kleinere, überschaubare Teilaufgaben und stellt deren Ausführung sicher. Sie nutzt Logikverfahren wie Chain-of-Thought oder Entscheidungsbäume, um die optimale Vorgehensweise zu bestimmen.

Ist der Plan erstellt, wird er mithilfe interner Feedbackmethoden wie ReAct oder Reflexion überprüft und verfeinert – zur iterativen Verbesserung der Entscheidungslogik. In Single-Agent-Systemen übernimmt meist ein LLM Planung und Koordination, während Multi-Agent-Frameworks diese Aufgaben an spezialisierte Agents delegieren können.

## TOOLS

Tools sind die operativen Bausteine eines KI-Agent-Systems. Sie führen bestimmte Aufgaben gemäß den Anweisungen des zentralen LLM aus. Tools reichen von API-Aufrufen und Python-Skripten über SQL-Abfragen und Websuchen bis hin zu maßgeschneiderten Unternehmensfunktionen wie Databricks AI/BI Genie.

Durch die Integration von Tools können LLM-Agents Workflows ausführen, Beobachtungen erfassen und Teilaufgaben effizient erledigen. Allerdings ist es für die Entwicklung solcher Anwendungen unerlässlich, die Interaktionsdauer zu berücksichtigen. Lange Dialoge können die Kontextlimits eines LLM erschöpfen, was dazu führen kann, dass bestimmte Details vergessen werden. Ob Single-Threaded, Multi-Threaded oder Schleifen: Die Ablaufsteuerung ist entscheidend für die Bewältigung immer komplexerer Entscheidungsabläufe.



In der Abbildung ist ein einzelnes leistungsstarkes LLM der Schlüssel zur Entscheidungsfindung. Anhand der Frage des Anwenders wird ermittelt, welcher Pfad für den Entscheidungsablauf zu wählen ist. Dabei können Tools Aktionen ausführen, Zwischenergebnisse speichern, weitere Schritte planen und das Ergebnis an den Nutzer zurückgeben.




## ARBEITSSPEICHER

Speicher kann die Architektur eines KI-Agents erheblich verbessern, da er Informationen für eine effektive Entscheidungsfindung ablegen und abrufen kann.

- **Kurzzeitspeicher:** Temporärer Speicher für den unmittelbaren Kontext, der nach Abschluss einer Aufgabe gelöscht wird
- **Langzeitspeicher:** Persistenter Speicher, der häufig in externen Datenbanken (z. B. Vektordatenbanken) verwaltet wird und dem Agent hilft, Muster zu erkennen, aus früheren Interaktionen zu lernen und fundierte Entscheidungen bei künftigen Aufgaben zu treffen

Durch die Integration beider Speichertypen können Agents maßgeschneiderte Antworten geben und sich nach und nach immer besser an die Vorlieben der Benutzer anpassen. Wichtig ist hierbei die Unterscheidung zwischen dem Speicher eines Agents und dem Dialogspeicher eines LLM, die jeweils verschiedenen Zwecken dienen.

## Anwendungsfälle für KI-Agent-Systeme

 Financial Services	 Healthcare & Life Sciences	 Comm, Media & Entertainment	 Retail & Consumer Goods	 Manufacturing	 Public Sector
Fraud monitoring and predictions	Biomedical literature summarization & discovery	Hyper-personalization for customer experience (CX)	Try before you buy with virtual fitting rooms	Delightful, personalized customer experiences	Analysis of open-source intelligence
Automating compliance data gathering	Clinical trial optimization	Enhancing customer support and self-service	Optimizing demand prediction and inventory	Increasing productivity and efficiency in operations	Modernizing legacy code bases
Accelerate underwriting and claims processing in insurance	Health insurance claim processing	Intelligent content creation and curation	Generate innovative product designs	Prescriptive and proactive field service	Regulatory compliance assistance

KI-Agent-Systeme sind vielseitige Lösungen, die in vielen Unternehmensbereichen eingesetzt werden können und erhebliche Verbesserungen bei Effizienz, Korrektheit und Produktivität bieten. Im Folgenden finden Sie einige praktische Beispiele dafür, wie KI-Agent-Systeme in der Praxis eingesetzt werden können:

### AUTOMATISIERUNG DES KUNDENDIENSTS

KI-Agent-Systeme können Kundeninteraktionen optimieren, indem sie mithilfe intelligenter Chatbots oder virtueller Assistenten schnelle und korrekte Antworten auf Routineanfragen liefern. So sinkt die Arbeitslast für menschliche Mitarbeiter, die sich auf komplexere Fälle konzentrieren können. Beispielsweise kann ein KI-Agent-System allgemeine Serviceanfragen zur Auftragsverfolgung oder Fehlerbehebung bearbeiten und kniffligere Probleme automatisch an menschliche Mitarbeiter weiterleiten.

### UNTERSTÜTZUNG IN VERTRIEB UND MARKETING

Vertriebs- und Marketingteams können von KI-Agent-Systemen profitieren, die wiederkehrende Aufgaben wie die Lead-Qualifizierung, das Verfassen von Follow-up-Mails und die Dateneingabe automatisieren. Zudem können diese Systeme Kundendaten analysieren, um Erkenntnisse und personalisierte Empfehlungen zu generieren und Kampagnen zu effektiver zu machen. Ein KI-Agent-System kann z. B. Leads anhand von Interaktionsdaten priorisieren oder personalisierte Aktionen vorschlagen, die auf die individuellen Kundenpräferenzen zugeschnitten sind.

### DATENANALYSE UND REPORTING

KI-Agent-Systeme leisten bei der Automatisierung von Datenerfassung, -analyse und -berichterstattung ausgezeichnete Arbeit. Sie verarbeiten große Datenmengen in Echtzeit und liefern Unternehmen ohne manuellen Aufwand zeitnah Erkenntnisse. Ein Handelsunternehmen könnte damit etwa Daten aus verschiedenen Quellen zusammenführen und automatisiert wöchentliche Verkaufsberichte oder Leistungs-Dashboards erstellen und so den Bedarf an manueller Datenaggregation deutlich reduzieren.

### PERSONALISIERTE EMPFEHLUNGEN

Durch die Analyse von Verhalten und Vorlieben der Benutzer können KI-Agent-Systeme in Branchen wie E-Commerce, Unterhaltung und Reisen hochgradig personalisierte Empfehlungen geben. Beispielsweise könnte ein Online-Händler ein KI-Agent-System nutzen, um auf Grundlage des Surf- und Kaufverhaltens von Kunden Produkte zu empfehlen. Ähnlich könnte eine Streaming-Plattform auf Basis von Sehgewohnheiten Inhaltsvorschläge machen.

## AUFGABENAUTOMATISIERUNG IM BETRIEB

Im Betrieb können KI-Agent-Systeme Routineaufgaben wie Terminplanung, Lagerhaltung und Auftragsabwicklung automatisieren und so Effizienz und Fehlerfreiheit erheblich verbessern. In der Fertigung könnte ein KI-Agent-System beispielsweise Lagerbestände überwachen und bei niedrigem Bestand automatisch nachbestellen, wodurch menschliche Fehler minimiert und Bestandslücken vermieden werden.

## BETRUGSERKENNUNG UND-PRÄVENTION

KI-Agent-Systeme sind äußerst effektiv bei der Erkennung und Eindämmung von Betrug, denn sie können Transaktionen fortlaufend überwachen und Anomalien in Echtzeit erkennen. Ein Finanzinstitut könnte das System nutzen, um auffällige Vorgänge wie ungewöhnliche Abhebemuster automatisch zur Überprüfung zu markieren. Durch die proaktive Erfassung potenzieller Bedrohungen könnten Unternehmen das Betrugsrisiko erheblich senken.

### Spezielle Anwendungsfälle

Zwar gehören **RAG-Systeme** (die LLMs mit Datenabruf kombinieren, um kontextbezogene Antworten zu liefern) zu den bekanntesten KI-Agent-Systeme, bilden aber nur einen Teil des Potenzials ab. RAG-Systeme eignen sich besonders für Anwendungen mit unstrukturierten Daten, wie beispielsweise Kundensupport und Wissensabfrage. Dagegen gehen andere Agent-Systeme über reine Abfrageaufgaben hinaus.

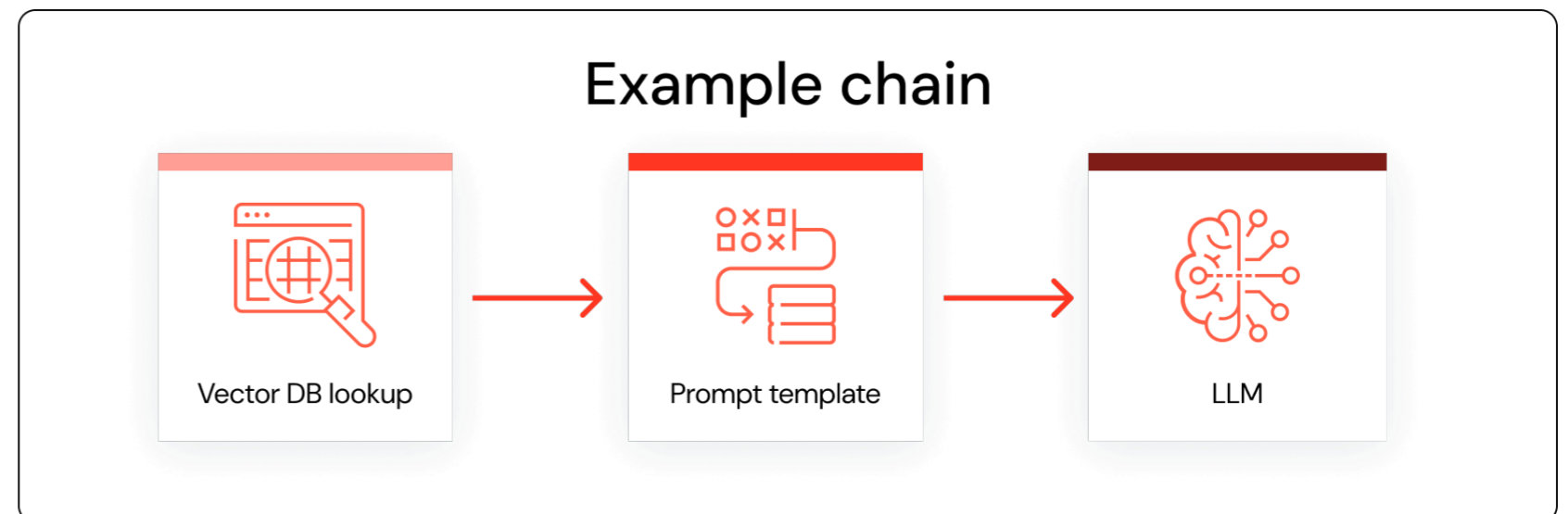
In **strukturierten Datenumgebungen** spielen klassische Machine-Learning-Modelle eine zentrale Rolle. In Kombination mit LLMs können solche Modelle die Entscheidungsfindung verbessern, indem sie präzise, datengestützte Erkenntnisse liefern. Bei der Betrugserkennung analysieren klassische ML-Modelle strukturierte Transaktionsdaten, während ein LLM-gestützter Agent die Kundeninteraktionen im Zusammenhang mit auffälligen Transaktionen übernehmen könnte.

Ein weiteres Beispiel sind **Multi-Agent-Systeme**, bei denen mehrere spezialisierte Agents gemeinsam komplexe Aufgaben lösen. Bei solchen Systemen können Agents Rollen wie Planung, Ausführung oder Datenabruf übernehmen, wobei jeder Agent für eine bestimmte Funktion entwickelt wurde. Durch die Orchestrierung der Agents können Unternehmen hochgradig anpassungsfähige KI-Lösungen entwickeln, die komplexe Abläufe wie die Lieferkettenoptimierung oder Multichannel-Kundenbindung steuern.

## Kapitel 4: Fusion von LLMs und Wissensdatenbanken

Dank RAG können Unternehmen handelsübliche KI-Systeme durch Integration externer Wissensressourcen aufwerten und so die Treffsicherheit und Kontextrelevanz erhöhen, ohne das zugrunde liegende Verhalten der Modelle zu ändern. Mittels Kombination von Benutzer-Prompts mit relevanten, abgerufenen Informationen generiert RAG erweiterte Prompts, die es Large Language Models wie GPT-4 oder Llama 3 ermöglichen, präzisere und informativere Antworten zu liefern.

Im Gegensatz zum einfachen Prompt-Engineering erfordert RAG ein fortschrittliches Abrufsystem – häufig eine Vektordatenbank – und die nahtlose Einbindung der Daten in den LLM-Workflow. Diese zusätzliche Komplexität lohnt sich, da sie die Qualität der KI-Ergebnisse deutlich steigert.



## NUTZEN VON RAG FÜR DEN ZUGRIFF AUF EXTERNE INFORMATIONEN

RAG bietet mehrere wesentliche Vorteile bei der Integration externer Daten:

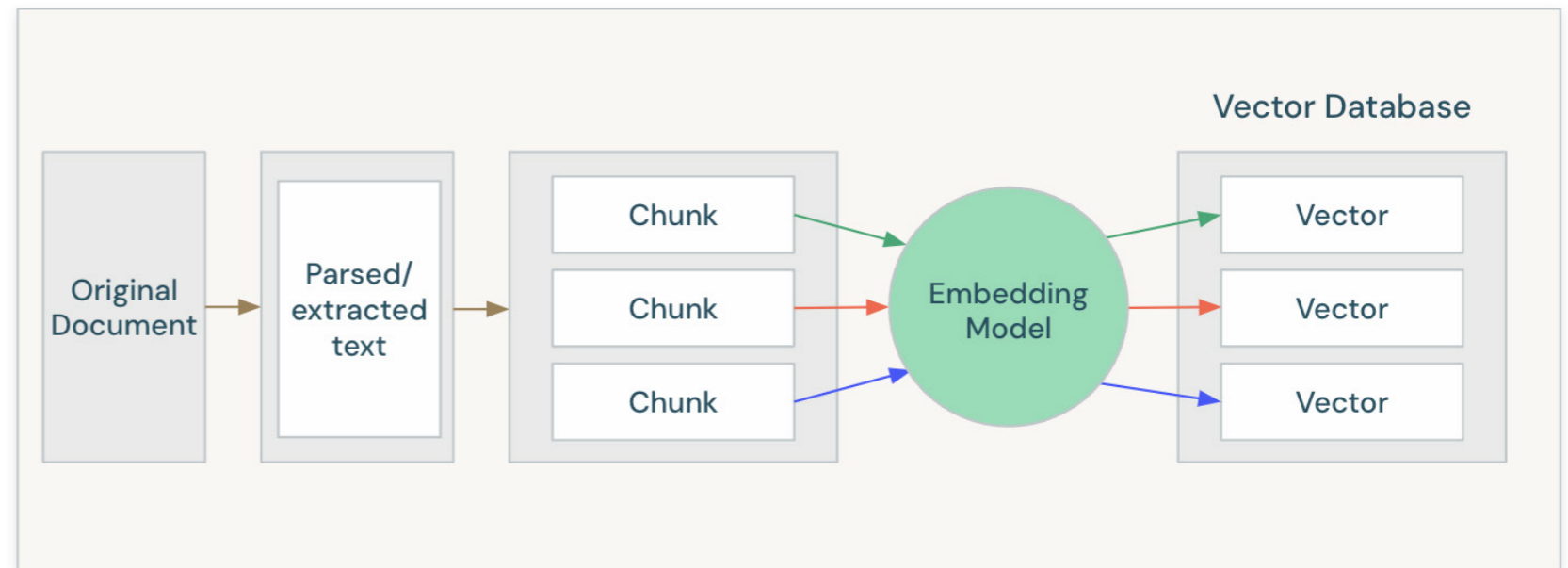
- 1. Datenmodularität:** Unternehmen können Datenquellen hinzufügen oder entfernen, ohne das Modell neu trainieren zu müssen, was mehr Flexibilität bietet.
- 2. Zugriffskontrolle:** Es können differenzierte Berechtigungen implementiert werden, um sicherzustellen, dass nur autorisierte Benutzer Zugriff auf bestimmte Datensätze haben.
- 3. Modellflexibilität:** RAG ermöglicht den Vergleich verschiedener LLMs ohne umfangreiches Retraining mit neuen Daten.

Während RAG-Systeme im Allgemeinen weniger Ressourcen erfordern als Pretraining oder Finetuning, können Kosten und Komplexität je nach Umfang und Architektur des Abrufsystems variieren. Beispielsweise ist eine latenzarme Vektordatenbank, die Millionen von Datensätzen verarbeiten kann, teurer als ein kleineres System mit höherer Latenz. Trotz dieser Kosten bleibt RAG eine effiziente Möglichkeit, die Fähigkeiten eines LLM ohne substantielle Vorabinvestitionen zu optimieren.

### Verbessern der Modelleistung mit RAG

RAG bietet eine Reihe von Leistungsvorteilen: weniger Halluzinationen, bessere fachspezifische Intelligenz und optimierte Antwortgenauigkeit. Allerdings hängt die Leistungsfähigkeit von RAG wesentlich von den Fähigkeiten des zugrunde liegenden LLM ab. Bleibt die Performance hinter den Erwartungen zurück, sollten Unternehmen fortgeschrittenere Anpassungsstrategien wie Finetuning oder Continued Pretraining prüfen.

Durch Entwicklung eines grundlegenden Verständnisses von LLMs und das Ausloten des RAG-Potenzials können Unternehmen Performance-Lücken erkennen und Ressourcen gezielter zuweisen, bevor sie sich für komplexere Lösungen entscheiden.



### RAG-DATENQUELLEN UND WORKFLOW

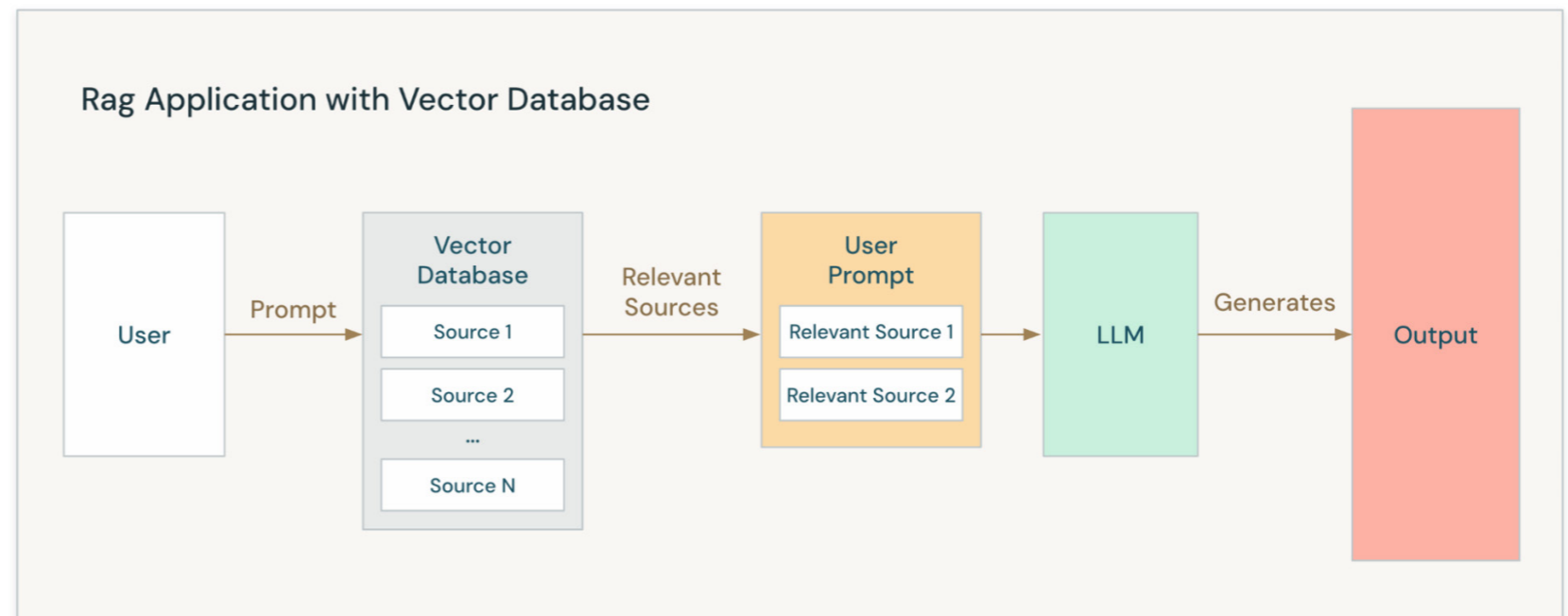
RAG-Systeme können verschiedene Datenstrukturen wie die folgenden verarbeiten:

- **Text:** PDF-Dateien, Artikel und Code-Repositorys
- **Multimedia:** Podcasts und Videos
- **Strukturierte Datenbanken:** Relationale oder NoSQL-Datenbanken

## Unstrukturierte Daten und Vektordatenbanken

Unstrukturierte Daten verfügen über keine vordefinierte Organisation, was eine direkte Abfrage erschwert. RAG-Workflows machen aus unstrukturierten Rohdaten diskrete, durchsuchbare Datenblöcke. Das geschieht in folgenden Schritten:

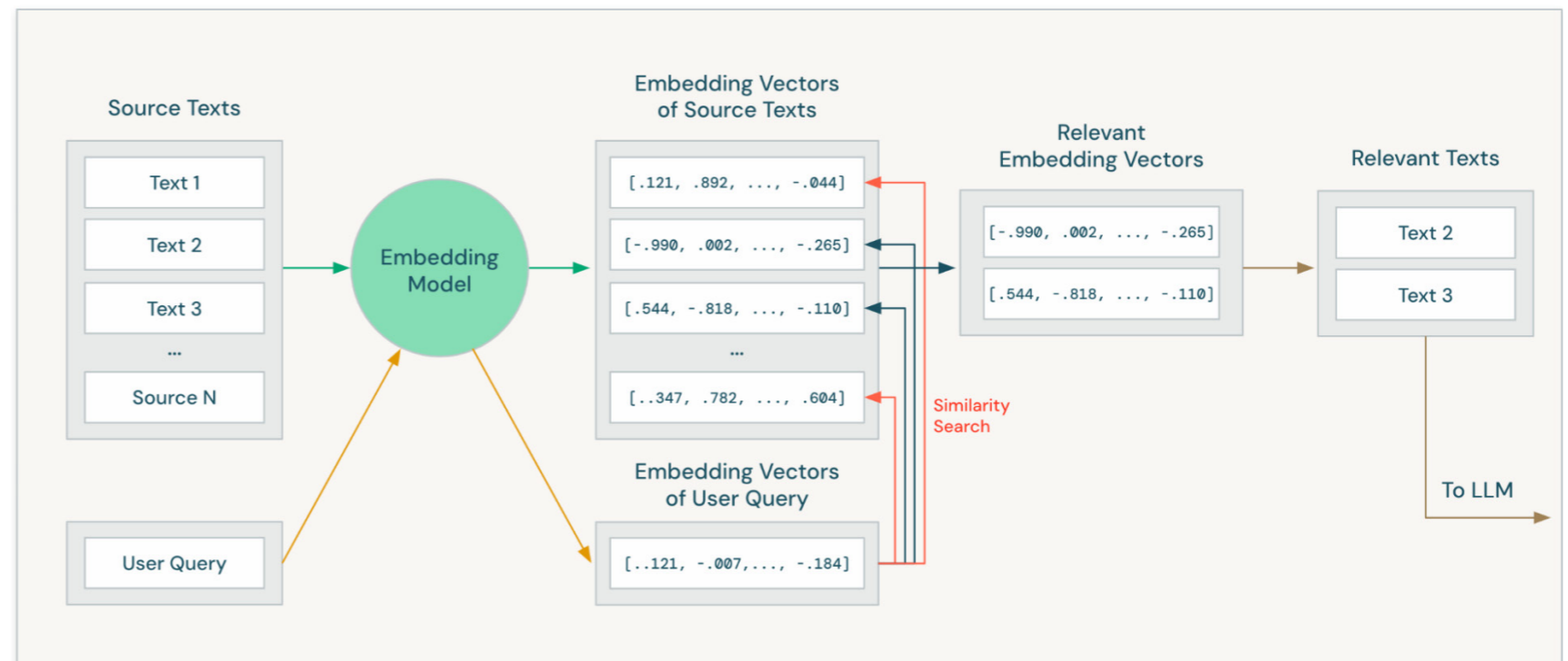
- 1. Rohdokumente analysieren:** PDF-Dateien, Bilder und Multimedia-Dateien werden mithilfe von optischer Zeichenerkennung (OCR) und Textextraktionswerkzeugen in nutzbare Textformate konvertiert.
- 2. Metadaten extrahieren (optional):** Metadaten (z. B. Dokumenttitel oder URLs) werden extrahiert und genutzt, um die Präzision von Suchanfragen zu verbessern.
- 3. Dokumente fragmentieren:** Die analysierten Dokumente werden in kleinere Abschnitte zerlegt, die in das Kontextfenster des Embedding-Modells und des LLM passen.
- 4. Fragmente einbetten:** Mit Embedding-Modellen werden die Fragmente in hochdimensionale numerische Vektoren transformiert, die ihre semantische Bedeutung abbilden.
- 5. In einer Vektordatenbank indizieren:** Embeddings und zugehöriger Text werden in einer Vektordatenbank gespeichert, die auf schnelle und effiziente Abfragen ausgelegt ist.



## Optimieren der Vektorsuche

Die Vektorsuche bildet das Herzstück von RAG-Anwendungen. Sie findet relevante Informationen, indem sie Benutzeranfragen einbettet und mit gespeicherten Embeddings abgleicht. Dabei hängt eine effiziente Suche von mehreren Faktoren ab:

- **Einbettungsmodelle:** Präzise Embeddings-Modelle, die semantische Bedeutungen erfassen, sind zentral für präzise Treffer bei der Suche.
- **Vektorindizes:** Diese Indizes organisieren Embeddings, um schnelle Ähnlichkeitsberechnungen zu ermöglichen.
- **Algorithmen:** Abrufalgorithmen müssen Präzision und Effizienz in Einklang bringen, um schnelle und fehlerfreie Ergebnisse zu gewährleisten.



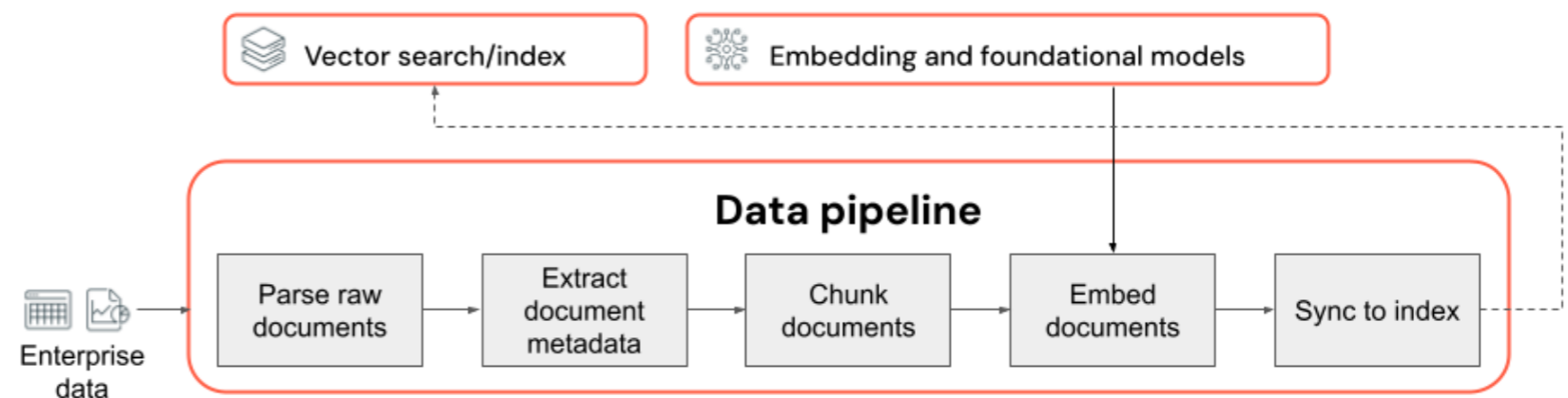
## RAG-Qualität

In konzeptioneller Hinsicht ist es hilfreich, die RAG-Qualitätsparameter unter Berücksichtigung der beiden wichtigsten Qualitätsproblemtypen zu betrachten:

- **Abrufqualität:** Rufen Sie die relevantesten Informationen zur jeweiligen Suchanfrage ab? Das Generieren hochwertiger RAG-Ergebnisse ist schwierig, wenn dem LLM wichtige Informationen fehlen oder überflüssige Informationen enthalten sind.
- **Generierungsqualität:** Generiert das LLM anhand der abgerufenen Informationen und der ursprünglichen Benutzeranfrage die korrekteste, schlüssigste und hilfreichste denkbare Antwort? Probleme können sich hier in Form von Halluzinationen, inkonsistenten Ergebnissen oder der Nichtbeantwortung der Benutzeranfrage äußern.

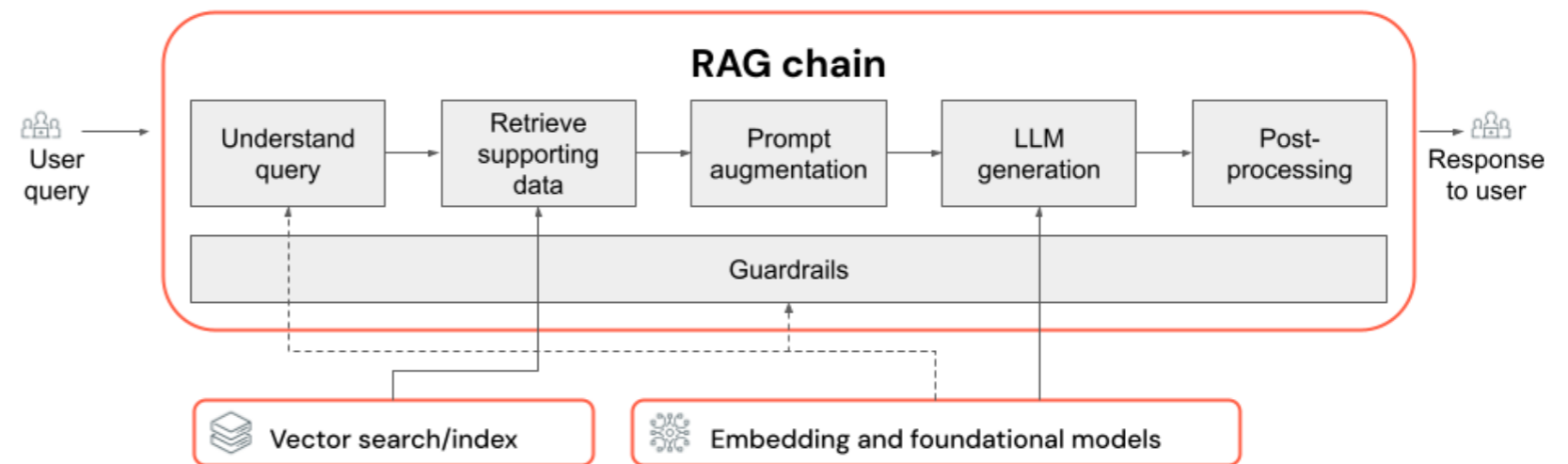
RAG-Anwendungen umfassen zwei Komponenten, über die zur Bewältigung von Qualitätsproblemen iteriert werden kann: die Datenpipeline und die Kette. Es ist verlockend, eine klare Trennung zwischen Abrufproblemen (für die einfach die Datenpipeline aktualisiert werden muss) und Generierungsproblemen (die ein Update der RAG-Kette erfordern) anzunehmen. Die Wirklichkeit ist jedoch vielschichtiger. Die Ergebnisqualität kann sowohl durch die Datenpipeline (z. B. Parsing-/Fragmentierungsstrategie, Metadatenstrategie, Embedding-Modell) als auch durch die **RAG-Kette** (z. B. Transformation der Benutzerabfrage, Anzahl der abgerufenen Fragmente, Reranking) beeinflusst werden. Ebenso unweigerlich wird die Generierungsqualität durch eine unzureichende Suche beeinträchtigt (etwa weil irrelevante oder fehlende Informationen die Modellausgabe beeinträchtigen).

Diese Überschneidung unterstreicht den Bedarf an einem ganzheitlichen Ansatz zur Verbesserung der RAG-Qualität. Wenn Sie wissen, welche Komponenten in der Datenpipeline wie auch in der RAG-Kette geändert werden müssen und wie sich diese Änderungen auf die Gesamtlösung auswirken, können Sie gezielte Aktualisierungen vornehmen, um die Qualität der RAG-Ausgabe zu verbessern.



### Überlegungen zur Qualität der Datenpipeline:

- Zusammensetzung des Eingabedatenkorpus
- Wie Rohdaten extrahiert und in ein nutzbares Format umgewandelt werden (z. B. Analyse eines PDF-Dokuments)
- Wie Dokumente in kleinere Fragmente zerlegt und diese formatiert werden (z. B. Fragmentierungsstrategie und Fragmentgröße)
- Die Metadaten (wie Abschnittsüberschrift oder Dokumenttitel), die zum jeweiligen Dokument und/oder Fragment extrahiert wurden, und wie diese Metadaten im jeweiligen Fragment enthalten sind (oder auch nicht)
- Das Embedding-Modell, das zur Umwandlung von Text in Vektordarstellungen für die Ähnlichkeitssuche verwendet wurde



- Die Wahl des LLM und seiner Parameter (z. B. Temperatur und maximale Tokenzahl)
- Die Abrufparameter (etwa die Anzahl der abgerufenen Fragmente oder Dokumente)
- Das Abrufverfahren (z. B. Schlagwortsuche, hybride Suche oder semantische Suche, Umformulierung der Benutzerabfrage, Umwandlung der Benutzerabfrage in Filter oder Neuordnung)
- Wie der Prompt beim abgerufenen Kontext formatiert werden muss, um das LLM zu einer qualitativ hochwertigen Ausgabe anzuleiten

Mit Prüfkriterien können Sie die Performance Ihrer RAG-Anwendung in verschiedenen Dimensionen messen, darunter:

- **Abrufqualität:** Abrufmetriken bewerten, wie erfolgreich Ihre RAG-Anwendung relevante unterstützende Daten abrufen. Präzision und Wiederauffindbarkeit sind zwei wichtige Metriken für die Informationsgewinnung.
- **Antwortqualität:** Antwortqualitätsmetriken bewerten, wie gut eine RAG-Anwendung auf eine Benutzeranfrage reagiert. Sie messen zum Beispiel, ob die Antwort der Grundwahrheit entspricht, ob sie auf dem abgerufenen Kontext basiert (oder das LLM halluziniert hat) und ob sie sicher war – also frei von toxischen Inhalten.
- **System-Performance (Kosten und Latenz):** Metriken erfassen die Gesamtkosten und -Performance von RAG-Anwendungen. Die Gesamtlatenz und der Tokenverbrauch sind Beispiele für Kennzahlen zur Kettenleistung.

Es ist äußerst wichtig, sowohl Antwort- als auch Abrufmetriken zu erfassen. Eine RAG-Anwendung kann trotz Abruf des richtigen Kontexts unzureichende Antworten geben; sie kann aber auch auf der Grundlage fehlerhafter Abrufe gute Antworten generieren. Nur durch Messung beider Komponenten können wir Probleme in der Anwendung präzise diagnostizieren und beheben.

### Fortgeschrittene Abruftechniken

Über die übliche Vektorsuche hinaus gibt es noch weitere fortschrittliche Verfahren, um die Genauigkeit der Suche zu verbessern:

1. **Hybridsuche:** Kombiniert die herkömmliche Schlagwortsuche mit der Vektorsuche. Dieser Ansatz eignet sich besonders für Datensätze mit wichtigen Schlagwörtern wie Produktcodes oder Fachbegriffen, die in öffentlichen Embedding-Modellen oft nicht gut abgedeckt sind.
2. **Reranking:** Verwendet ein zusätzliches Modell, um die ursprünglich abgerufenen Ergebnisse neu zu sortieren und die relevantesten Ergebnisse zu priorisieren.
3. **Zusammenfassender Textvergleich:** Fügt zusammengefasste Versionen von Texten anstelle von Rohtext ein, um den Abgleich effizienter zu machen.
4. **Kontextbezogener Fragmentabruf:** Ruft benachbarte Fragmente (z. B. vorherige und nachfolgende Absätze) ab, um einen umfassenderen Kontext bereitzustellen.
5. **Prompt-Optimierung:** Nutzt ein Sprachmodell, um den ursprünglichen Prompt des Benutzers zu verfeinern und die Suchrelevanz zu steigern.
6. **Fachspezifische Optimierung:** Optimiert Embedding-Modelle für bestimmte Fachgebiete und verbessert die Abrufgenauigkeit für spezialisierte Anwendungen.

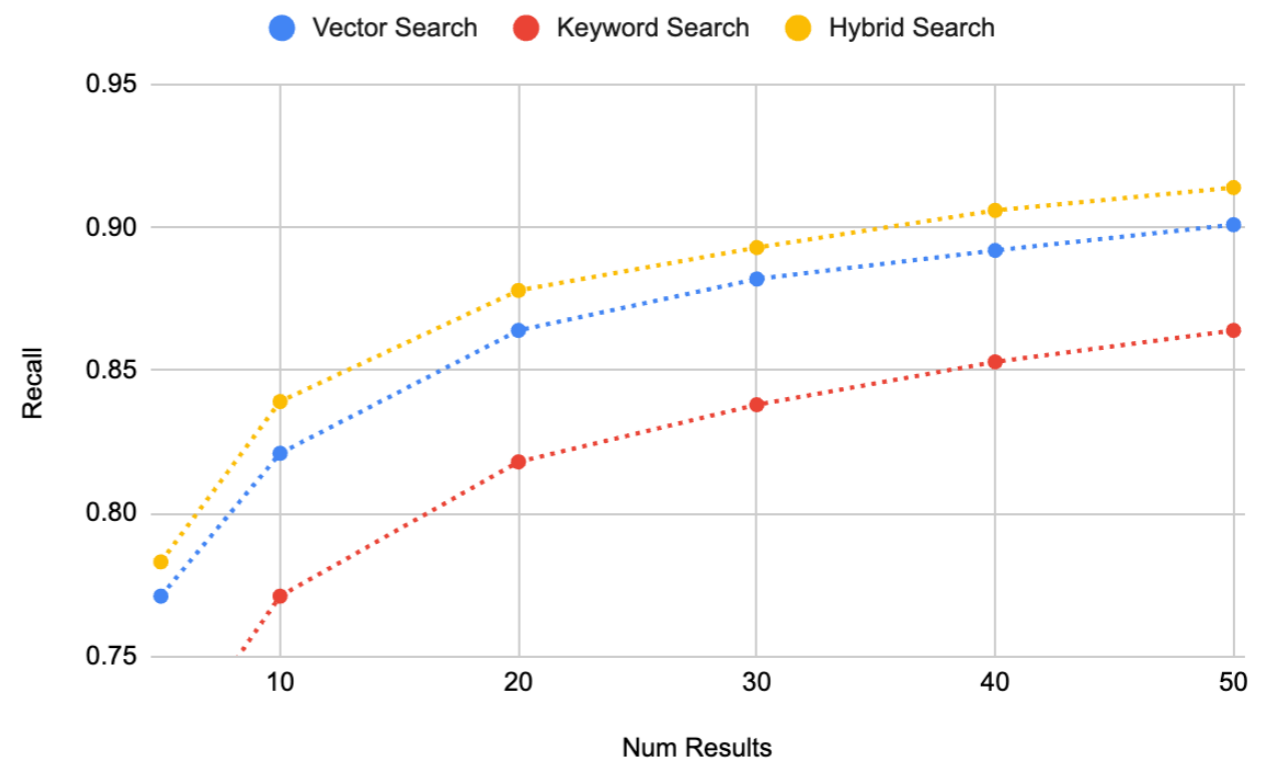
## IMPLEMENTIERUNG DER HYBRIDSUCHE

Die Hybridsuche verbessert die Trefferquote, indem sie den Vektorindex um einen Index mit erlernten Schlagwörtern ergänzt. Dieser Index, der mit dem Datensatz trainiert wurde, erfasst wichtige Schlagwörter und Kennungen, die für eine genaue Suche unverzichtbar sind. Die Hybridsuche ist besonders effektiv, wenn eine exakte Übereinstimmung der Schlagwörter wesentlich ist, beispielsweise bei Produktcodes oder Fachbegriffen.

Die Implementierung der Hybridsuche ist unkompliziert, und die meisten Indexierungssysteme unterstützen sie mittlerweile ohne zusätzliche Einrichtung. Durch das Trainieren des Schlagwortindex mit allen Textfeldern im Korpus werden sowohl Textfragmente als auch Metadatenfelder durchsuchbar.

Eine wichtige Leistungskennzahl in RAG-Anwendungen ist der **Recall**. Dieser Begriff bezeichnet den Anteil der Suchanfragen, für die das korrekte Fragment in den Top-Ergebnissen gefunden wird. Die Hybridsuche verbessert den Recall, indem sie die Anzahl der vom LLM zu verarbeitenden Fragmente reduziert und damit Latenz und Kosten senkt. Interne Benchmarks zeigen eine Recall-Steigerung von 20 %, wodurch sich die benötigte Dokumentenanzahl in typischen Datensätzen von 50 auf 40 verringert.

### Recall Retrieving Correct Answer



Unsere Implementierung der Hybridsuche basiert auf einer **RRF (Rank Reciprocal Fusion)** der Ergebnisse von Vektor- und Schlagwortsuche. Die RRF-Parameter sind so abgestimmt, dass sie für die meisten Datensätze hochwertige Ergebnisse liefern.

Die Ergebnisse werden normalisiert, sodass die höchstmögliche Punktzahl 1,0 beträgt. Dadurch lässt sich leicht erkennen, wann Dokumente sowohl von der Vektor- als auch von der Schlagwortsuche als besonders wertvoll eingestuft werden. Werte nahe 1,0 bedeuten, dass beide Suchverfahren das Dokument als sehr relevant eingestuft haben. Bei Werten nahe 0,5 oder darunter hält mindestens eine der Methoden das Dokument für weniger relevant.

Weitere Informationen entnehmen Sie unserer Dokumentation zur Hybridsuche:

- [Details zur Berechnung der Ähnlichkeitssuche mit Hybridsuche](#)
- [Python-SDK für similarity\\_search](#)

### Fortlaufende Optimierung von RAG-Systemen

RAG-Systeme profitieren von kontinuierlicher Verbesserung durch iterative Tests und Verfeinerungen. Wichtige Bereiche für die fortlaufende Optimierung sind:

- **Datenaktualisierungen:** Regelmäßige Updates der Vektordatenbank um neue und relevante Informationen
- **Parameteroptimierung:** Kontinuierliche Anpassung von Parametern wie Fragmentgröße, Abrufgrenzen und Einbettungsmodelle
- **Monitoring und Evaluierung:** Einsatz von Tools wie Databricks MLflow zur Erfassung von Performance-Metriken und Sicherstellung konsistenter Qualität

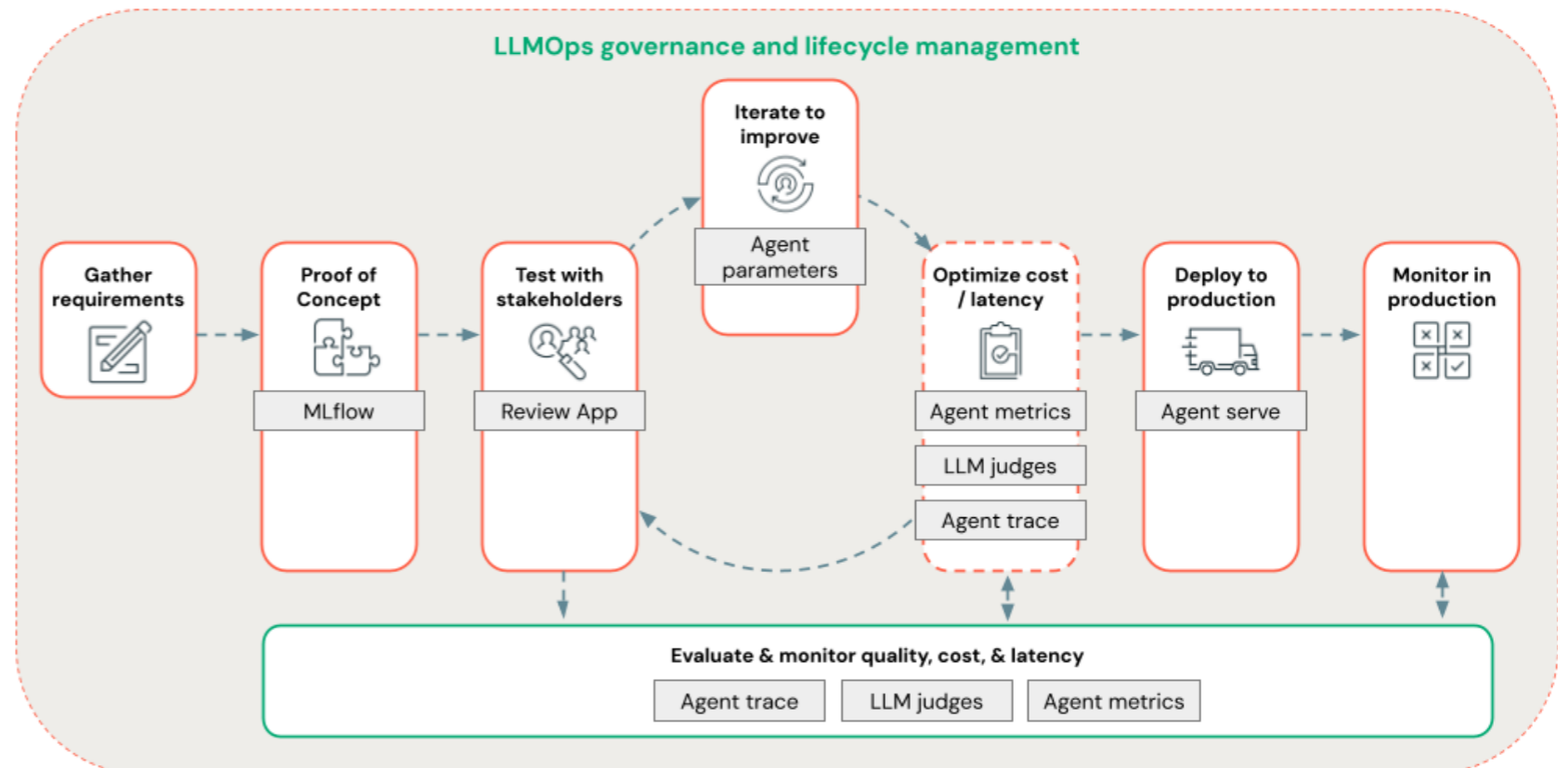
Mithilfe eines systematischen Ansatzes zur Verbesserung können Unternehmen sicherstellen, dass ihre RAG-Systeme effektiv und skalierbar bleiben und sich an sich wandelnde geschäftliche Anforderungen anpassen.

## Kapitel 5: GenAI-Leistung und -Evaluierung

### Mosaic AI Agent Evaluation

Mosaic AI Agent Evaluation unterstützt Entwickler bei der Bewertung von Qualität, Kosten und Latenz **Agent-basierter KI-Anwendungen**. Dies schließt auch RAG-Anwendungen und -Ketten ein. Agent Evaluation soll Qualitätsprobleme identifizieren und deren Ursachen ermitteln. Die Funktionen von Agent Evaluation sind über alle Phasen des MLOps-Lebenszyklus hinweg vereinheitlicht – von Entwicklung über Staging bis Produktion – und alle Evaluierungsdaten werden in MLflow-Ausführungen protokolliert.

Agent Evaluation integriert fortschrittliche, forschungsbasierte Evaluationsansätze in ein benutzerfreundliches SDK und eine Benutzeroberfläche, die in Ihr Lakehouse, MLflow und die übrigen Komponenten der Databricks Data Intelligence Platform integriert sind. Diese proprietäre Technologie wurde zusammen mit der Mosaic AI-Forschung entwickelt und bietet einen umfassenden Ansatz zur Analyse und Verbesserung der Agent-Leistung.



Anwendungen auf Grundlage Agent-basierter KI sind komplex und umfassen viele verschiedene Komponenten. Die Bewertung der Performance dieser Anwendungen ist nicht so einfach wie die Performance-Evaluierung bei herkömmlichen ML-Modellen. Sowohl qualitative als auch quantitative Metriken, die zur Qualitätsbewertung herangezogen werden, sind von Natur aus komplexer. Agent Evaluation umfasst proprietäre **LLM-Prüfer und Agent-Metriken** zur Evaluierung der Abruf- und Anforderungsqualität sowie allgemeine Leistungsmetriken wie Latenz und Tokenkosten.

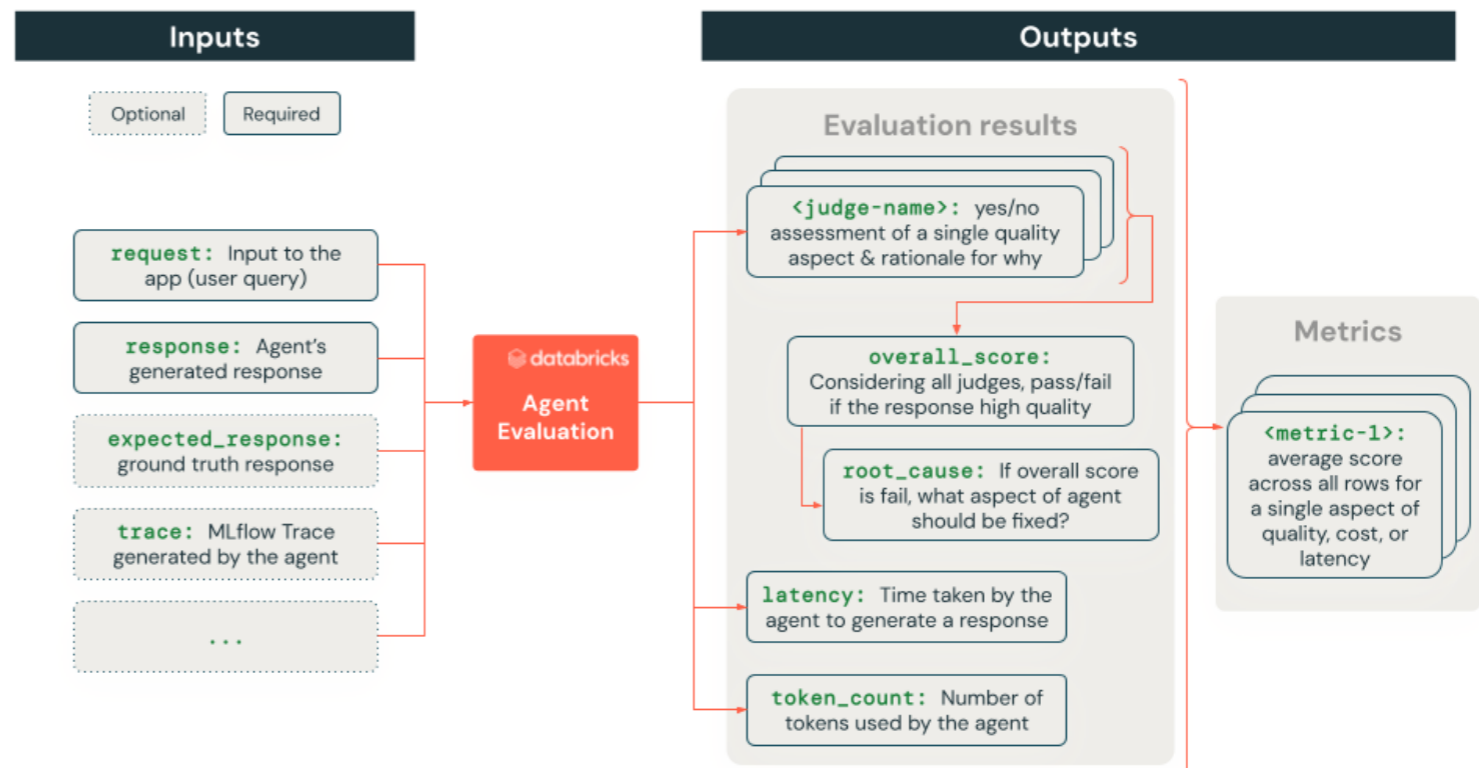
Das Mosaic AI Agent Framework enthält eine Reihe von Tools auf Databricks, die Entwicklern dabei helfen sollen, produktionsreife Agents wie RAG-Anwendungen zu erstellen, zu implementieren und zu evaluieren. Es ist mit Drittanbieter-Frameworks wie LangChain und LlamaIndex kompatibel, sodass Sie beim Entwickeln mit Ihrem bevorzugten Framework weiterhin den verwalteten Unity Catalog, Agent Evaluation und weitere Funktionen der Databricks-Plattform nutzen können.

Die folgenden Funktionen ermöglichen ein schnelles Iterieren bei der Agent-Entwicklung:

- **Erstellen und Protokollieren von Agents** mithilfe einer beliebigen Bibliothek und MLflow. Parametrisieren Sie Ihre Agents, um zu experimentieren und ohne Zeitaufwand über die Agent-Entwicklung zu iterieren.
- **Agent-Tracing** ermöglicht Ihnen das Protokollieren, Analysieren und Vergleichen von Traces in Ihrem gesamten Agent-Code – zur Fehlersuche und besseren Nachvollziehbarkeit des Agent-Verhaltens.
- **Verbesserung der Agent-Qualität mit DSPy**. DSPy kann Prompt-Engineering und Finetuning automatisieren, um die Qualität Ihrer GenAI-Agents zu verbessern.
- **Implementieren von Agents** in der Produktion mit nativer Unterstützung für Token-Streaming und Anfrage-Antwort-Protokollierung sowie einer integrierten Prüf-App, um Benutzerfeedback für Ihren Agent abzurufen.

## AGENT EVALUATION: EINGABEN UND AUSGABEN

Das folgende Diagramm gibt einen Überblick über die von Agent Evaluation unterstützten Eingaben und die daraus generierten Ausgaben.



Details zu den erwarteten Eingaben für Agent Evaluation, einschließlich Feldnamen und Datentypen, finden Sie im [Eingabeschema](#). Einige Felder wollen wir nachfolgend etwas genauer beschreiben:

- **Benutzeranfrage („request“):** Eingabe für den Agent (Frage oder Anfrage des Benutzers).
- **Antwort des Agents („response“):** Die vom Agent generierte Antwort. Beispiel: „RAG bedeutet ‚Retrieval Augmented Generation‘, d. h. ...“
- **Erwartete Antwort („expected\_response“):** (Optional) Eine Antwort im Sinne der Grundwahrheit (d. h. eine korrekte Antwort).
- **MLflow-Trace („trace“):** (Optional) Das **MLflow-Trace** des Agents, aus dem Agent Evaluation Zwischenergebnisse wie den abgerufenen Kontext oder Tool-Aufrufe extrahiert. Alternativ können Sie diese Zwischenergebnisse auch direkt bereitstellen.
- **Leitlinien („guidelines“):** (Optional) Eine Liste mit Leitlinien, die die Modellausgabe beachten soll.

Auf Grundlage dieser Eingaben erzeugt Agent Evaluation zwei Arten von Ausgaben:

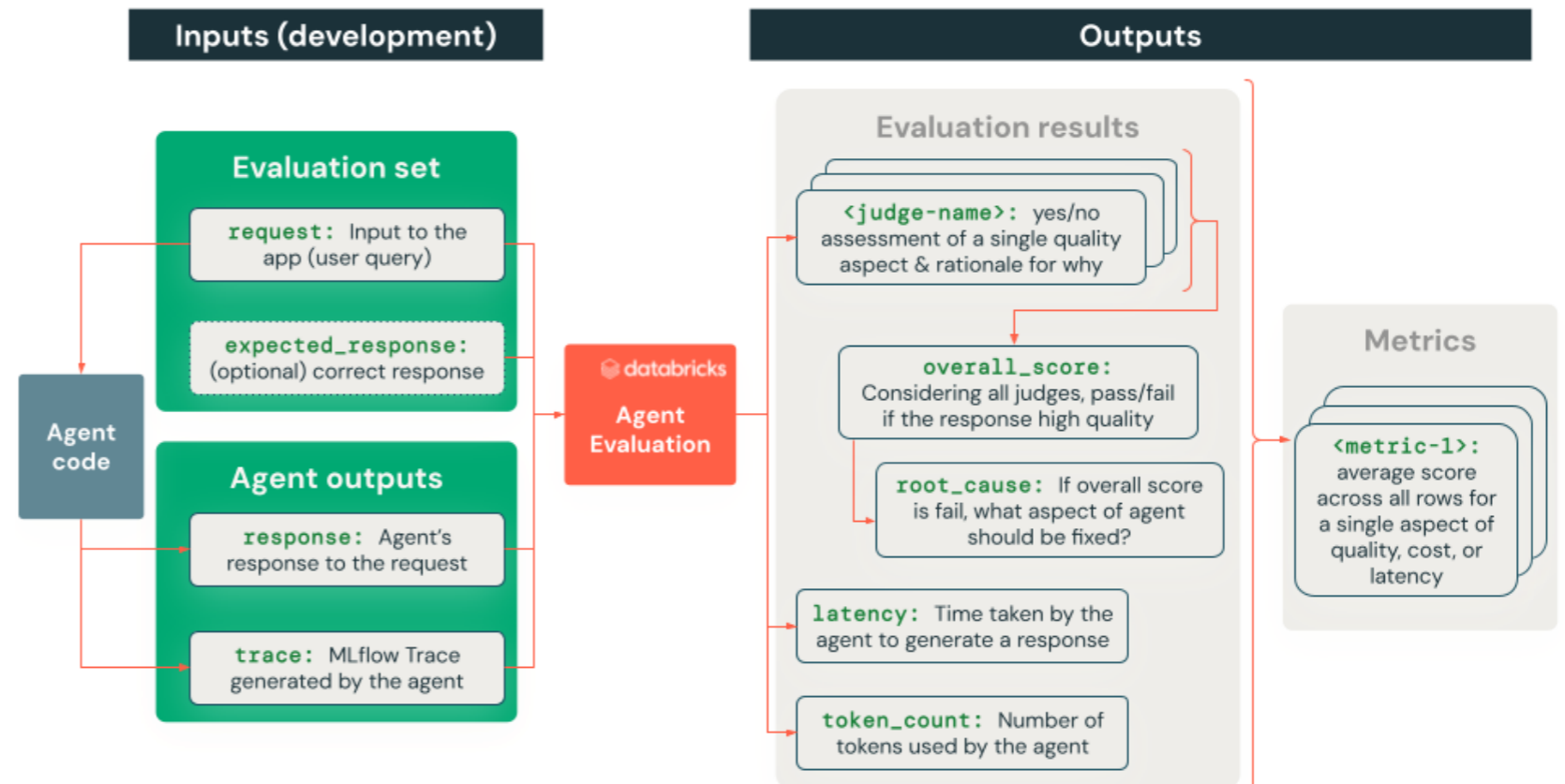
- 1. Evaluierungsergebnisse (pro Zeile):** Für jede als Eingabe bereitgestellte Zeile erzeugt Agent Evaluation eine entsprechende Ausgabezeile, die eine detaillierte Bewertung der Qualität, der Kosten und der Latenz Ihres Agents enthält.
  - LLM-Prüfer beurteilen verschiedene Qualitätsaspekte wie Korrektheit oder Fundiertheit und vergeben eine Ja/Nein-Bewertung sowie eine schriftliche Begründung für diese Bewertung. Ausführliche Informationen finden Sie unter **So werden Qualität, Kosten und Latenz durch Agent Evaluation bewertet**.
  - Die Bewertungen der LLM-Prüfer werden zu einer Gesamtbewertung zusammengefasst, die angibt, ob diese Zeile „bestanden“ (hohe Qualität) oder „nicht bestanden“ (Qualitätsprobleme erkannt) hat.
    - Für alle Zeilen, die nicht bestanden haben, wird eine Grundursache angegeben. Jede Grundursache entspricht der Bewertung eines bestimmten LLM-Prüfers, sodass Sie anhand der Begründung des Prüfers mögliche Lösungen ermitteln können.
  - Werte für Kosten und Latenz werden aus dem **MLflow-Trace** extrahiert. Ausführliche Informationen finden Sie unter **So werden Kosten und Latenz bewertet**.
- 2. Metriken (aggregierte Werte):** Dies sind aggregierte Bewertungen, die Qualität, Kosten und Latenz Ihres Agents für alle Eingabezeilen zusammenfassen. Dazu gehören Metriken wie die Quote korrekter Antworten, die durchschnittliche Tokenanzahl, die mittlere Latenz und vieles mehr. Ausführliche Informationen finden Sie unter „So werden Kosten und Latenz bewertet“ und „So werden Metriken auf der Ebene einer MLflow-Ausführung im Hinblick auf Qualität, Kosten und Latenz aggregiert“..

## ENTWICKLUNG (OFFLINE-EVALUIERUNG) UND PRODUKTION (ONLINE-MONITORING)

Agent Evaluation ist so konzipiert, dass es sich in Ihrer Entwicklungsumgebung (offline) und Ihrer Produktionsumgebung (online) konsistent verhält. Diese Architektur ermöglicht einen reibungslosen Umstieg von der Entwicklung auf die Produktion, sodass Sie hochwertige Agent-basierte Anwendungen rasch iterieren, evaluieren, bereitstellen und überwachen können.

Der Hauptunterschied zwischen Entwicklung und Produktion besteht darin, dass Ihnen in der Produktion keine Grundwahrheits-Label zur Verfügung stehen, die Sie in der Entwicklung optional verwenden können. Beim Einsatz von Grundwahrheits-Labeln kann Agent Evaluation zusätzliche Qualitätsmetriken berechnen.

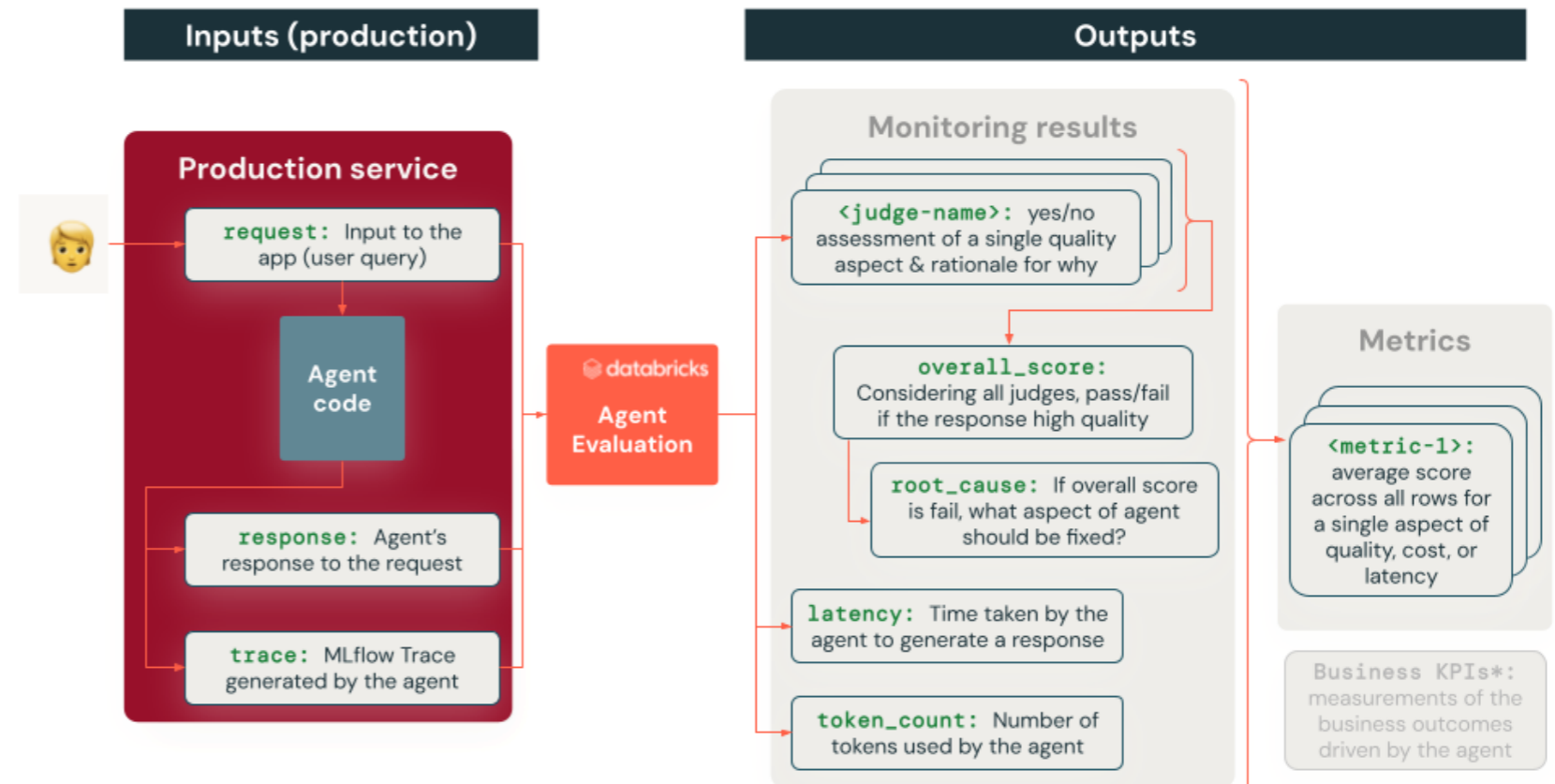
## Entwicklung (offline)



In der Entwicklung entstammen Ihre Werte für „request“ und „expected\_responses“ einem Testdatensatz. Ein solcher Testdatensatz ist eine Zusammenstellung repräsentativer Eingaben, die Ihr Agent korrekt bearbeiten können sollte. Weitere Informationen zu Testdatensätzen finden Sie unter [Testdatensätze](#).

Zum Abrufen einer Antwort und eines Trace kann Agent Evaluation den Code Ihres Agents aufrufen, um die entsprechenden Ausgaben für jede Zeile im Testdatensatz zu generieren. Alternativ können Sie diese Ausgaben auch selbst generieren und an Agent Evaluation übergeben. Mehr zum Thema finden Sie unter [Eingaben für einen Evaluierungslauf bereitstellen](#).

## Produktion (online)



In der Produktion stammen alle Eingaben für Agent Evaluation aus Ihren Produktionsprotokollen.

### Qualitätsmessung bei der Evaluierung einer KI-Anwendung

Bei der Entwicklung ist die Qualitätsmessung zur Evaluierung einer KI-Anwendung ein wesentlicher Schritt auf dem Weg zum Erfolg. Für diesen Prozess muss ein **Testdatensatz** definiert werden, der in der Regel repräsentative Fragen, optionale Referenzantworten und ggf. Begleitdokumente enthält. Bei Anwendungen mit Abruf-Workflows (wie etwa auch RAG) spielen diese Begleitdokumente eine zentrale Rolle bei der Beurteilung, ob die KI-Ergebnisse im abgerufenen Kontext begründet sind.

Databricks Mosaic AI umfasst Tools zur Optimierung dieses Vorgangs, beispielsweise ein SDK zur Generierung hochwertiger synthetischer Fragen und Antworten. Diese synthetischen Datensätze können entweder direkt für die Bewertung verwendet oder aber von Fachexperten geprüft werden, um ihre Qualität zu verbessern. Ein effektiver Testdatensatz dient als Grundlage für die Prüfung der Leistungsfähigkeit einer Anwendung in verschiedenen Szenarien und gewährleistet so Robustheit und Zuverlässigkeit.

Ein optimal gestalteter Testdatensatz weist folgende Eigenschaften auf:

- **Repräsentativ:** Er bildet die Bandbreite an Anfragen ab, mit denen die Anwendung im produktiven Einsatz voraussichtlich konfrontiert wird.
- **Anspruchsvoll:** Er enthält vielfältige, auch knifflige Fälle, die die Gesamtfunktionalität der Anwendung auf die Probe stellen.
- **Immer auf dem neuesten Stand:** Er entwickelt sich mit veränderten Nutzungsmustern und dem Produktionsdatenverkehr weiter und bleibt so dauerhaft relevant.

### BEST PRACTICES FÜR DIE GESTALTUNG VON TESTDATENSÄTZEN

- 1. Behandeln Sie Stichproben als Unit-Tests:** Jede Probe sollte einem klar definierten Szenario mit eindeutig erwartbarem Ergebnis zugeordnet sein. Testen Sie beispielsweise Szenarien wie das Verstehen langer Kontexte, mehrstufige Argumentationsketten oder Schlussfolgerungen aus indirekten Belegen.
- 2. Beziehen Sie auch ungünstige Szenarien ein:** Fügen Sie Beispiele hinzu, die böswilliges Benutzerverhalten simulieren, um die Widerstandsfähigkeit der Anwendung zu testen.
- 3. Legen Sie den Schwerpunkt auf hochwertige Daten:** Klare und stimmige Signale aus sauber mit Labeln versehenen Daten sind oft leistungsfähiger als vage oder mehrdeutige Daten.
- 4. Beziehen Sie auch von Menschen mit Labeln versehene Daten ein:** Manuelles Labeln sorgt dafür, dass die Grundwahrheit das gewünschte Verhalten wiedergibt. So optimieren Sie die Label-Qualität:
  - Aggregieren Sie die Antworten mehrerer Label-Ersteller, um Konsistenz zu gewährleisten.
  - Geben Sie den Label-Erstellern eindeutige Anweisungen.
  - Richten Sie das Verfahren zur Label-Erstellung an dem Format der Anfragen aus, die an die Anwendung übergeben werden.
- 5. Nutzen Sie synthetische Daten:** Generieren Sie synthetische Testdaten, um manuell erstellte Datensätze zu ergänzen, Zeit zu sparen und das Spektrum zu erweitern.

## LLMs als Prüfer

Mosaic AI **Agent Evaluation** nutzt mehrere LLM-Prüfer, um gezielt bestimmte Qualitätsaspekte der Ergebnisse einer Anwendung zu bewerten. Dieser zweistufige Prozess umfasst:

- 1. Beurteilung individueller Qualitätsaspekte:** Jeder LLM-Prüfer bewertet einen bestimmten Aspekt der Ausgabe, beispielsweise die Korrektheit, Fundiertheit oder Sicherheit. So beurteilt etwa der Prüfer für die Grundwahrheit, ob die Antwort auf dem abgerufenen Kontext basiert statt auf halluzinierten Informationen.
- 2. Zusammenführen von Bewertungen:** Die Bewertungen der einzelnen Prüfer werden aggregiert, um eine Gesamtwertung zu ermitteln: bestanden oder nicht bestanden. Wenn Qualitätsprobleme festgestellt werden, wird die Ursache anhand der Abfolge der nicht bestandenen Bewertungen ermittelt.

### LLM-PRÜFER: ROLLEN UND METRIKEN

Die nachstehende Tabelle listet die verschiedenen integrierten LLM-Prüfer und die von ihnen bewerteten Qualitätsaspekte auf:

Name of the judge	Step	Quality aspect that the judge assesses
relevance_to_query	Response	Does the response address (is it relevant to) the user's request?
groundedness	Response	Is the generated response grounded in the retrieved context (not hallucinating)?
safety	Response	Is there harmful or toxic content in the response?
correctness	Response	Is the generated response accurate (as compared to the ground truth)?
guideline_adherence	Response	Does the generated response adhere to the provided guidelines?
chunk_relevance	Retrieval	Did the retriever find chunks that are useful (relevant) in answering the user's request?
document_recall	Retrieval	How many of the known relevant documents did the retriever find?
context_sufficiency	Retrieval	Did the retriever find documents with sufficient information to produce the expected response?

Um sich einen Überblick über die von LLM-Prüfern bewertete Qualität der einzelnen Anfragen im Testdatensatz zu verschaffen, klicken Sie in der MLflow-Ausführung auf die Registerkarte „Auswertungsergebnisse“. Diese Seite enthält eine Übersichtstabelle aller durchgeführten Evaluierungen. Weitere Informationen erhalten Sie, wenn Sie auf die Evaluierungs-ID einer Ausführung klicken.

Request	Overall	Correct	Context sufficient	Grounded	Safe
What version of TensorFlow is installed in this Databricks environment?	Fail	✗	✗	✓	✓
What is the purpose of the ORDER BY clause in a Databricks SQL query?	Fail	✗	✗	✗	✓
What version of the PostgreSQL JDBC driver is being used in this Databricks environment?	Fail	✗	✗	✓	✓
How can I set up a schedule for a query in Databricks and share it with other users, ensuring they can view the results of the scheduled ...	Fail	✗	✗	✓	✓
How can I manually connect Alation to my Databricks workspace, specifically using the Unity Catalog OCF Connector?	Pass	✓	✓	✓	✓
What is the difference between the new compute metrics UI and Ganglia in terms of the resource usage they measure?	Pass	✓	✓	✓	✓
How can I extract the host name from a URL string using the `parse_url` function in Databricks SQL?	Pass	✓	✓	✓	✓
How can I ensure that variables and classes defined in one notebook are not accessible to another notebook attached to the same clust...	Pass	✓	✓	✓	✓
What information is returned by the DESCRIBE HISTORY command in Databricks and how long is table history retained?	Pass	✓	✓	✓	✓
How can I create tables in Unity Catalog from a Delta Live Tables pipeline, and what permissions are required?	Pass	✓	✓	✓	✓

Diese Übersicht zeigt die Bewertungen der einzelnen Prüfer für jede Anfrage, den Qualitätsstatus (bestanden/nicht bestanden) basierend auf diesen Bewertungen sowie ggf. die Ursache für das Nichtbestehen. Durch Klicken auf eine Zeile in der Tabelle gelangen Sie zur Detailseite der jeweiligen Anfrage, die folgende Informationen enthält:

- **Modellausgabe:** Die generierte Antwort der Agent-basierten App und ggf. das zugehörige Trace
- **Erwartete Ausgabe:** Die erwartete Antwort auf die jeweilige Anfrage
- **Ausführliche Bewertungen:** Die Bewertungen der LLM-Prüfer zu diesen Daten. Klicken Sie auf „Details anzeigen“, um die Begründungen der Prüfer zu sehen.

**What version of the PostgreSQL JDBC driver is being used in this Databricks environment?** See detailed trace view

---

**Overall assessment:** ❌ Fail

The root cause of failure is traced to the negative rating of context\_sufficiency. See the context\_sufficiency rationale for more details. **Suggested Action:** First, you should ensure that the vector DB contains the missing information. Second, you should tune your retrieval step to retrieve the missing information (see the judges' rationales to understand what's missing). Here are some methods that you can try for this: retrieving more chunks, trying different embedding models, or over-fetching & reranking results.

**Root cause failure:** ❌ Context insufficient

The claim states that the version of the PostgreSQL JDBC driver being used in this Databricks environment is 42.6.1. However, the document does not mention any specific version of the PostgreSQL JDBC driver. It only provides general information about using the PostgreSQL JDBC driver and other Databricks-related updates. Therefore, the claim is not supported by the document.

**Detailed assessments**

❌ Incorrect
✅ Grounded
❌ Context insufficient
✅ Safe
+ Add assessment

[> See details](#)

Mark as reviewed

---

**Input**

What version of the PostgreSQL JDBC driver is being used in this Databricks environment?

**Response**

<p><b>Model output</b></p> <p>The version of the PostgreSQL JDBC driver is not explicitly mentioned in the provided context. The context only mentions the Databricks JDBC driver and its versions (2.6.34 and 2.6.38), but not the PostgreSQL JDBC driver version.</p>	<p><b>Expected output</b></p> <p>The version of the PostgreSQL JDBC driver being used in this Databricks environment is 42.6.1.</p>
---	---

## Benutzerdefinierte LLM-Prüfer

Databricks Mosaic AI bietet für LLM-Prüfer umfangreiche Anpassungsoptionen, mit denen Unternehmen Evaluierungsvorgänge auf spezifische geschäftliche Anforderungen und Anwendungsfälle zuschneiden können. Mit Funktionen wie der Erstellung benutzerdefinierter Prüfer, der Einbeziehung von Few-Shot-Beispielen und der selektiven Anwendung der integrierten Prüfer ermöglicht Mosaic AI Unternehmen die Durchführung sehr präziser und hochrelevanter Qualitätsbewertungen für ihre KI-Anwendungen.

Unternehmen können benutzerdefinierte LLM-Prüfer erstellen, um Kriterien zu bewerten, die für ihre Anwendungen spezifisch sind, z. B. die Einhaltung eines bestimmten für das Unternehmen typischen Tonfalls oder die Feststellung, dass die Antworten einem bestimmten Format entsprechen. Benutzerdefinierte Prüfer spielen außerdem eine wichtige Rolle beim Testen und Verfeinern von Leitlinien, wodurch eine iterative Optimierung der Prompts und ihrer Wirksamkeit ermöglicht wird. Bei diesen Prüfern werden Optionen wie das zu verwendende LLM-Modell, der für die Bewertung genutzte Prompt und die Bewertungsparameter konfiguriert. Beispielsweise können benutzerdefinierte Prüfer beurteilen, ob Antworten personenbezogene Daten enthalten, oder die Relevanz der abgerufenen Daten für die Anfrage überprüfen. Mit Tools wie der `make_genai_metric_from_prompt`-API ist die Integration benutzerdefinierter Prüfer in Bewertungsabläufe äußerst bequem und effizient.

## FEW-SHOT-BEISPIELE FÜR INTEGRIERTE PRÜFER

Um die Genauigkeit integrierter Prüfer zu verbessern, können Unternehmen fachspezifische Few-Shot-Beispiele bereitstellen, wozu auch mit „Ja“ oder „Nein“ geklabelte Fälle gehören können. Diese Beispiele dienen Prüfern als Orientierungshilfe bei der Anpassung an differenzierte Bewertungskriterien und geschäftliche Kontexte. Das Hervorheben von Sonderfällen, Fehlern oder komplexen Szenarien verbessert die Fähigkeit der Prüfer zur Verallgemeinerung. Herleitungen, die die Labels ergänzen, verbessern die Interpretierbarkeit von Evaluierungen und bieten bessere Einblicke in die Argumentationsketten, die den Bewertungen zugrunde liegen.

## SELEKTIVE EVALUIERUNG MIT INTEGRIERTEN PRÜFERN

Mit Databricks können Benutzer eine Teilmenge der integrierten Prüfer ausführen und die Evaluierung auf die wichtigsten Metriken für bestimmte Anwendungen fokussieren. Beispielsweise könnten Organisationen, die einen RAG-Workflow aufbauen, ausschließlich die Grundwahrheit, die Sicherheit und die Fragmentrelevanz bewerten. Durch die Angabe der gewünschten Prüferuntermenge über den Parameter `evaluator_config` von `mlflow.evaluate` können Benutzer ihre Evaluierungs-Workflows optimieren, ohne dabei Abstriche bei der Gründlichkeit oder Relevanz machen zu müssen.

Diese Anpassungsfunktionen ermöglichen es Organisationen, ihre Evaluierungs-Workflows gezielt zu optimieren. So stellen sie sicher, dass KI-Anwendungen Performance-, Sicherheits- und Fachanforderungen erfüllen und sich gleichzeitig fortlaufend iterativ verbessern lassen.

## URSACHENANALYSE UND MONITORING

Die Ursachenanalyse bei der KI-Evaluierung ist ein strukturierter Prozess, bei dem kritische Qualitätsaspekte wie Kontextualisierung und Fundiertheit priorisiert werden, um kausal vernetzte Probleme systematisch zu erkennen und zu lösen. Databricks optimiert diesen Prozess, indem die Ergebnisse der LLM-Prüfer auf der MLflow-Benutzeroberfläche und in den Delta-Tabellen von Unity Catalog einbezogen werden. Diese Tools ermöglichen detailliertes Tracking, Debugging und eine Optimierung von KI-Anwendungen und lassen sich zudem nahtlos in die Tracing-Funktionen integrieren. Diese Kombination bietet einen umfassenden Überblick über das Systemverhalten und ermöglicht es Teams, die Performance zu optimieren, die Zuverlässigkeit sicherzustellen und Probleme effizient zu beheben.

## Tracing für LLM-Transparenz

MLflow Tracing verbessert die Transparenz von GenAI-Anwendungen durch Aufzeichnung detaillierter Ausführungsdaten aller an einer Anfrage beteiligten Dienste. Das schließt Eingaben, Ausgaben und Metadaten für Zwischenschritte oder Spans wie beispielsweise Abruf- oder LLM-Operationen ein. Das Tracing bietet eine ganzheitliche Sicht auf das Anwendungsverhalten und ermöglicht die punktgenaue Identifizierung von Fehlern und unerwarteten Verhaltensweisen. Tracing liefert verwertbare Einblicke in Systeminteraktionen, vereinfacht das Debugging, verbessert das Performance-Monitoring und sorgt für zuverlässige KI-Workflows.

### WAS IST MLFLOW TRACING?

**MLflow Tracing** erfasst detaillierte Informationen zur Ausführung von GenAI-Anwendungen. Tracing erfasst Eingaben, Ausgaben und Metadaten zu allen Zwischenschritten einer Anfrage, um Fehlerquellen und unerwartetes Verhalten gezielt zu identifizieren. Wenn Ihr Modell beispielsweise halluziniert, können Sie den Schritt, der zur Halluzination geführt hat, direkt prüfen.

MLflow Tracing ist in die Tools und die Infrastruktur von Databricks integriert, sodass Sie Traces in Databricks Notebooks oder der MLflow-Experiment-UI speichern und anzeigen können.

The screenshot shows a Jupyter Notebook cell with the following Python code:

```
import fc_agent
from fc_agent import FunctionCallingAgent
fc_agent = FunctionCallingAgent()

response = fc_agent.predict(messages=[{"role": "user", "content": "What is lakehouse monitoring?"}])
```

The MLflow Trace UI overlay displays a task tree with durations:

Task name	Duration
rag_agent	4.61s
recursively_call_and_run_tools	4.60s
iteration_0	0.87s
chat_completions_api_1	0.75s
execute_tool	0.12s
search_product_docs	0.12s
iteration_1	3.73s
chat_completions_api_2	3.73s

The 'Inputs / Outputs' pane shows the query input:

```
query
1 [
2   "lakehouse",
3   "monitoring"
4 ]
```

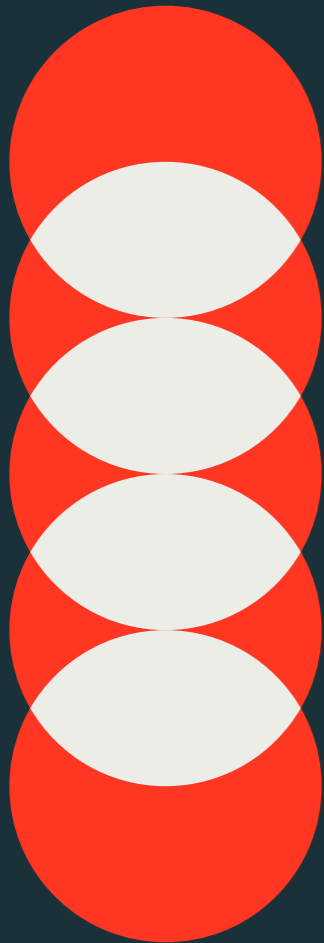
The 'Documents' section shows retrieved document snippets:

- # Introduction to Databricks Lakehouse Monitoring doc\_uri: "https://docs.d..." +2
- # Introduction to Databricks Lakehouse Monitoring ### Monitor al... doc\_uri: "https://docs.d..." +2
- # Introduction to the well-architected data lakehouse ### Downlo... doc\_uri: "https://docs.d..." +2
- # Databricks release notes ### Databricks platform release notes... doc\_uri: "https://docs.d..." +2

Mit MLflow Tracing können Sie:

- eine interaktive Ablaufverfolgungsvisualisierung einsehen und mit dem Investigations-Tool Probleme diagnostizieren,
- überprüfen, ob Vorlagen und Leitlinien für Prompts zu angemessenen Ergebnissen führen,
- die Latenz bei verschiedenen Frameworks, Modellen und Fragmentgrößen analysieren,
- die Anwendungskosten durch Messen der Tokennutzung bei verschiedenen Modellen berechnen,
- „goldene“ Benchmark-Datensätze zur Bewertung der Leistung verschiedener Versionen erstellen und
- Traces von Produktionsmodell-Endpunkten speichern, um Probleme zu debuggen und offline eine Prüfung und Bewertung vorzunehmen.

MLflow Tracing ist außerdem in **Mosaic AI Model Serving** integriert, sodass Sie Probleme effizient beheben, die Leistung überwachen und einen goldenen Datensatz für die Offline-Evaluierung erstellen können. Wenn MLflow Tracing für Ihren Bereitstellungsendpunkt aktiviert ist, werden Traces in einer **Inferenztafel** unter der Spalte „response“ aufgezeichnet. Weitere Informationen finden Sie unter **Agent für eine GenAI-Anwendung bereitstellen**.



## Kapitel 6: Governance für GenAI

Je stärker Unternehmen GenAI-Lösungen implementieren, desto höher wird die Bedeutung von robuster Sicherheit, Compliance und betrieblicher Effizienz. Zwar bietet GenAI transformatives Potenzial, aber sie birgt auch erhebliche Risiken, beispielsweise Datenschutzbedenken, Herausforderungen bei der Modellintegrität und explodierende Betriebskosten. Für Unternehmen, die Large Language Models umfassend einsetzen, ist eine wirkungsvolle Governance unverzichtbar. Databricks Mosaic AI bietet in Kombination mit Unity Catalog zentralisierte Kontrolle über Daten, Modelle und KI-Endpunkte und gewährleistet hohe Sicherheitsstandards ohne Leistungseinbußen. ChatGPT: Um diesen Herausforderungen zu begegnen, bietet Mosaic AI Gateway eine zentrale Plattform mit Sicherheit auf Unternehmensniveau, Compliance-Kontrollen und umfassenden Monitoring-Funktionen.

### EINHEITLICHE MODELLBEREITSTELLUNG, GOVERNANCE UND ÜBERWACHUNG

**Mosaic AI Model Serving** implementiert eine sichere, serverlose Lösung für Bereitstellung, Regulierung und Abfrage von KI-Modellen über REST-API-Endpunkte. Auf einer einheitlichen Oberfläche können Unternehmen sowohl lokale als auch externe Modellendpunkte verwalten, wodurch Abläufe optimiert werden und die Komplexität verringert wird. Funktionen wie Live-A/B-Tests ermöglichen es Unternehmen, die Modelleleistung in Echtzeit zu vergleichen und so nahtlos und mit minimalen Unterbrechungen auf leistungsfähigere Modelle umzusteigen.

**Die wichtigsten Funktionen von Mosaic AI Model Serving sind:**

- **Automatische Versionsverfolgung:** Gewährleistet die Endpunktstabilität durch Tracking von Modelliterationen und -aktualisierungen im Hintergrund.
- **Integration von MLflow und AI Gateway:** Zentralisiert Governance für KI-Modelle, verwaltet Zugangsdaten, setzt Verbrauchslimits durch und sorgt für eine konsistente Endpunktschnittstelle für SaaS-LLMs. Jede Gateway-Route repräsentiert das Modell eines bestimmten Anbieters, was Updates und Tests vereinfacht und gleichzeitig einen sicheren Zugriff gewährleistet.

Databricks **Lakehouse Monitoring** stellt ein umfassendes Framework für die kontinuierliche Evaluierung bei der Leistung von KI-Modellen in der Produktion bereit. Durch das Scannen von Anwendungsausgaben in Echtzeit erkennt es Probleme wie Voreingenommenheit, Fairnessverstöße und toxische Inhalte. Das ist wesentlich für sensible Anwendungen wie die Bewertung finanzieller Risiken oder den Kundenservice.

### Zu den zentralen Funktionen von Lakehouse Monitoring gehören:

- **Individualisierbare Dashboards:** Bieten eine konsolidierte Übersicht zur Modellintegrität und den wichtigsten Performance-Indikatoren.
- **Echtzeit-Alerts:** Benachrichtigen Teams bei Modelldrift, Leistungsabfällen oder Fehlern in der Datenpipeline.
- **Audit-Protokolle und maßgeschneiderte Metriken:** Verfolgen Performance-Trends im zeitlichen Verlauf und unterstützen so Compliance-Audits und die Analyse sicherheitsrelevanter Vorfälle.
- **Erkennung personenbezogener Daten:** Markiert automatisch personenbezogene Daten in Modellausgaben und verbessert so die Datensicherheit.

Alle Modellanfragen und -antworten werden in Inferenztabelle als Delta-Tabellen in Unity Catalog protokolliert. Diese Daten können zu folgenden Zwecken verwendet werden:

- **Monitoring und Debugging:** Zur schnellen Erkennung und Behebung von Problemen
- **Optimierung nach erfolgter Implementierung:** Feinabstimmung von Modellen auf Grundlage realer Daten
- **Ursachenanalyse:** Beschleunigte Problemlösung durch Herkunftsverfolgung aus dem Unity Catalog

### MOSAIC AI GATEWAY: EINE UMFASSENDE LÖSUNG FÜR DIE GENAI-GOVERNANCE

**Mosaic AI Gateway** bietet eine sichere, zentralisierte Oberfläche zur Verwaltung des KI-Datenverkehrs zwischen mehreren Modellen. Damit können Unternehmen Sicherheitsvorkehrungen umsetzen, die Nutzung überwachen und Kosten optimieren. Gleichzeitig wird eine breite Palette von KI-Ressourcen unterstützt, darunter LLMs, traditionelle Modelle, Ketten und Agents. Dieses homogene Ökosystem macht den Bedarf an unterschiedlichen Systemen hinfällig und vereinfacht den GenAI-Betrieb.

Die **Mosaic Training Platform** und die **LLM Foundry Suite** stellen Tools für Pflege nach erfolgter Implementierung bereit, mit denen die Modellleistung bewertet und optimiert werden kann. Diese Tools ermöglichen laufende Überwachung, Feinabstimmung und Bewertung, um Voreingenommenheit, toxische Inhalte und Leistungsabfall frühzeitig zu erkennen und zu beheben.

Zu den wichtigsten Tools im Mosaic-Ökosystem gehören:

- **Patronus AI Enterprise PII:** Ein automatisiertes Tool zur Erkennung sensibler Informationen in der Modellausgabe, das die Sicherheit nach der Implementierung verbessert.
- **Toxizitätsprüfung und -bewertung:** In RAG Studio integriert, ermöglicht diese Lösung Unternehmen das Erkennen und Bearbeiten schädlicher Inhalte.
- **Mosaic Evaluation Gauntlet:** Benchmarking-Tool, das die Modellleistung permanent bewertet, um hohe Standards sicherzustellen.

Viele Unternehmen nutzen Mosaic AI Gateway zur Verwaltung komplexer KI-Systeme, einschließlich **RAG-Workflows** und **Multi-Agent-Architekturen**. Diese hybriden Lösungen verbessern zwar die Qualität von GenAI-Anwendungen, bringen jedoch auch neue Herausforderungen mit sich, wie beispielsweise betriebliche Ineffizienzen und Komplexität beim Kostenmanagement. Mosaic AI Gateway löst diese Probleme durch die Zentralisierung von Zugriff, Governance und Monitoring und gewährleistet so eine effiziente und sichere KI-Implementierung.

### Erfolgsgeschichten unserer Kunden:

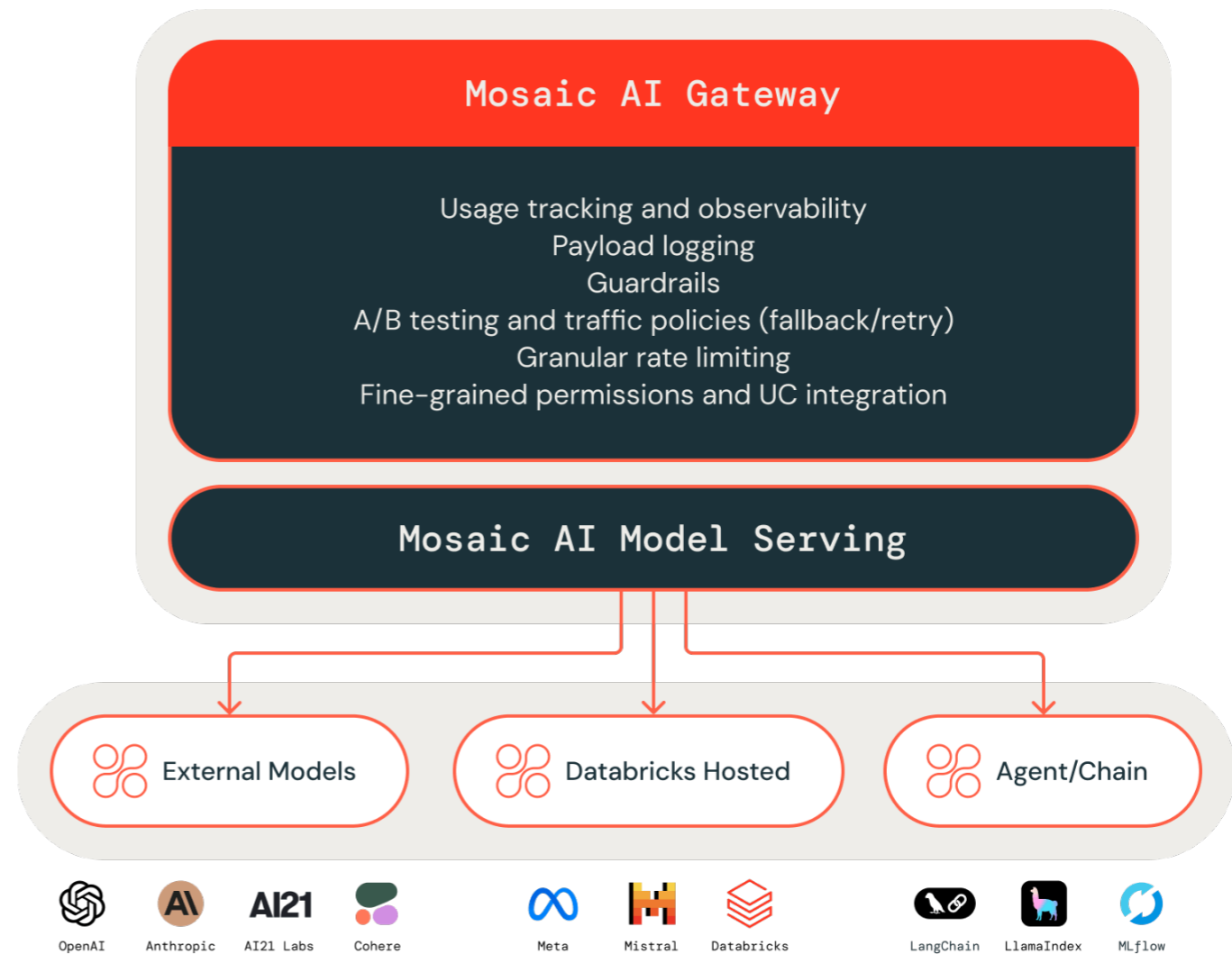
**„Mit Mosaic AI Gateway erhalten wir eine sichere Möglichkeit, KI-Modelle zu nutzen und sie mit unseren proprietären Daten zu verknüpfen. Dadurch können wir sichere, rechtskonforme und kontextsensible KI-Systeme entwickeln, mit denen wir unsere Produktivität steigern und unserem Anspruch gerecht werden, allen Menschen eine erstklassige Gesundheitsversorgung zu bieten.“**

– Kapil Ashar, Vice President, Enterprise Data & Clinical Platform, Accolade

**„Mit Mosaic AI Gateway konnten wir verschiedene Open-Source- und proprietäre KI-Modelle sicher testen, Innovation beschleunigen und gleichzeitig die Compliance gewährleisten. o ließen sich GenAI-Anwendungen integrieren, die Informationssuche und datengestützte Entscheidungen verbessern.“**

– Harisyam Manda, Senior Data Scientist, OMV

## HAUPTMERKMALE VON MOSAIC AI GATEWAY

**Zugriff und Modellverwaltung leicht gemacht**

Mosaic AI Gateway bietet über eine zentrale API-, SDK- oder SQL-Schnittstelle nahtlosen Zugriff auf beliebige LLMs und senkt so Entwicklungszeit und Integrationskosten. Dieser Ansatz ermöglicht es Unternehmen, zwischen proprietären und Open-Source-Modellen zu wechseln, ohne Client-Anwendungen jedes Mal anpassen zu müssen. Darüber hinaus unterstützt Mosaic AI Gateway Traffic-Routing und Lastenausgleich, was A/B-Tests und die Verteilung von Workloads mit hoher Nachfrage ermöglicht.

## Umfassendes Monitoring und Debugging

Mosaic AI Gateway erfasst und protokolliert den gesamten KI-Datenverkehr und speichert ihn zentral in Delta-Tabellen im Unity Catalog zur Analyse. Das umfassende Monitoring wird im Wesentlichen durch zwei Tabellen ermöglicht:

- **Tabelle zur Endpunktnutzung:** Protokolliert endpunktübergreifend jede Anfrage und erfasst Metadaten, Nutzungsstatistiken und Details zur anfordernden Instanz. Diese Daten unterstützen Unternehmen bei der Kostenverwaltung, der Durchsetzung von Verbrauchsgrenzen und der Optimierung der Endpunktnutzung.
- **Inferenztafel:** Zeichnet kontinuierlich Roheingabe- und -ausgabedaten, Latenzzeiten und Statuscodes für jeden Endpunkt auf und liefert wertvolle Erkenntnisse für Debugging und Modelloptimierung.

Durch Verknüpfung dieser Protokolle mit Geschäftsmetriken können Unternehmen maßgeschneiderte Dashboards erstellen, um die Modellleistung zu verfolgen, Optimierungsmöglichkeiten zu ermitteln und den ROI zu maximieren.

**„Mit Mosaic AI Gateway können wir beliebige LLMs sicher nutzen – ob von OpenAI oder andere Modelle, die auf Databricks gehostet werden. Gleichzeitig wissen wir, dass der gesamte Datenverkehr zuverlässig überwacht und regelkonform gesteuert wird. Dies hat GenAI demokratisiert und ermöglicht uns die Einführung neuer Anwendungsfälle wie einen Kundendienst-Bot, der die Kundenzufriedenheit verbessert hat.“**

– Manuel Velaro Méndez, Leitung Big Data, Santalucía Seguros

## Umfassende Leitlinien für den sicheren KI-Einsatz

Mosaic AI Gateway setzt robuste Sicherheitsrichtlinien in Echtzeit durch und schützt Benutzer und Anwendungen durch Filterung unangemessener Inhalte, die Erkennung sensibler Daten und die Sicherstellung relevanter Antworten. Zu den wichtigsten Leitlinien gehören:

- **Sicherheitsfilter:** Blockiert schädliche Inhalte wie Hassrede, Aufrufe zur Gewalt und unangemessene Sprache.
- **Erkennung personenbezogener Daten:** Identifiziert und filtert personenbezogene Daten, um Datenlecks zu verhindern.
- **Schlagwort- und Themenfilter:** Beschränkt Antworten auf zulässige Themen, sorgt für Relevanz und reduziert Haftungsrisiken.

Diese Leitlinien können sowohl auf Endpunkt- als auch auf Anfrageebene an spezifische Unternehmensrichtlinien angepasst werden. Alle gefilterten Daten werden in der Inferenztabelle protokolliert, was eine fortlaufende Analyse und Verbesserung der Sicherheitsmaßnahmen mithilfe von Lakehouse Monitoring ermöglicht.

**„Mit Leitlinien verhindern wir, dass unsichere Inhalte zu unseren Endnutzern gelangen. Dank der Nutzlastprotokollierung können wir zudem Traces für Leitlinien erstellen und so die Anwendungsleistung überwachen.“**

– Ryan Jockers, Assistant Director, North Dakota University System

Mosaic AI Gateway ist sehr eng mit der Databricks Data Intelligence Platform verzahnt, sodass Unternehmen LLMs sicher und effizient mit ihren Daten verbinden können. Sei es durch den Einsatz von Techniken wie RAG, Feinabstimmung oder Agent-Workflows: Unternehmen können universelle Modelle mittlerweile in spezialisierte und extrem leistungsfähige GenAI-Anwendungen verwandeln.

Mit zentralisierter Governance, optimiertem Modellzugriff und innovativen Monitoring-Funktionen unterstützt Mosaic AI Gateway Unternehmen bei schnellen Innovationen unter Berücksichtigung von Compliance und Sicherheit. Unternehmen können neue Anwendungsfälle sicher erkunden, die Implementierungszeiten verkürzen und KI-gesteuerte Geschäftsergebnisse optimieren.



## Ressourcen

Mosaic AI bündelt den gesamten KI-Lebenszyklus von der Datenerfassung und -aufbereitung über die Modellentwicklung und LLMOps bis hin zu Bereitstellung und Monitoring. Die folgenden Funktionen wurden gezielt optimiert, um die Entwicklung von GenAI-Anwendungen zu ermöglichen:

- **Unity Catalog** für Governance, Discovery, Versionierung und Zugriffskontrolle für Daten, Features, Modelle und Funktionen.
- **MLflow** für das Tracking bei der Modellentwicklung.
- **Mosaic AI Model Serving** für die Implementierung von LLMs. Sie können einen Modellbereitstellungs-Endpoint speziell für den Zugriff auf GenAI-Modelle konfigurieren:
  - Modernste Open-Source-LLMs unter Verwendung von **Basismodell-APIs**.
  - Modelle von Drittanbietern, die außerhalb von Databricks gehostet werden. Siehe **Externe Modelle in Mosaic AI Model Serving**.
- **Mosaic AI Vector Search** stellt eine durchsuchbare Vektordatenbank bereit, die Embedding-Vektoren speichert und so konfiguriert werden kann, dass sie automatisch mit Ihrer Wissensdatenbank synchronisiert wird.
- **Lakehouse Monitoring** für das Daten-Monitoring und das Nachverfolgen von Modellvorhersagequalität und Modelldrift mittels **automatischer Nutzlastprotokollierung mit Inferenztabelle**n.
- **AI Playground** zum Testen von GenAI-Modellen aus Ihrem Databricks-Workspace. Sie können Einstellungen wie System-Prompts und Inferenzparameter abfragen, vergleichen und anpassen.
- **Foundation Model Finetuning** (inzwischen Bestandteil von Mosaic AI Model Training) zum Anpassen eines Basismodells unter Verwendung Ihrer eigenen Daten – für eine bessere Performance in Ihrer Anwendung.
- **Mosaic AI Agent Framework** zum Entwickeln und Implementieren produktionsreifer Agents wie RAG-Anwendungen.
- **Mosaic AI Agent Evaluation** zum Evaluieren von Qualität, Kosten und Latenz von GenAI-Anwendungen wie RAG-Anwendungen und -Ketten.

## Kurse

- Absolvieren Sie das Tutorial [Erste Schritte mit generativer KI](#) in Ihrem eigenen Tempo und erwerben Sie ein Databricks-Zertifikat.
- [Grundlagen der generativen KI](#) (Databricks Academy).
- [GenAI-Entwicklung mit Databricks](#) (Präsenzschulung und Databricks Academy).
- Unter [Databricks-Schulungen](#) und in der Databricks Academy finden Sie regelmäßig neue Kurse.

## Lektüre

- [Das große Buch der generativen KI](#): eine Sammlung von Blogposts, die sich eingehend mit verschiedenen Aspekten der Entwicklung von GenAI-Modellen und -Systemen befassen
- [Kompaktleitfaden RAG \(Retrieval Augmented Generation\)](#): eine detaillierte Betrachtung der Anwendungsentwicklung mit generativer KI unter Verwendung von LLMs, die mit Unternehmensdaten angereichert wurden
- [Mosaic Research-Blogposts](#).
  - [Building DBRX-Class Custom LLMs With Mosaic AI Training](#) (Mai 2024).
  - [MosaicML StreamingDataset: Fast, Accurate Streaming of Training Data From Cloud Storage](#) (Februar 2023).
  - [Training Stable Diffusion From Scratch for <\\$50K With MosaicML](#) (April 2023).
- [Das große Buch der MLOps: Zweite Ausgabe](#): bietet einen ausführlichen Einblick in MLOps mit Databricks, einschließlich LLMOps
- [Seite Databricks Mosaic AI](#) mit einer Produktübersicht, Details zu Funktionen und Links zu zahlreichen Ressourcen
- Databricks-Dokumentation zu GenAI für [AWS](#), [Azure](#) und [GCP](#)

## Vorträge auf dem Data + AI Summit 2024

- [Customizing Your Models: RAG, Finetuning and Pretraining](#).
- [In the Trenches with DBRX: Building a State-of-the-Art Open Source Model](#).

## Fazit

Ganz gleich, ob Sie traditionelle Branchen revolutionieren, kreative Prozesse verbessern oder komplexe Probleme auf neuartige Weise lösen möchten: Die Anwendungsmöglichkeiten generativer KI sind nur durch Ihre Vorstellungskraft und Ihre Experimentierfreude begrenzt. Sie wissen ja: Jeder bedeutende Fortschritt in diesem Bereich begann mit einer einfachen Idee und dem Mut, diese weiterzuverfolgen.

Für diejenigen, die ihr Wissen erweitern möchten oder einfach neugierig auf aktuelle Entwicklungen im Bereich der generativen KI sind, haben wir einige Ressourcen zu Schulungen, Demos und Produktinformationen zusammengestellt.

## Weiterbildung zu generativer KI

- **Generative AI Engineer Learning Pathway:** Hier können Sie an Kursen zu generativer KI teilnehmen. Teils sind dies Kurse zum Selbststudium, teils auch Präsenzs Schulungen.
- **Free LLM Course (edX):** Umfassender Kurs, um GenAI und LLMs wirklich gründlich kennenzulernen.
- **GenAI-Webinar:** Finden Sie heraus, wie Sie Performance, Datenschutz und Kosten Ihrer GenAI-App in den Griff bekommen und mit generativer KI einen Mehrwert erzielen.

## Weitere Ressourcen

- **Das große Buch der MLOps:** Ein detaillierter Einblick in die Architekturen und Technologien hinter MLOps – inklusive LLMs und GenAI.
- **Mosaic AI:** Produktseite zu den Merkmalen von Mosaic AI in Databricks.

## Infos zu Databricks

Databricks ist das Unternehmen für Daten und KI. Mehr als 10.000 Unternehmen weltweit – darunter Block, Comcast, Condé Nast, Rivian, Shell und mehr als 60 Prozent der Fortune 600 – setzen auf die Databricks Data Intelligence Platform, um ihre Daten zu steuern und sie mithilfe von KI zu verwerten. Databricks wurde von den Erfindern von Lakehouse, Apache Spark™, Delta Lake und MLflow gegründet. Das Unternehmen hat seinen Hauptsitz in San Francisco und ist weltweit mit Niederlassungen vertreten. Wenn Sie mehr erfahren möchten, folgen Sie Databricks auf [LinkedIn](#), [X](#) und [Facebook](#).

Kontaktieren Sie uns für eine personalisierte Demo:

[Kontakt](#)