Innovation mit KI – Praxis, Projekte, Perspektiven

„Innovation mit KI – Praxis, Projekte, Perspektiven“ zeigt, wie künstliche Intelligenz heute konkret eingesetzt werden kann. Das Buch verbindet technische Grundlagen mit praktischen Beispielen aus Unternehmen und Organisationen. Ein besonderer Fokus liegt dabei auf Datenschutz und IT-Sicherheit, um verantwortungsvolle Innovation zu ermöglichen. Leserinnen und Leser erfahren, wie aus Ideen umsetzbare Projekte entstehen und welche Werkzeuge dabei unterstützen. Chancen und Risiken von KI werden realistisch eingeordnet, ergänzt durch praxisnahe Tipps und weiterführende Ressourcen auf **gerds-it.de**.

- KI auf dem Windows-Notebook – Chancen nutzen, Datenschutz wahren, Sicherheit stärken
- KI Modellfamilien - Übersicht (Stand April 2026) - Arbeitsdokument
- Übersicht gängiger KI-Plattformen (Stand April 2026) - Arbeitsdokument

KI auf dem Windows-Notebook – Chancen nutzen, Datenschutz wahren, Sicherheit stärken

Warum überhaupt KI lokal betreiben?

Viele Unternehmen und Organisationen wollen die Chancen von KI nutzen, ohne dabei Datenschutz, DSGVO und IT-Sicherheit zu gefährden. Cloud-Dienste sind bequem, bedeuten aber immer, dass sensible Daten das eigene Haus verlassen – ein Risiko, das gerade im geschäftlichen Umfeld schwer vertretbar ist. **Lokale Sprachmodelle** laufen dagegen direkt auf dem eigenen Notebook oder Server: Daten bleiben intern, Zugriffe sind kontrollierbar, und auch offline ist die Nutzung möglich.

Was wird dafür benötigt?

- **Hardware**: ein aktuelles Windows-Notebook oder ein Server mit mindestens 16 GB RAM/VRAM (für Modelle wie *gpt-oss-20b* ausreichend).
- **Software**: eine lokale KI-Plattform wie **Ollama** (Open Source, flexibel) oder **LM Studio** (GUI-basiert, einfach).
- **Modelle**: frei verfügbare Open-Weight-Modelle (z. B. *gpt-oss-20b*, *Phi-3*, *Mixtral*) je nach Anwendungsfall.
- **Organisation**: Updates oder RAG-Anbindung, um Modelle mit aktuellem Wissen zu versorgen.

□□ Damit entsteht eine **unabhängige und sichere KI-Umgebung**, die Innovation ermöglicht, ohne die Kontrolle über die eigenen Daten zu verlieren.

Lokale Sprachmodelle gibt es mittlerweile in vielen Varianten – von schlanken Community-Projekten bis hin zu professionell gepflegten Plattformen. Für die Praxis im Innovation Lab sind vor allem Lösungen interessant, die **datenschutzfreundlich, einfach nutzbar und breit verfügbar** sind. Unter den bekanntesten Tools stechen **Ollama** und **LM Studio** hervor, weil sie jeweils einen anderen Ansatz verfolgen: maximale Transparenz und Integration auf der einen Seite, besonders einfache Bedienung auf der anderen.
Um den Überblick zu vervollständigen, sind in der folgenden Tabelle auch weitere relevante Projekte wie **GPT4All**, **KoboldCpp**, **Text Generation WebUI** und **Jan AI** enthalten. Neben technischen Merkmalen zeigt die Übersicht auch, mit welcher Besonderheit sich die Anbieter selbst positionieren.

KI-Plattformen:

Merkmal	Ollama	LM Studio	GPT4All	KoboldCpp	Text Generation WebUI	Jan AI
Lizenzmodell	Open Source (MIT)	Proprietär, kostenlos, Enterprise-Pläne	Open Source (Apache 2.0)	Open Source	Open Source	Proprietär, kostenlos
Quellcode	Offen	Geschlossen	Offen	Offen	Offen	Geschlossen
Bedienung	CLI + API	GUI	GUI	CLI	Web-Oberfläche (umfangreich)	GUI
Plattformen	Linux, macOS, Windows	Windows, macOS, Linux (Beta)	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS, Linux	Windows, macOS
Datenschutz	Komplett lokal	Lokal	Lokal	Lokal	Lokal	Lokal

Stärken	Transparent, flexibel, integrationsfähig	Einfach, schnell, nutzerfreundlich	Viele Modelle, einfache Installation	Leichtgewichtig, ressourcenschonend	Sehr flexibel, viele Erweiterungen	Moderne Oberfläche, intuitive Nutzung
Schwächen	Einstieg erfordert technisches Know-how	Proprietär, weniger transparent	Weniger „polished“, Community-getrieben	Fokus auf Nischen (z. B. Rollenspiele)	Komplexe Einrichtung, eher für Enthusiasten	Noch geringe Verbreitung, unreifer
Zielgruppe	Entwickler, Integratoren	Einsteiger, Teams	Experimentierfreudige Anwender	Technikaffine mit wenig Ressourcen	Power-User, Bastler	Early Adopter
Besonderheit (Hersteller)	*„Privacy-first AI“ – volle lokale Kontrolle und einfache Modellintegration*	*„AI for everyone“ – lokale Nutzung so einfach wie ChatGPT in der Cloud*	*„Open ecosystem for local LLMs“ – Zugang zu vielen Modellen über eine App*	*„Lightweight & fast“ – KI auf nahezu jeder Hardware nutzbar*	*„Maximum flexibility“ – unzählige Erweiterungen und Schnittstellen*	*„Next-gen local AI“ – elegante, moderne Benutzeroberfläche für KI*
Webseite	ollama.com	lmstudio.ai	gpt4all.io	github.com/LostRuins/koboldcpp	github.com/oobabooga/text-generation-webui	jan.ai

---

Die Wahl des richtigen Sprachmodells ist entscheidend, um KI sinnvoll und sicher einzusetzen. Während manche Modelle als **Allrounder** überzeugen, sind andere auf **Effizienz** oder **Forschung** spezialisiert. Für das Innovation Lab sind vor allem Modelle relevant, die **lokal laufen**, um Datenschutz und IT-Sicherheit zu gewährleisten.

Die Landschaft der Sprachmodelle entwickelt sich rasant. Während LLaMA, Mistral oder Falcon wichtige Meilensteine waren, bestimmen heute vor allem neue **Open-Weight-Modelle** wie **gpt-oss-20b** von OpenAI und die **aktuellen Phi-3-Varianten** das Innovationsgeschehen. Diese Modelle sind nicht nur leistungsstark, sondern auch auf **lokale Nutzung** optimiert – ein entscheidender Vorteil für Datenschutz, DSGVO und IT-Sicherheit. Die folgende Tabelle stellt die **wichtigsten aktuellen Modelle** vor und ergänzt ältere Klassiker, die weiterhin in speziellen Szenarien relevant sein können.

Die unterschiedlichsten Modelle können auch im Innovations Lab getestet werden: Zukunftswerkstatt KI

Vergleichstabelle: Top-Innovationsmodelle (2025)Sprachmodelle 2025 – Herstellerfokus & Innovation

Modell	Besonderheit / Fokus laut Hersteller	Erstveröff.	Ollama	LM Studio	Datenschutz & IT-Sicherheit

gpt-oss-20b (OpenAI)	Erstes Open-Weight-MoE von OpenAI, Apache-2.0, optimiert für 16 GB RAM/VRAM	2025	✔	✔	Lokal, offen, DSGVO-konform
LLaMA 3.1 (Meta)	Größtes offenes Modell (bis 405 B), multilingual, Open-Weight, breite Community	2025	✔	✔	Lokal, Open Source, Lizenzbedingungen beachten
Qwen-3 (Alibaba)	Neueste Generation, Multilingualität & lange Kontexte (>128k Tokens), starke Benchmarks	2025	✔	✔	Lokal, Apache-2.0, DSGVO-konform
DeepSeek-R1-Distill-7B	Kompaktes Modell mit starkem Reasoning, ressourcenschonend, Notebook-freundlich	2025	✔	✔	Lokal, quelloffen, effizient
Phi-3 (Microsoft)	Effizienz auf geringster Hardware, Edge-tauglich, optimiert für Alltagseinsatz	2025	✔	✔	Lokal, sicher nutzbar
Mixtral 8x22B (Mistral)	High-End-MoE-Leistung, offene Gewichte, Spitzenmodell für Forschung	2024	✔	✔	Lokal, Open Source
Falcon (TII)	Forschungsmodell aus VAE, Open Source, früher Benchmarkführer	2023	✔	✔	Lokal, sicher, aber weniger innovativ
Qwen-2.5 (Alibaba)	Vorgänger von Qwen-3, stabil, weit verbreitet, gute Integration	2024	✔	✔	Lokal, Apache-2.0

GPT4All (Nomic)	Community-getrieben, einfache GUI-App, ideal für Einsteiger & Experimente	ab 2023	Teilweise	✔	Lokal, Qualität je nach Quelle

---

Sprachmodelle 2025 – Innovation im Überblick

Modell	Zweck / Empfehlung	Aktualität & Umgang mit neuen Themen
gpt-oss-20b (OpenAI, 2025)	Modernes Allround-Modell, optimiert für Notebooks (16 GB RAM/VRAM), produktive Arbeit mit DSGVO-Anspruch	Neu (2025), Trainingsstand sehr aktuell; Inhalte bis Anfang 2025, kein Live-Webzugriff
LLaMA 3.1 (Meta, 2025)	Offenes Spitzenmodell mit Community-Support, Allrounder für Text, Forschung & Prototyping	Trainingsstand 2024/25; große Community hält es durch Feintuning und Adaptionen aktuell
Qwen-3 (Alibaba, 2025)	Neueste Generation mit starkem Multilingual-Fokus und langen Kontexten (>128k Tokens); sehr leistungsfähig auch in Nicht-Englisch	Neu (2025), frisch trainiert; erste Benchmarks zeigen deutliche Sprünge gegenüber 2.5
DeepSeek-R1-Distill-7B (2025)	Kompakt & effizient, gutes Reasoning bei wenig Ressourcen, Notebook-freundlich	Neu (2025), stark auf aktuelles Reasoning optimiert; weniger Fokus auf Breitenwissen
Phi-3 (Microsoft, 2025)	Ressourcenschonend, ideal für schwächere Hardware oder Edge-Geräte	2024/25; Microsoft pflegt regelmäßige Updates, aber ohne Live-Web
Mixtral 8x22B (Mistral, 2024)	High-End-Modell für Forschung & komplexe Analysen, braucht starke GPU	Trainingsstand 2023/24; leistungsfähig, aber nicht tagesaktuell
Falcon (TII, 2023)	Forschung & Business-Texte, solider Klassiker	Stand 2023; heute weniger innovativ, aber für bestimmte Szenarien noch brauchbar
Qwen-2.5 (Alibaba, 2024)	Vorgänger von Qwen-3, sehr verbreitet, stabil und breit integriert (Ollama/LM Studio)	Trainingsstand 2024; weiterhin stabil nutzbar, aber nicht mehr top-aktuell
GPT4All-Modelle (Nomic, seit 2023)	Einfacher Einstieg, Experimente & Lernen mit GUI-App	Basieren meist auf älteren Modellen; Aktualität abhängig von Community-Updates

---

? Praxis: Wie halte ich lokale KI-Modelle aktuell?

Lokale Sprachmodelle haben ein **festes Wissensstand-Datum** – sie lernen nicht automatisch Neues dazu. Damit sie dennoch aktuell und nützlich bleiben, gibt es drei Wege:

1. **Neue Modellversion installieren**
   - Regelmäßig erscheinen Updates (z. B. *LLaMA-3*, *Phi-3*, *gpt-oss-20b*).
   - Diese müssen manuell heruntergeladen und eingerichtet werden.
2. **Feintuning oder Adapter nutzen**
   - Mit *LoRA*- oder *QLoRA*-Techniken lässt sich ein bestehendes Modell schnell auf eigene Daten (z. B. Firmenwissen) anpassen.
   - Vorteil: kostengünstig und gezielt.
3. **RAG (Retrieval Augmented Generation)**
   - Das Modell bleibt unverändert, erhält aber bei jeder Anfrage aktuelle Dokumente oder Daten aus einer Wissensdatenbank.
   - So können auch Nachrichten von heute verarbeitet werden, obwohl das Modell selbst sie nicht „kennt“.

Vergleich zur Cloud: Während Cloud-Modelle automatisch durch den Anbieter aktualisiert werden, bedeutet lokale Nutzung **mehr Eigenverantwortung** – dafür bleiben alle Daten **unter deiner Kontrolle** und DSGVO-konform.

KI Modellfamilien - Übersicht (Stand April 2026) - Arbeitsdokument

Übersicht gängiger KI-Modellfamilien

> Diese Übersicht beschreibt verbreitete KI-Modellfamilien, ihre Herkunft, typische Ausprägungen, die grundsätzliche lokale Nutzbarkeit, den inhaltlichen Fokus sowie eine grobe Einordnung ihrer Verbreitung. Sie dient der schnellen Orientierung und erhebt keinen Anspruch auf Vollständigkeit.

Grundsätzliche Hinweise

- Modellbezeichnungen sind **nicht einheitlich standardisiert**.
- Viele Namen bestehen aus einer Kombination aus:
  - **Modellfamilie**
  - **Versions- oder Generationsnummer**
  - **Größenangabe**
  - **Spezialisierung oder Einsatzschwerpunkt**
- Begriffe wie `mini`, `pro`, `thinking`, `reasoning`, `vision` oder `turbo` sind häufig **anbieterspezifisch**.
- Die lokale Nutzbarkeit hängt zusätzlich von Faktoren wie **Quantisierung**, **Kontextlänge**, **RAM**, **VRAM** und **CPU/GPU-Leistung** ab.

Tabelle: Gängige Modellfamilien

Modellfamilie	Herkunft / Anbieter	Typische Ausprägungen	Lokal nutzbar	Grobe lokale Anforderung	Verbreitung	Typischer Fokus
Llama	Meta	z. B. 1B, 3B, 8B, 70B	Ja	kleine bis mittlere Varianten gut lokal nutzbar; große Varianten eher für starke Systeme	sehr hoch	Allround, Chat, Assistenz, Basis für viele lokale Setups
Qwen	Alibaba / Qwen Team	z. B. 0.6B, 4B, 8B, 14B, 32B, MoE-Varianten	Ja	kleine und mittlere Varianten gut lokal; größere Varianten deutlich anspruchsvoller	sehr hoch	multilingual, Coding, Reasoning, Tools

DeepSeek	DeepSeek	z. B. kleine Distill-Modelle, größere Reasoning-Varianten	Ja, vor allem kleine Varianten	kleine Modelle gut lokal; große Modelle nur eingeschränkt sinnvoll lokal	hoch	Reasoning, Analyse, Mathematik
Gemma	Google DeepMind / Google	z. B. 2B, 4B, 12B, 27B, Gemma-4-Varianten	Ja	kleine Modelle lokal gut einsetzbar; größere Varianten benötigen mehr Speicher	mittel bis hoch	allgemeine Nutzung, Reasoning, teils multimodal
Mistral	Mistral AI	z. B. 7B, Small, Large, Mixtral, Devstral, Magistral	Teilweise	offene kleinere Modelle lokal gut nutzbar; große Varianten eher nicht	hoch	Allround, Coding, Reasoning
Phi	Microsoft	z. B. Mini-, Multimodal- und Reasoning-Varianten	Ja	meist vergleichsweise ressourcenschonend	mittel bis hoch	effiziente lokale Nutzung, kompakte Modelle
GPT / o-Serie	OpenAI	z. B. GPT-4.x, mini, nano, o-Modelle	in der Regel nein	primär Cloud-/API-Nutzung	sehr hoch	allgemeine Nutzung, Reasoning, Produktivbetrieb
Claude	Anthropic	Haiku, Sonnet, Opus	in der Regel nein	primär Cloud-/API-Nutzung	hoch	Sprachverständnis, Analyse, Assistenz
Gemini	Google	Flash, Pro	in der Regel nein	primär Cloud-/API-Nutzung	hoch	allgemeine Nutzung, multimodale Verarbeitung
Grok	xAI	verschiedene Grok-Varianten	in der Regel nein	primär Cloud-/API-Nutzung	mittel	allgemeine Nutzung, Echtzeit-/Plattformintegration

Command / Aya	Cohere	Command, Aya	eher nein	primär API-/Unternehmensnutzung	mittel	Enterprise, RAG, mehrsprachige Anwendungen

---

Einordnung der Größenangaben

Bedeutung von `B`

Das Kürzel **B** steht in der Regel für **Billion**, also **Milliarden Parameter**.

Beispiele:

- **7B** = 7 Milliarden Parameter
- **14B** = 14 Milliarden Parameter
- **70B** = 70 Milliarden Parameter

Grundsätzlich gilt:

- kleinere B-Werte = geringerer Ressourcenbedarf
- größere B-Werte = tendenziell höhere Leistungsfähigkeit, aber auch höherer Speicherbedarf

Bedeutung von Angaben wie `A4B`

Zusätze wie **A4B** oder ähnliche Schreibweisen kommen häufig bei **Mixture-of-Experts-Modellen (MoE)** vor.

Dabei gilt typischerweise:

- die erste Größenangabe beschreibt die **Gesamtgröße des Modells**
- die A-Angabe beschreibt die **aktiv genutzten Parameter pro Verarbeitungsschritt**

Beispiel:

- **26B A4B** = Modell mit 26 Milliarden Parametern insgesamt, davon sind pro Schritt etwa 4 Milliarden aktiv

Wichtig ist dabei:

- auch wenn nur ein Teil aktiv genutzt wird, muss lokal oft **das gesamte Modell geladen werden**
- dadurch kann der Speicherbedarf weiterhin hoch bleiben

Typische Zusatzbezeichnungen

Kürzel / Begriff	Übliche Bedeutung
B	Milliarden Parameter
AxxB	aktivierte Milliarden Parameter, häufig bei MoE
MoE	Mixture of Experts
mini / nano / small / flash / fast	kleinere, schnellere oder günstigere Variante
pro / large / opus / ultra / heavy	leistungsstärkere Variante

reasoning / thinking	auf mehrstufiges Denken oder Problemlösen optimiert
vision	Bildverarbeitung möglich
multimodal	Verarbeitung mehrerer Eingabetypen, z. B. Text und Bild
instruct / IT	für Anweisungen bzw. Chatnutzung optimiert
preview	Vorabversion
turbo	meist auf Geschwindigkeit optimiert, jedoch nicht einheitlich definiert

Grobe Einschätzung lokaler Anforderungen

Die folgende Übersicht dient als praxisnahe Orientierung für lokale Nutzung, etwa mit Ollama oder vergleichbaren Laufzeitumgebungen:

Modellgröße	Grobe Einschätzung
1B bis 4B	leicht lokal nutzbar
7B bis 8B	gut lokal nutzbar
12B bis 14B	mittlere Anforderungen
26B bis 32B	hohe Anforderungen
70B und größer	meist nur mit leistungsstarker Hardware sinnvoll
MoE-Modelle	abhängig von Gesamtgröße und aktivem Anteil; Speicherbedarf kann trotz Effizienz hoch bleiben

Verbreitung der Modellfamilien

Die Verbreitung lässt sich nur grob einordnen, da keine einheitlichen Vergleichswerte über alle Anbieter hinweg vorliegen.

Sehr stark verbreitet

- Llama
- Qwen
- DeepSeek
- GPT

Stark verbreitet

- Claude
- Gemini
- Mistral
- Gemma

Mittel bis stark verbreitet

- Phi
- Cohere Command / Aya
- Grok

Typische Fokusse nach Modellfamilie

Modellfamilie	Typischer Schwerpunkt
Llama	allgemeine lokale Nutzung, Allround
Qwen	multilingual, Coding, Reasoning
DeepSeek	Reasoning, Analyse
Gemma	allgemeine Nutzung, effiziente moderne Open-Modelle
Mistral	Allround, teils Coding und Reasoning
Phi	kompakte, effiziente Modelle
GPT	allgemeine Nutzung, starke Cloud-Modelle
Claude	Sprachverständnis, Analyse, Assistenz
Gemini	multimodal, allgemeine Nutzung
Grok	plattformnahe Assistenz, allgemeine Nutzung
Command / Aya	Enterprise, RAG, Mehrsprachigkeit

Aktuelle Hauptversionen gängiger Modellfamilien

Stand: 21. April 2026

Modellfamilie	Aktuelle Hauptversion / aktuelle Linie	Typische aktuelle Varianten
OpenAI GPT / o-Serie	**GPT-5.4** als aktuelle Frontier-Modelllinie; zusätzlich **GPT-5.4-mini** und **GPT-5.4-nano**. Die ältere Reasoning-Linie **o3 / o4-mini** ist weiterhin dokumentiert, wird aber in den OpenAI-Modellseiten teils bereits als abgelöst eingeordnet. (OpenAI Entwickler)	GPT-5.4, GPT-5.4-mini, GPT-5.4-nano, o3, o4-mini
Claude	**Claude 4.7** ist aktuell die neueste genannte Generation; Anthropic hebt insbesondere **Claude Opus 4.7** hervor. In den Modelllinien bleiben außerdem **Sonnet** und **Haiku** als Klassen relevant. (Claude API Docs)	Opus 4.7, Sonnet 4.5+, Haiku 4.5+

Gemini	**Gemini 3.1 Pro** ist aktuell die neueste hervorgehobene Pro-Generation in den Gemini-API-Modellseiten. Zusätzlich werden **Gemini 3 Flash** und **Gemini 3.1 Flash-Lite** geführt. Parallel ist **Gemini 2.5 Pro** weiterhin dokumentiert. (Google AI for Developers)	Gemini 3.1 Pro, Gemini 3 Flash, Gemini 3.1 Flash-Lite, Gemini 2.5 Pro
Gemma	**Gemma 4** ist die aktuelle Hauptgeneration. Google listet für diese Generation die Größen **E2B, E4B**,**31B** und **26B A4B**. (Google AI for Developers)	Gemma 4 E2B, E4B, 31B, 26B A4B
Llama	**Llama 4** ist die aktuelle Hauptgeneration. Meta nennt derzeit insbesondere **Llama 4 Scout** und **Llama 4 Maverick** als veröffentlichte Modelle. (ai.meta.com)	Llama 4 Scout, Llama 4 Maverick
Mistral	Bei Mistral ist die aktuelle Hauptlinie breit aufgefächert. In den Modellübersichten werden besonders **Mistral Large 3**, **Devstral 2** und **Mistral Medium 3.1** als aktuelle Featured-Modelle hervorgehoben. (Mistral AI)	Mistral Large 3, Devstral 2, Mistral Medium 3.1
Qwen	**Qwen3** ist die aktuelle Hauptgeneration der allgemeinen Qwen-Familie. Daneben existiert mit **Qwen3-Coder** eine spezialisierte Coding-Linie. (Qwen)	Qwen3-0.6B bis 32B, Qwen3-30B-A3B, Qwen3-235B-A22B, Qwen3-Coder
DeepSeek	**DeepSeek-V3.2** ist die aktuelle allgemeine Hauptlinie laut DeepSeek. Zusätzlich wird **DeepSeek-V3.2-Speciale** genannt. In der Praxis bleibt auch **DeepSeek-R1** als Reasoning-Linie sehr relevant. (DeepSeek)	DeepSeek-V3.2, DeepSeek-V3.2-Speciale, DeepSeek-R1
Phi	**Phi-4** wird von Microsoft als aktuelle Hauptgeneration der Phi-Familie hervorgehoben. (Microsoft Azure)	Phi-4, Phi-4-mini, weitere spezialisierte Phi-4-Varianten je Plattform

Grok	**Grok 4.1** ist aktuell die neueste genannte Hauptversion in den xAI-News; zuvor wurde **Grok 4** eingeführt. (xAI)	Grok 4.1, Grok 4 Heavy, Grok Code Fast 1
Command / Aya	Bei Cohere ist **Command A** die aktuelle zentrale Hauptlinie; zusätzlich gibt es spezialisierte Varianten wie **Command A Reasoning** und **Command A Vision** . **Aya** bleibt die mehrsprachige Modellfamilie. (Cohere)	Command A, Command A Reasoning, Command A Vision, Aya

DSGVO-Einordnung beim Einsatz von KI-Modellen

Einsatzszenario	Beispiel	DSGVO-Einordnung	Typische Prüfpunkte	Risikoniveau
Rein lokal, offline	Modell läuft auf eigenem Rechner oder internem Server ohne externe Datenübertragung	Datenschutzrechtlich meist **am einfachsten** zu bewerten, weil keine automatische Übermittlung an einen externen KI-Anbieter erfolgt. Die DSGVO bleibt dennoch anwendbar, sobald personenbezogene Daten verarbeitet werden. (EDPB)	Rechtsgrundlage, Zweckbindung, Zugriffskontrolle, Protokollierung, Löschkonzept, Berechtigungskonzept	**Niedrig bis mittel**

Lokal im Unternehmensnetz, mit mehreren Nutzern	Interner KI-Server für Mitarbeitende	Ebenfalls vergleichsweise günstig, aber organisatorisch anspruchsvoller als Einzelnutzung. Es sind interne Rollen, Berechtigungen und Schutzmaßnahmen sauber zu regeln. (CNIL)	Rollen und Verantwortlichkeiten, TOMs, Zugriffstrennung, Logging, Datenschutzinformationen, interne Richtlinien	**Mittel**
Selbst gehostet in eigenem Rechenzentrum oder bei EU-Hoster	Eigene KI-Anwendung auf VPS oder dediziertem Server in der EU	Häufig gut beherrschbar, sofern Hosting, Fernzugriffe und Administratorenzugänge sauber geregelt sind. Auch bei EU-Hosting sind Verträge und technische Maßnahmen erforderlich. (CNIL)	AV-Vertrag, Serverstandort, Admin-Zugriffe, Verschlüsselung, Backups, Löschung, Incident-Prozesse	**Mittel**
Externe API / Cloud innerhalb der EU	KI-Dienst mit Verarbeitung in der EU	DSGVO-relevant und regelmäßig prüfungsbedürftig. EU-Standort erleichtert die Bewertung, ersetzt aber keine Prüfung von Rechtsgrundlage, Transparenz und Verträgen. (EDPB)	Anbieterrolle, AV-Vertrag oder Rollenabgrenzung, Zweckbindung, Datennutzung durch Anbieter, Speicherfristen, Betroffenenrechte	**Mittel bis hoch**

Externe API / Cloud mit Drittlandtransfer	US- oder sonstiger Nicht-EU-Anbieter	Regelmäßig **prüfungsintensiver** , weil zusätzlich Fragen zu internationaler Datenübermittlung, Schutzmechanismen und Anbieterzugriffen zu klären sind. Für US-Anbieter kann das EU-US Data Privacy Framework relevant sein, sofern der konkrete Empfänger darunter fällt. (European Commission)	Drittlandtransfer, Zertifizierung/Transfermechanismus, Vertragslage, Subprozessoren, Transparenz, Datennutzung für Training oder Verbesserung	**Hoch**
Nutzung mit anonymisierten oder wirksam pseudonymisierten Daten	Vorverarbeitete Fälle ohne direkte Personenbeziehbarkeit	Kann die datenschutzrechtlichen Risiken deutlich senken. Ob Daten oder sogar ein Modell als anonym gelten, ist jedoch im Einzelfall zu prüfen; die EDPB behandelt diese Frage ausdrücklich als fallbezogene Bewertung. (EDPB)	Qualität der Anonymisierung/Pseudonymisierung, Re-Identifikationsrisiko, Trennung von Zusatzwissen, Zugriffskonzepte	**Niedrig bis mittel**
Nutzung mit sensiblen Daten	Gesundheitsdaten, Sozialdaten, Beschäftigtendaten, besondere Kategorien	Besonders kritisch. Hier steigen die Anforderungen an Rechtsgrundlage, Schutzmaßnahmen, Zugriffsbeschränkungen und Dokumentation deutlich. (CNIL)	Art. 9 DSGVO, Erforderlichkeit, Datensparsamkeit, Schutzbedarf, DPIA, Zugriffstrennung, Verschlüs	

DSGVO-Einordnung nach Modellfamilie

> **Hinweis:** Diese Übersicht ist eine **praxisnahe Orientierung** für Dokumentationszwecke und **keine Rechtsberatung**. Maßgeblich ist immer der konkrete Einsatz: lokal, selbst gehostet, über API/Cloud, mit oder ohne personenbezogene Daten. Die EDPB hat klargestellt, dass die DSGVO auch für KI-Modelle gilt. (Anthropic)

Modellfamilie	Typischer Betriebsmodus	DSGVO-Tendenz	Modellbezug / praktische Einordnung	Prüfschwerpunkt
Llama	häufig **lokal / self-hosted**, da offen herunterladbar und „deploy anywhere“ beworben	**günstig bis mittel**, wenn lokal betrieben	Für Llama ist der DSGVO-Vorteil vor allem der **lokale oder eigene Betrieb**: Meta stellt die Modelle zum Download bereit und beschreibt sie als überall deploybar. Dadurch kann eine externe Übermittlung an einen Modellanbieter oft vermieden werden. (llama.com)	Serverstandort, interne Zugriffe, Logging, Löschung, keine unnötige Cloud-Anbindung
Gemma	häufig **lokal / self-hosted**	**günstig bis mittel**, wenn lokal betrieben	Gemma ist als offene Modellfamilie für lokale Nutzung relevant. Datenschutzrechtlich ist sie daher typischerweise **einfacher** als reine Cloud-Modelle, sofern keine personenbezogenen Daten an externe Dienste fließen. Die DSGVO-Frage hängt hier eher am Hosting als am Modellnamen. (Anthropic)	Lokale Verarbeitung, Zugriffskonzepte, Datenminimierung

Mistral	**hybrid** : Cloud, private Cloud, VPC, on-prem	**mittel** , bei on-prem günstiger	Mistral bewirbt ausdrücklich **private Deployments** , VPC- und On-Prem-Optionen sowie „your data stays within your walls“. Dadurch ist Mistral aus DSGVO-Sicht oft flexibler als reine SaaS-Modelle. Zusätzlich gibt es ein DPA. (Mistral AI)	AV-Vertrag, Hosting-Variante, Admin-Zugriffe, Datenfluss zwischen Cloud und On-Prem
Phi	oft **lokal** , alternativ Azure-/Microsoft-Umfeld	**günstig bis mittel** , je nach Hosting	Phi ist als kleine Modellfamilie gut für lokale Nutzung geeignet; Microsoft positioniert Phi zudem unter dem Aspekt „privacy and security“. Datenschutzrechtlich ist lokal am einfachsten, bei Azure-Betrieb kommt die übliche Cloud-/Vertragsprüfung hinzu. (Microsoft Azure)	Lokal vs. Azure, Verträge, Region, Speicherort
Qwen	sowohl **lokal** als auch **Cloud/API**	**lokal günstiger** , Cloud **mittel bis hoch**	Qwen ist als Modellfamilie lokal nutzbar, hat aber auch einen eigenen API-/Cloud-Zugang. Für die DSGVO ist deshalb wichtig, **welcher Weg genutzt wird** . Bei lokaler Nutzung ist die Bewertung deutlich einfacher als bei Verarbeitung über die Qwen-Cloud. (qwen.ai)	Ob lokal oder API, Anbieterrolle, Übermittlung, vertragliche Einbindung

DeepSeek	sowohl **lokal** als auch **Cloud/API**	**lokal klar günstiger**, Cloud **hoch**	Bei DeepSeek ist der Modellbezug besonders wichtig: Die Modelle können lokal laufen, aber die DeepSeek-Privacy-Policy sagt ausdrücklich, dass personenbezogene Daten zur Diensterbringung **in der Volksrepublik China verarbeitet und gespeichert** werden können. Für personenbezogene Daten ist deshalb der **lokale Betrieb** deutlich günstiger. (cdn.deepseek.com)	Drittlandtransfer, Speicherort China, sensible Daten vermeiden, möglichst self-hosted
GPT / o-Serie	typischerweise **Cloud/API**	**mittel bis hoch**	OpenAI ist klar cloud-orientiert. Positiv ist: Für Business-Angebote gibt es ein DPA; für Kunden im EWR/der Schweiz wird dieses mit **OpenAI Ireland Ltd.** geschlossen. OpenAI erklärt außerdem, dass **Business-Daten standardmäßig nicht zum Training verwendet werden**. Trotzdem bleibt Cloud-Verarbeitung DSGVO-prüfungsintensiv. (OpenAI)	Rechtsgrundlage, AV/DPA, Anbieterrolle, Transfermechanismus, Speicher- und Löschregeln

Claude	typischerweise **Cloud/API**	**mittel bis hoch**	Claude ist primär ein Cloud-Modell. Bei Anthropic ist relevant, dass es für Consumer-Nutzung eine **Opt-in-Logik** zur Datennutzung für Trainings-/Verbesserungszwecke gibt; zugleich verweist Anthropic für Unternehmen auf Trust-/Compliance-Unterlagen. Für DSGVO-Zwecke ist Claude daher vor allem als **vertrags- und plattformabhängiges Cloud-Modell** zu prüfen. (Anthropic)	Consumer vs. Business trennen, Opt-in/Datennutzung, Verträge, Speicherort

Gemini	typischerweise **Cloud/API**	**mittel bis hoch**	Bei Gemini ist die Unterscheidung zwischen **unpaid** und **paid tiers** besonders relevant: Google weist für Unpaid Services darauf hin, dass Eingaben/Ausgaben von Menschen geprüft und zur Produktverbesserung genutzt werden können und dass **keine sensiblen, vertraulichen oder personenbezogenen Daten** eingereicht werden sollen. Für Paid Tiers heißt es, dass Prompts/Responses **nicht** zur Produktverbesserung genutzt werden. (Google AI for Developers)	Tarifmodell, Datennutzung zur Verbesserung, Human Review, sensible Daten nur in geeignetem Vertragsrahmen

Grok	typischerweise **Cloud/API**	**hoch**	Grok ist derzeit im Kern ein cloudbasiertes Modellangebot. Für DSGVO-Zwecke ist es daher ähnlich wie andere Cloud-Modelle zu behandeln: rechtliche Grundlage, Datenfluss, Empfänger, Speicherort und Nutzungsbedingungen sind vor produktivem Einsatz mit personenbezogenen Daten zu prüfen. (Anthropic)	Vertragslage, Speicherort, Empfänger, Drittlandtransfer
Command / Aya (Cohere)	typischerweise **API/Enterprise**, teils **private deployment**	**mittel**, bei Private Deployment günstiger	Cohere ist für DSGVO-Zwecke relativ interessant, weil das Unternehmen sowohl **DPA** , **Zero Data Retention** für Enterprise-Fälle als auch **private deployment options** nennt. In der Privacy Policy steht zugleich, dass Trial-/Research-Umgebungen **nicht** für personenbezogene Daten gedacht sind. (Cohere)	Produktstufe prüfen, DPA anfordern, ZDR/Retention, Private Deployment bevorzugen

Kurztext für unter die Tabelle

> **Einordnung:**
> Der DSGVO-Bezug hängt bei KI-Modellen nicht nur am Hersteller, sondern stark am **typischen Betriebsmodus** der jeweiligen Modellfamilie. **Open-weight-Modelle** wie Llama, Gemma oder oft auch Qwen/DeepSeek/Mistral können lokal oder selbst gehostet betrieben werden und sind deshalb datenschutzrechtlich häufig günstiger zu bewerten. **Cloud-first-Modelle** wie GPT, Claude, Gemini oder Grok erfordern regelmäßig eine vertiefte Prüfung von Rechtsgrundlage, Vertragslage, Speicherort, möglichem Drittlandtransfer und Datenverwendung durch den Anbieter. (Anthropic)

Noch kürzere Fassung

Gruppe	DSGVO-Tendenz	Typische Modelle
Open-weight / lokal betreibbar	meist günstiger	Llama, Gemma, Phi, oft Qwen, DeepSeek, Mistral (llama.com)
Cloud-first / API-zentriert	meist prüfungsintensiver	GPT, Claude, Gemini, Grok, Command (OpenAI)

Zusammenfassung

Zur schnellen Einordnung kann folgende Struktur verwendet werden:

[Familie] + [Version] + [Größe] + [Spezialisierung]

Beispiele:

- **Gemma 4 27B IT**
- **Qwen3 14B**
- **DeepSeek R1**
- **Claude Sonnet**
- **GPT-4.1 mini**

Dabei gilt:

- **Familie** = Modellreihe
- **Version** = Generation oder Entwicklungsstand
- **Größe** = Parameteranzahl
- **Spezialisierung** = Einsatzschwerpunkt oder Optimierung

Übersicht gängiger KI-Plattformen (Stand April 2026) - Arbeitsdokument

> **Hinweis:**
> Diese Tabelle beschreibt **Plattformen**, nicht Modellfamilien. Eine Plattform kann eigene Modelle anbieten, Drittmodelle bündeln oder vor allem die lokale Ausführung vereinfachen. (Ollama Dokumentation)Kurze Einordnung

Übersicht gängiger KI-Plattformen mit Anbindung

Plattform	Anbieter	Typ	Typischer Einsatz	Lokal / Cloud	Anbindung / Ansprache	Typische Besonderheiten
Ollama	Ollama	Laufzeitumgebung und Modellplattform	Lokales Ausführen von LLMs, optional Cloud-Nutzung	**Lokal und Cloud**	**Lokale REST-API** über `localhost`, zusätzlich **OpenAI-kompatible Endpunkte**; dadurch oft auch über Tools nutzbar, die OpenAI-kompatible APIs erwarten. Lokal ist standardmäßig keine Authentifizierung nötig. (Ollama Dokumentation)	Lokaler Betrieb, OpenAI-Kompatibilität, Tool-Support

LM Studio	LM Studio	Desktop-Plattform für lokale Modelle	Modelle lokal laden, testen und per UI oder lokaler API nutzen	**Vor allem lokal**	**Lokaler API-Server** auf `localhost` oder im Netzwerk; nutzbar per **REST API**, über eigene Client-Bibliotheken und über **OpenAI-** sowie **Anthropic-kompatible Endpunkte**. (LM Studio)	Starker Fokus auf lokale/private Nutzung
OpenRouter	OpenRouter	Modell-Router / Unified API	Zugriff auf viele Modelle über eine einheitliche API	**Cloud**	**Einheitliche API**, die ausdrücklich mit dem **OpenAI SDK** nutzbar ist; OpenRouter beschreibt die Schemas als sehr ähnlich zur OpenAI Chat API. (OpenRouter)	Ein API-Zugang für viele Modelle und Provider

Perplexity API	Perplexity	Such- und Antwortplattform / API	Websuche, Recherche, Sonar, Agenten-Workflows	**Cloud**	**REST und SDKs** ; Perplexity nennt vier Kern-APIs: **Agent** , **Search** , **Sonar** und **Embeddings** . Die **Sonar API** ist zusätzlich mit **OpenAI-kompatiblen Clients** nutzbar. (Perplexity)	Fokus auf webgestützte Antworten und Recherche
OpenAI Platform	OpenAI	Modell- und API-Plattform	Produktive KI-Anwendungen, Agenten, multimodale Workflows	**Cloud**	**REST, Streaming, Realtime APIs** sowie offizielle **SDKs** ; zentrale Schnittstelle ist die **Responses API** . (OpenAI Entwickler)	Herstellerplattform mit breiter Tool- und Agentenunterstützung

Claude Platform	Anthropic	Modell- und API-Plattform	Claude-basierte Anwendungen und Assistenzsysteme	**Cloud**	**Messages API** plus offizielle **SDKs**; Anthropic dokumentiert den Einstieg über API-Key, Quickstart und Client-SDKs. (Claude API Docs)	Herstellerplattform für Claude-Modelle
Gemini API / Google AI Studio	Google	Modell- und Entwicklerplattform	Prototyping, multimodale Anwendungen, API-Nutzung	**Cloud**	**REST API** und offizielle **SDKs**; Google AI Studio dient als schneller Einstieg und zur API-Key-Erstellung. (Google AI for Developers)	Schneller Einstieg über AI Studio
Vertex AI	Google Cloud	Enterprise-AI-Plattform	Produktive KI-Anwendungen, verwaltete Bereitstellung	**Cloud**	Typisch über **Google-Cloud-/Vertex-AI-APIs** und SDKs; Anthropic dokumentiert Claude-Integrationen ausdrücklich auch über **Google Vertex AI**. (Anthropic)	Enterprise-Betrieb und Multi-Modell-Umgebungen

Hugging Face Inference Providers	Hugging Face	Modell-Hub und Inferenzplattform	Modelle testen, serverless nutzen oder produktiv anbinden	**Cloud**	Zugriff über **einheitliche API** und **Client-SDKs**; Hugging Face nennt ausdrücklich einen **OpenAI-kompatiblen Endpoint** für Inference Providers. (Hugging Face)	Viele Modelle, viele Inferenzpartner
IONOS AI Model Hub	IONOS	Europäische Modell- und API-Plattform	API-Zugriff auf Modelle in europäischer Cloud	**Cloud**	**API-first**; IONOS bietet eine reguläre API sowie eine **OpenAI-kompatible API**. (docs.ionos.com)	Europäischer Fokus, OpenAI-Kompatibilität
STACKIT AI Model Serving	STACKIT	Europäische AI-Serving-Plattform	Sichere Modellnutzung in der STACKIT Cloud	**Cloud**	Modelle sind über **API** nutzbar; STACKIT nennt die Schnittstelle ausdrücklich **OpenAI-kompatibel**. (Docs)	Europäische/souveräne Cloud-Ausrichtung

Amazon Bedrock	AWS	Voll gemanagte GenAI-Plattform	Zugriff auf viele Foundation Models in AWS	**Cloud**	**AWS API** , **API Keys** , regionale Endpunkte und **AWS SDKs** wie Boto3; Amazon dokumentiert zusätzlich OpenAI-kompatible Service-Endpunkte in Teilen des Angebots. (AWS Dokumentation)	Viele Drittmodelle unter AWS-Steuerung
noris network KI as a Service	noris network	Infrastruktur- und Serviceplattform	KI-Betrieb für Unternehmen , souveräne Hosting-/Cloud-Umgebungen	**Cloud / Private Cloud / Managed**	Anbieter beschreibt **KI as a Service** , Private-Cloud- und GPU-nahe Betriebsmodelle; die konkrete technische Ansprache ist eher **projekt- bzw. serviceabhängig** als über eine standardisierte öffentliche Entwickler-API dokumentiert . (STACKIT)	Fokus auf Betrieb, Hosting und Souveränität

Praktische Kurzregel für „API, n8n etc.“

Anbindungsart	Typische Plattformen	Einordnung

Direkt per REST/API	OpenAI, Anthropic, Gemini, Perplexity, OpenRouter, Bedrock, IONOS, STACKIT	Standardweg für Anwendungen und Automatisierung. (OpenAI Entwickler)
OpenAI-kompatibel	Ollama, LM Studio, OpenRouter, Perplexity Sonar, Hugging Face Inference Providers, IONOS, STACKIT	Besonders praktisch, weil viele vorhandene Tools und Clients weiterverwendet werden können. (Ollama Dokumentation)
SDKs / Client-Bibliotheken	OpenAI, Anthropic, Gemini, Hugging Face, AWS, LM Studio, Perplexity	Sinnvoll für produktive Anwendungen statt reiner HTTP-Aufrufe. (OpenAI Entwickler)
n8n direkt	Am einfachsten mit **OpenAI** über den offiziellen OpenAI-Node	n8n hat einen offiziellen **OpenAI-Node** ; für andere Dienste ist meist der **HTTP Request Node** nutzbar, sofern sie eine REST-API anbieten. (n8n Docs)
n8n indirekt über OpenAI-Kompatibilität	Ollama, LM Studio, OpenRouter, Perplexity Sonar, IONOS, STACKIT, teils Hugging Face	In der Praxis oft möglich, wenn in n8n ein OpenAI-kompatibler Endpoint oder notfalls der HTTP Request Node verwendet wird; offiziell dokumentiert ist bei n8n selbst vor allem der OpenAI-Node und der generische HTTP Request Node. (n8n Docs)

Neutrale Formulierung

> **Anbindung / Ansprache**
> KI-Plattformen werden typischerweise über **REST-APIs**, **offizielle SDKs** oder **OpenAI-kompatible Schnittstellen** angesprochen. Für Automatisierungswerkzeuge wie **n8n** ist dies relevant, weil entweder ein dedizierter Anbieter-Node genutzt werden kann oder ein generischer **HTTP-Request-Ansatz**. OpenAI-kompatible Plattformen sind dabei besonders integrationsfreundlich, da bestehende Clients und Workflows häufig mit geringem Anpassungsaufwand weiterverwendet werden können. (n8n Docs)

Sehr kurze Fassung

> **Plattformen** sind von **Modellfamilien** zu unterscheiden. Während Modellfamilien die eigentlichen KI-Modelle bezeichnen, stellen Plattformen die Umgebung für Auswahl, Ausführung, Routing, Hosting oder API-Zugriff bereit. Manche Plattformen sind lokal ausgerichtet, andere bündeln viele Fremdmodelle oder bieten eigene Hersteller-APIs an. (Ollama Dokumentation)