Local LLMs im Unternehmen:

Wann lohnt sich der lokale Betrieb?

Warum lokal betriebene Sprachmodelle 2026 zur ernsthaften Alternative zur Cloud geworden sind und wie Unternehmen den Umstieg strukturiert angehen

Ein Whitepaper von HMS Analytical Software
20.10.2026

Inhalt

1 Executive Summary 3
2 Einleitung und Problemstellung 3
1.2 Das eigentliche Problem 4
3 Technologische Entwicklungen: Was sich bei lokalen Sprachmodellen verändert hat . 5
3.1 Der geringe Leistungsabstand 5
3.2 Hardware für lokale Sprachmodelle: Warum die Einstiegshürden sinken 6
3.3 Ein ausgereiftes Ökosystem für Unternehmen 6
3.4 Der regulatorische Kontext 8
4 Local LLMs oder Cloud-KI: Analyse und Lösungsansätze 8
4.1 Welche KI-Workloads gehören wohin? 8
4.2 Vier Betriebsmodelle im Vergleich 10
4.3 Kosten lokaler Sprachmodelle: Wirtschaftlichkeit realistisch berechnen 11
4.4 Qualität lokaler Sprachmodelle sichern: Evaluation statt Bauchgefühl 12
4.5 Sicherheit: lokaler Sprachmodelle: Ein anderes Risikoprofil, nicht automatisch weniger Risiken 13
5 Praxisbeispiele und Use Cases von LLMs im Unternehmen 14
Use Case 1 – Dokumentenprüfung im regulierten Umfeld 14
Use Case 2 – Internes Wissensmanagement und Forschungsrecherche (RAG) 14
Use Case 3 – Entwicklungs- und Migrationsunterstützung mit Quellcode-Schutz..... 15
Use Case 4 – Hybrides Routing im Kundenservice 15
6 Lokale Sprachmodelle im Unternehmen einführen: Konkrete Handlungsempfehlungen 16
6.1 Local LLMs im Unternehmen: Typische Fehler vermeiden 16
6.2 Local LLMs im Unternehmen: Entscheidungshilfe für die Führungsebene 17
7 Wie HMS Analytical Software bei der Umsetzung unterstützt 17
8 Fazit 18
9 Quellen und weiterführende Literatur 19

1 Executive Summary

Lange galt: Wer Sprachmodelle auf Spitzenniveau nutzen will, muss in die Cloud. Diese Gleichung stimmt so nicht mehr. Offen verfügbare Modelle haben den Leistungsabstand zu den proprietären Spitzenmodellen in den vergangenen zwei Jahren drastisch verkürzt - von Jahren auf wenige Monate. Gleichzeitig sind die Betriebskosten pro Token gefallen, Quantisierungsverfahren machen leistungsfähige lokale Sprachmodelle (Local LLMs) auf geeigneter Hardware lauffähig, und der regulatorische Druck (EU AI Act, DSGVO, NIS2, DORA) erhöht den Wert von Datensouveränität und Datenschutz bei generativer KI.

Für Unternehmen bedeutet das: Die Frage lautet nicht mehr zwangsläufig „lokal **oder** Cloud", sondern „welche Last gehört wohin". Dieses Whitepaper beschreibt die Problemlage, ordnet die technische Entwicklung und den regulatorischen Rahmen ein, vergleicht lokale LLMs, Cloud-basierte Sprachmodelle und hybride Betriebsmodelle und liefert ein Entscheidungsraster für die Workload-Zuordnung. Vier Praxisszenarien und konkrete Handlungsempfehlungen helfen Unternehmen, die Kosten, Leistung und Datensicherheit lokaler Sprachmodelle zu bewerten und den Einstieg strukturiert zu planen.

2 Einleitung und Problemstellung

1.1 Ausgangslage: Lokale Sprachmodelle zwischen Innovationsdruck und Kontrollverlust

Generative KI ist in den Fachbereichen angekommen. Was 2023 als Experiment begann, ist heute in Kundenservice, Dokumentenprüfung, Softwareentwicklung und Wissensmanagement produktiv im Einsatz. Der Weg dorthin führte fast immer über einen API-Aufruf an einen der großen US-Cloudanbieter – das war der schnellste Weg zum Ergebnis.

Mit der Produktivsetzung tauchen jedoch Fragen auf, die im Prototypenstadium niemand stellen musste:

- **Wo liegen unsere Daten wirklich?** Prompts enthalten in der Praxis Kundendaten, Vertragsklauseln, Konstruktionsdetails, Patientenbezüge oder Quellcode. Je nach Autonomiegrad von Nutzern und KI-Systemen ist der Datenabfluss schwer zu kontrollieren.
- **Was kostet uns das im Jahr X?** Nutzungsbasierte Abrechnung skaliert linear mit dem möglichen Erfolg eines KI-Anwendungsfalls. Ein gelungener Rollout kann betriebswirtschaftlich ein Kostenrisiko sein.
- **Wie stabil ist unsere Grundlage?** Modelle werden abgekündigt, Antwortverhalten ändert sich nach Anbieter-Updates, Preise und Nutzungsbedingungen werden angepasst. Reproduzierbarkeit ist bei einem API-Modell dadurch nur eingeschränkt gegeben.

- **Wer haftet, wenn es schiefgeht?** Der EU AI Act verlangt von Betreibern Nachweise über eingesetzte Systeme, Risikomanagement und – bei Hochrisiko-Anwendungen – Protokollierung und menschliche Aufsicht.

Die drei häufigsten Missverständnisse

„Lokale Modelle sind deutlich schlechter."

Das war 2023 zutreffend. Heute erreichen offene Modelle auf Standard-Benchmarks Werte, die den proprietären Spitzenmodellen der jeweils vorangegangenen 6–12 Monate entsprechen. Für die meisten **unternehmensinternen** Aufgaben – Zusammenfassen, Klassifizieren, Extrahieren, Umformulieren, RAG-Antworten – ist der Unterschied in der Praxis kaum spürbar. Relevant bleibt der Abstand bei sehr langen Reasoning-Ketten, komplexer agentischer Werkzeugnutzung und Nischen-Fachwissen.

„Lokal ist automatisch günstiger."

Nein. Lokaler Betrieb hat eine hohe Fixkostenbasis (Hardware, Betrieb, Personal) und niedrige variable Kosten. Cloud ist umgekehrt. Der Break-even hängt fast ausschließlich von der Auslastung ab – nicht von der Technologie.

„Lokal ist automatisch DSGVO-konform."

Nein. On-Premises löst die Frage der Auftragsverarbeitung und des Drittlandtransfers, ersetzt aber weder Rechtsgrundlage, Zweckbindung, Löschkonzept, Berechtigungsmodell noch Betroffenenrechte. Ein schlecht abgesichertes lokales RAG-System kann datenschutzrechtlich schlechter dastehen als ein sauber vertraglich geregelter Cloud-Dienst.

1.2 Das eigentliche Problem

Die Problemstellung dieses Whitepapers ist **nicht** „Cloud ist schlecht". Sie lautet:

Viele Unternehmen treffen ihre KI-Architekturentscheidung einmalig, früh und implizit – und zwar auf Basis eines Informationsstands, der inzwischen überholt ist. Dadurch entstehen vermeidbare Kosten, vermeidbare Compliance-Risiken und eine strategische Abhängigkeit, die später nur teuer aufzulösen ist.

Wer heute nicht bewusst entscheidet, welche Workload lokal und welche in der Cloud läuft, entscheidet sich faktisch für maximale Abhängigkeit.

3 Technologische Entwicklungen: Was sich bei lokalen Sprachmodellen verändert hat

3.1 Der geringe Leistungsabstand

Die entscheidende Kennzahl zwischen offenen Modellen und proprietären Modellen ist nicht der absolute Benchmark-Wert, sondern der **zeitliche Abstand**: Wie lange dauert es, bis ein offen verfügbares Modell das Niveau des jeweils besten proprietären Modells erreicht?
Seit Januar 2026 liegen die leistungsfähigsten offenen Modelle im Schnitt rund vier Monate hinter den proprietären Spitzenmodellen.

Quelle: Epoch AI, „Open models lag state-of-the-art closed models by 4 months", Data Insight vom 29.05.2026 (Epoch Capabilities Index): https://epoch.ai/data-insights/open-closed-eci-gap

Vier Treiber stehen dahinter:
1. Wettbewerb um offene Gewichte. Mehrere große Anbieter veröffentlichen Modelle mit offenen Gewichten – Meta (Llama/Muse-Familie), Alibaba (Qwen), Mistral AI, Google (Gemma), Moonshot (Kimi), Z.ai (GLM), DeepSeek sowie kurzfristig in 2025 sogar OpenAI mit gpt-oss. Offene Gewichte sind zum strategischen Instrument im Plattformwettbewerb geworden.

2. Bessere Trainingsrezepte statt reiner Größe. Destillation, hochwertige synthetische Trainingsdaten und mehrstufiges Post-Training (SFT, RLHF/DPO, Reasoning-Training) machen kleine Modelle überproportional stark. Ein Modell mit 20–30 Mrd. Parametern erreicht heute Fähigkeiten, für die vor zwei Jahren mehrere hundert Milliarden Parameter nötig waren.

3. Mixture-of-Experts (MoE). Moderne Architekturen aktivieren pro Token nur einen Bruchteil ihrer Parameter. Ein Modell kann so hohe Gesamtkapazität besitzen, aber mit dem Rechenaufwand eines viel kleineren Modells arbeiten. Das verschiebt den Engpass von Rechenleistung hin zu Speicherkapazität und -Bandbreite – eine Ressource, die vergleichsweise günstig skaliert.

4. Quantisierung als Standard. Die Reduktion der Zahlendarstellung von 16 auf 8 oder 4 Bit senkt den Speicherbedarf um Faktor 2 bis 4, bei modernen Verfahren (AWQ, GPTQ, GGUF K-Quants, FP8, MXFP4) mit geringem Qualitätsverlust. Dadurch wird ein 70-Mrd.-Parameter-Modell auf einer einzelnen Workstation-GPU lauffähig.

3.2 Hardware für lokale Sprachmodelle: Warum die Einstiegshürden sinken

Klasse	Beispielhafte Ausstattung	Realistisch betreibbar	Typischer Einsatz
Arbeitsplatz	Notebook/Workstation mit Neural Processing Unit oder 16–32 GB Unified Memory	Modelle bis ca. 8–14 Mrd. Parameter, quantisiert	Einzelplatz, Offline-Assistenz, Entwickler-Prototyping
Abteilungsserver	1–2 Profi-GPUs mit je 48 GB	Modelle bis ca. 70 Mrd. Parameter, quantisiert; parallele Nutzung	Fachbereichs-RAG, Dokumentenverarbeitung, interner Chat
Rechenzentrum	4–8 GPUs mit 80–141 GB HBM	Große MoE-Modelle, hoher Durchsatz, mehrere Modelle parallel	Unternehmensweite Plattform, Batch-Verarbeitung, Feintuning

Wichtiger als das Datenblatt ist die **Speicherfaustformel**:

Benötigter Videospeicher ≈ (Parameteranzahl × Bit pro Parameter ÷ 8) + KV-Cache + Overhead

Beispiel: 70 Mrd. Parameter bei 4 Bit ≈ 35 GB Gewichte, plus KV-Cache. Der KV-Cache wächst linear mit Kontextlänge **und** Zahl paralleler Anfragen – er wird bei langen Kontexten schnell zum eigentlichen Kostentreiber und ist der am häufigsten unterschätzter Faktor in Kapazitätsplanungen.

3.3 Ein ausgereiftes Ökosystem für Unternehmen

Der Betrieb lokaler Modelle war lange Bastelarbeit. Heute existiert ein ausdifferenzierter Werkzeugkasten:

- **Serving mit hohem Durchsatz:** vLLM, SGLang, NVIDIA TensorRT-LLM, Hugging Face TGI – mit Continuous Batching, PagedAttention und Tensor-Parallelität.
- **Einfacher Einstieg und Edge:** Ollama, llama.cpp, LM-Studio – schnelle Inbetriebnahme, Unterstützung heterogener Hardware.
- **Kompatibilität:** Praktisch alle Serving-Lösungen bieten eine OpenAI-kompatible API. Das ist strategisch der wichtigste Punkt: **Ein Wechsel zwischen lokal und Cloud ist heute überwiegend eine Konfigurations-, keine Umbau-Entscheidung** – vorausgesetzt, die Anwendung wurde von Anfang an gegen diese Abstraktion entwickelt.

- **Orchestrierung und Governance:** Gateway-/Router-Schicht, Prompt- und Modellregister, Evaluations-Pipelines, Tracing und Kostenattribution.



Kernaussage: Das Gateway entkoppelt Anwendungen von Modellen und macht Lock-in beherrschbar.

3.4 Der regulatorische Kontext

Die folgende Übersicht dient ausschließlich der Orientierung. Das regulatorische Umfeld rund um KI ist ein umfangreiches und dynamisches Themenfeld, das wir hier bewusst nicht vertiefen. Die Tabelle benennt lediglich die Regelwerke, die für die Frage „lokal oder Cloud" typischerweise relevant sind. Sie erhebt keinen Anspruch auf Vollständigkeit und stellt **keine Rechtsberatung** dar. Die konkrete Bewertung für den jeweiligen Anwendungsfall sollte stets gemeinsam mit der Rechts-, Datenschutz- und Compliance-Abteilung erfolgen.

Regelwerk	Relevanz für die Betriebsentscheidung
EU AI Act	Pflichten zu Transparenz, Risikomanagement, technischer Dokumentation und Protokollierung – abhängig von der Risikoklasse der Anwendung. Lokaler Betrieb erleichtert Nachweisführung und Reproduzierbarkeit, entbindet aber nicht von den Pflichten.
DSGVO	Rechtsgrundlage, Zweckbindung, Datenminimierung, Betroffenenrechte, Auftragsverarbeitung, Drittlandtransfer. Lokaler Betrieb reduziert die Komplexität bei Transfer und AVV erheblich.
NIS2 / DORA	Anforderungen an Lieferkettensicherheit, Ausfallsicherheit und Auslagerungsmanagement. Ein externer Modellanbieter ist ein IKT-Drittdienstleister mit entsprechenden Nachweispflichten.
GxP / GAMP 5 (Life Sciences)	Computerized System Validation, Nachvollziehbarkeit und Änderungskontrolle. Eingefrorene, lokal gehostete Modellversionen sind hier ein struktureller Vorteil.
Weiteres Branchenrecht	Berufsgeheimnisträger (§ 203 StGB), Bank- und Versicherungsaufsichtsrecht, Medizinprodukterecht und Exportkontrolle können externe Verarbeitung einschränken.

4 Local LLMs oder Cloud-KI: Analyse und Lösungsansätze

4.1 Welche KI-Workloads gehören wohin?

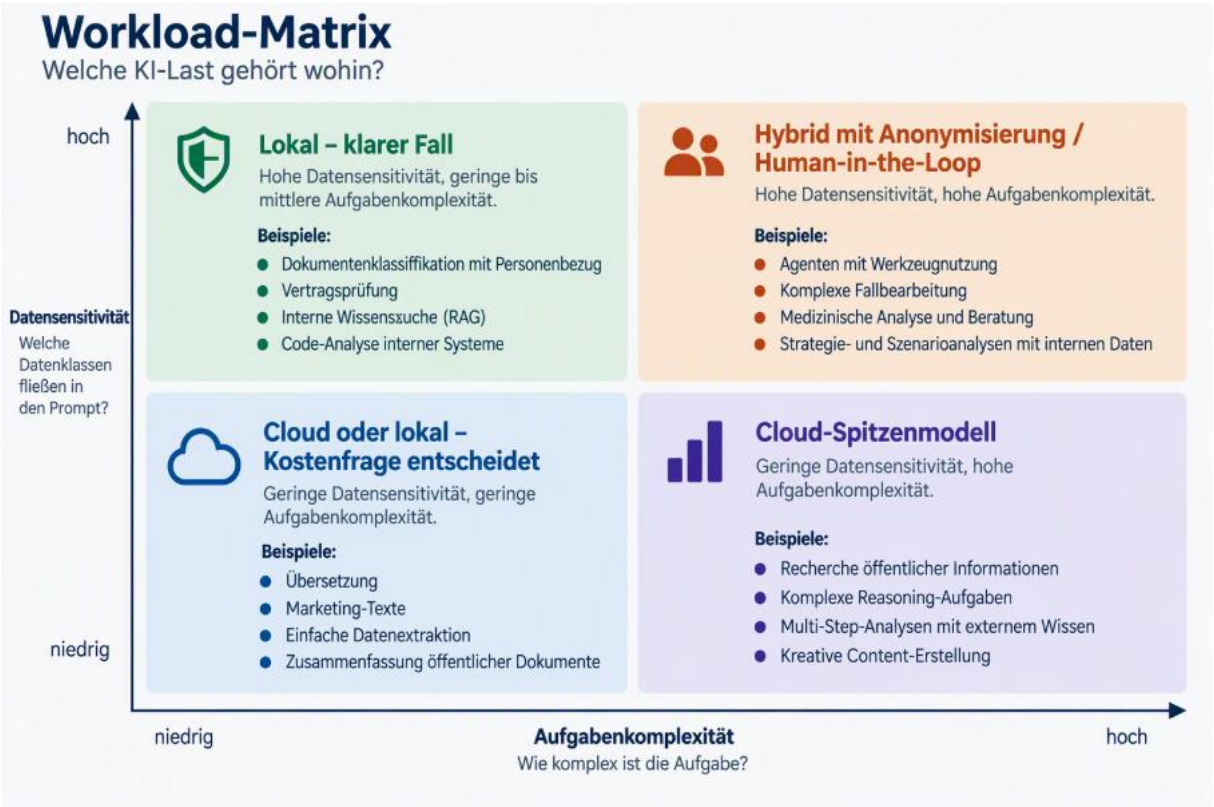
Ein hybrides Szenario – lokale und Cloud-Modelle parallel – ist eine mögliche Zielarchitektur, aber kein Selbstzweck. Der Parallelbetrieb bedeutet zusätzlichen Aufwand für Infrastruktur, Betrieb und Governance, und nicht jedes Unternehmen will oder braucht ihn. Wer ausschließlich Cloud-Dienste nutzt, kann damit gut aufgestellt sein, solange keine zwingenden Gründe für lokalen Betrieb vorliegen.
Sprechen jedoch einzelne Aspekte für lokale Modelle – etwa Kosten bei hohem Volumen, regulatorische Anforderungen oder bestimmte Anwendungsfälle mit sensiblen Daten –, ist es kein „Entweder- oder" mehr, sondern ein **Hybrid**. Dann lohnt es sich, jede Workload nach vier Kriterien zuzuordnen:

Kriterium 1 – Datensensitivität. Welche Datenklassen fließen in den Prompt? Je höher die Schutzklasse, desto stärker das Argument für lokalen Betrieb.

Kriterium 2 – Aufgabenkomplexität. Routineaufgaben (Klassifikation, Extraktion, Zusammenfassung, Formulierungshilfe, RAG-Antwort auf klar abgegrenztem Korpus) laufen lokal exzellent. Aufgaben mit langen Reasoning-Ketten, mehrstufiger Werkzeugnutzung oder breitem Weltwissen profitieren weiterhin von Spitzenmodellen.

Kriterium 3 – Volumen und Auslastung. Hohes, gleichmäßiges Volumen spricht für lokal (Fixkosten amortisieren sich). Niedriges oder stark schwankendes Volumen spricht für Cloud.

Kriterium 4 – Latenz- und Verfügbarkeitsanforderung. Produktionsnahe, offline-fähige oder latenzkritische Szenarien sprechen für lokal bzw. Edge.



4.2 Vier Betriebsmodelle im Vergleich

Kriterium	Public Cloud API	Managed / EU-Cloud, dediziert	Private Cloud / VPC	On-Premises / Edge
Datenkontrolle	gering	mittel	hoch	vollständig
Fixkosten	keine	gering	mittel	hoch
Variable Kosten	hoch	mittel	gering	sehr gering
Time-to-Value	Tage	Wochen	Wochen–Monate	Monate
Modellauswahl	Anbieterportfolio	eingeschränkt	frei	frei
Reproduzierbarkeit	eingeschränkt	mittel	hoch	vollständig
Betriebsaufwand	minimal	gering	mittel	hoch
Skalierung bei Lastspitzen	unbegrenzt	gut	begrenzt	hart begrenzt

Empfehlung: Für die meisten Unternehmen ist die Cloud heute der pragmatische Ausgangspunkt – schnell verfügbar, ohne eigenen Betriebsaufwand und mit Zugriff auf die leistungsfähigsten Modelle. Lokale oder dedizierte Modelle ergänzen sie dort, wo es dafür einen konkreten Grund gibt: sensible Daten, regulatorische Vorgaben, hohes und planbares Volumen oder Offline-Anforderungen. Sie sind damit die begründete Ausnahme, nicht der Normalfall. Entscheidend ist, die Architektur so zu gestalten, dass diese Ergänzung später ohne Umbau möglich bleibt.

Ein Zwischenweg wird oft übersehen: Zwischen „Public API" und „eigenes Rechenzentrum" liegt der Betrieb offener Modelle auf souveräner europäischer Infrastruktur oder in der eigenen Cloud-Tenancy. Man erhält damit freie Modellwahl, eingefrorene Modellversionen und Reproduzierbarkeit – ohne selbst Hardware zu beschaffen und zu betreiben. Für viele Unternehmen ist das der pragmatischste Einstieg: Die Vorteile offener Modelle lassen sich realisieren, bevor über eigene GPU-Investitionen entschieden werden muss.

Fünf Fragen an jeden Betriebsmodell-Kandidaten

1. Können wir eine Modellversion einfrieren und über Jahre unverändert weiterbetreiben?
2. Wo liegen die Daten während der Verarbeitung – und wer hat theoretisch Zugriff?
3. Bekommen wir die Protokolle, die wir für Nachweispflichten brauchen?
4. Was passiert, wenn der Anbieter das genutzte Modell abkündigt?
5. Wie hoch ist der Aufwand, dieselbe Anwendung auf einem anderen Modell zu betreiben?

4.3 Kosten lokaler Sprachmodelle: Wirtschaftlichkeit realistisch berechnen

Die häufigste Fehlkalkulation vergleicht Cloud-Tokenpreise mit dem Anschaffungspreis einer GPU. Das ist unvollständig. Ein belastbarer TCO-Vergleich (Total Cost of Ownership) über 36 Monate umfasst:

Lokale Seite:

- Hardware (GPU-Knoten, Netzwerk, Storage), Abschreibung über 3–4 Jahre
- Rechenzentrumskosten: Stellfläche, Strom, Kühlung, Redundanz
- Betriebspersonal: MLOps/Plattformbetrieb – **der am häufigsten vergessene Posten**
- Software, Monitoring, Sicherheit
- Redundanz für Ausfallsicherheit
- Opportunitätskosten der längeren Einführungszeit

Cloud-Seite:

- Token-Kosten in realistischer Hochrechnung, inklusive System-Prompts, RAG-Kontext, Wiederholungen und Agenten-Schleifen (Agenten verbrauchen ein Vielfaches der Tokens einer einfachen Chat-Anfrage – Faktor 5–20 ist keine Seltenheit
- Datentransfer- und Egress-Kosten
- Aufwand für Vertragsgestaltung, AVV, TIA, Anbieter-Audit
- Preis- und Modellrisiko über die Laufzeit

Die entscheidende Kennzahl heißt Auslastung

Eine lokale Inferenz-Infrastruktur ist umso wirtschaftlicher, je gleichmäßiger sie ausgelastet ist. Eine GPU, die nachts leerläuft, kostet trotzdem.

Praxisregel: Lokaler Betrieb wird interessant, wenn (a) das tägliche Token-Volumen hoch und einigermaßen planbar ist, (b) sich mehrere Anwendungsfälle dieselbe Infrastruktur teilen, und (c) Batch-Lasten (nächtliche Dokumentenverarbeitung, Reindexierung, Evaluationsläufe) die Lücken füllen.

Umgekehrt gilt: Für eine einzelne Anwendung mit wenigen hundert Anfragen pro Tag ist lokaler Betrieb betriebswirtschaftlich selten zu rechtfertigen – es sei denn, Compliance-Anforderungen erzwingen ihn.



4.4 Qualität lokaler Sprachmodelle sichern: Evaluation statt Bauchgefühl

Um den Erfolg eines lokalen LLM-Projektes zu bewerten, ist es wichtig, dass das Team messen kann, ob eine Änderung zu besseren oder schlechteren Ergebnissen führt. Das Modell allein entscheidet selten über Erfolg oder Misserfolg.

Was gebraucht wird:

1. **Ein anwendungsspezifisches Gold-Set.** mehrere repräsentative Fälle aus dem echten Fachbereich, mit erwarteten Ergebnissen. Öffentliche Benchmarks sagen wenig über die eigene Aufgabe aus.
2. **Definierte Metriken.** Je nach Aufgabe: Genauigkeit/F1 bei Klassifikation, Groundedness und Quellentreue bei RAG, Vollständigkeit bei Extraktion, dazu Latenz (P50/P95), Durchsatz und Kosten pro Vorgang.
3. **Automatisierte Regressionsläufe.** Bei jedem Modell-, Prompt- oder Retrieval-Wechsel. Ohne diese Automatisierung wird jeder Modellwechsel zum Risiko.
4. **Fachliche Abnahme.** Ein Bewertungsdurchlauf durch Fachanwender vor Produktivsetzung – nicht durch das Entwicklungsteam.
5. **Kontinuierliches Monitoring** im Betrieb, inklusive Stichprobenprüfung und Feedback-Kanal für Anwender.

Modellgröße ist selten die Stellschraube mit dem größten Hebel

In vielen Projekten bringt die Reihenfolge der Optimierung deutlich mehr als ein größeres Modell:

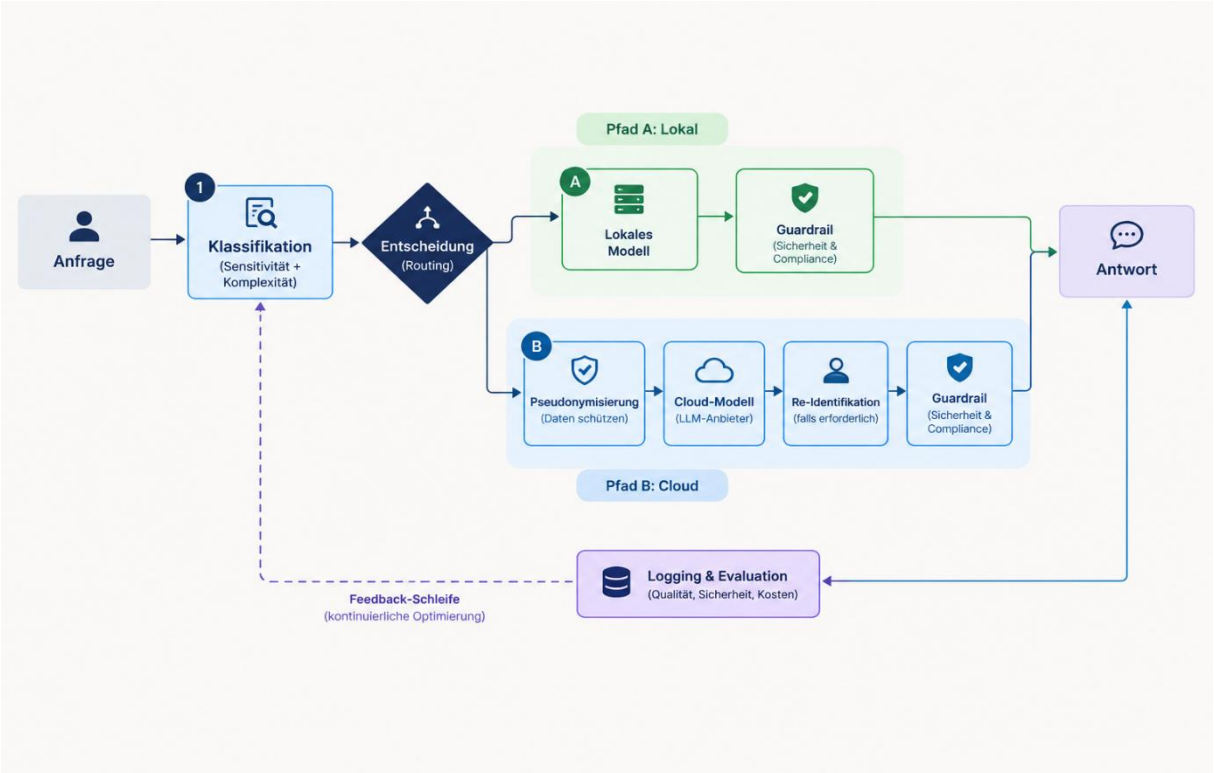
1. Datenqualität und Chunking → 2. Retrieval-Qualität (Hybrid-Suche, Reranking) → 3. Prompt- und Kontextgestaltung → 4. Strukturierte Ausgabe erzwingen (JSON-Schema/Grammar) → 5. erst dann größeres Modell oder Feintuning.

Teams, die mit Schritt 5 beginnen, verbrennen Budget und lösen das eigentliche Problem nicht.

4.5 Sicherheit: lokaler Sprachmodelle: Ein anderes Risikoprofil, nicht automatisch weniger Risiken

Lokaler Betrieb eliminiert das Risiko des unkontrollierten Datenabflusses zu Dritten. Er schafft aber neue Verantwortlichkeiten:

- **Prompt Injection und indirekte Injection** über Inhalte aus RAG-Quellen, E-Mails oder Webseiten. Dies ist die derzeit praktisch relevanteste Angriffsklasse gegen LLM-Anwendungen. Gegenmaßnahmen: Trennung von Instruktion und Daten, Least-Privilege bei Werkzeugzugriffen, Bestätigungsschritte vor wirkungsvollen Aktionen, Ausgabefilterung.
- **Lieferkette der Modelle.** Modellgewichte sind Artefakte aus externer Quelle. Erforderlich: Bezug aus verifizierten Quellen, Prüfsummen, sichere Serialisierungsformate, interne Registry, Lizenzprüfung (nicht jedes „offene" Modell ist frei kommerziell nutzbar – Nutzungsbeschränkungen und Schwellenwerte beachten).
- **Berechtigungen im RAG.** Der häufigste schwere Fehler: Ein Index, in dem alle Dokumente liegen, und ein Chatbot, der sie allen Nutzern zugänglich macht. Berechtigungen müssen **auf Ebene des Retrievals** durchgesetzt werden, nicht erst in der Antwortgenerierung.
- **Protokollierung.** Prompts und Antworten enthalten personenbezogene Daten. Logging braucht Rechtsgrundlage, Aufbewahrungsfristen, Zugriffsbeschränkung und Löschkonzept.



5 Praxisbeispiele und Use Cases von LLMs im Unternehmen

Use Case 1 – Dokumentenprüfung im regulierten Umfeld

Branche: Versicherung, Finanzdienstleistung, öffentliche Verwaltung

Ausgangslage: Ein Fachbereich prüft täglich mehrere tausend eingehende Dokumente mit Personenbezug auf Vollständigkeit und Kategorie. Externe Verarbeitung ist aufgrund der Datenklassifizierung ausgeschlossen.

Lösung: Ein quantisiertes Modell mittlerer Größe auf zwei GPUs im eigenen Rechenzentrum. Strukturierte Ausgabe über erzwungenes JSON-Schema. Confidence-Schwelle: Fälle unterhalb des Schwellwerts gehen automatisch in die manuelle Prüfung.

Use Case 2 – Internes Wissensmanagement und Forschungsrecherche (RAG)

Branche: Pharma und Life Sciences, Fertigung, Energie

Ausgangslage: Verteiltes Wissen in Wiki, Fileshares, Ticketsystem, SharePoint und Studien- bzw. Versuchsdokumentation. Suchzeiten und Rückfragen an Kollegen binden erhebliche Arbeitszeit. Forschungs- und Konstruktionsunterlagen dürfen das Unternehmen nicht verlassen.

Lösung: Lokales Embedding-Modell plus lokales Generierungsmodell, hybride Suche (semantisch + Stichwort) mit Reranking, Berechtigungsfilter auf Retrieval-Ebene, verpflichtende Quellenangabe in jeder Antwort.

Use Case 3 – Entwicklungs- und Migrationsunterstützung mit Quellcode-Schutz

Branche: branchenübergreifend, besonders relevant bei Legacy-Modernisierung

Ausgangslage: Entwicklungsteams wollen KI-Unterstützung – etwa bei der Analyse und Migration gewachsener Altanwendungen. Der Quellcode enthält Geschäftslogik und darf das Unternehmensnetz nicht verlassen.

Lösung: Lokal betriebenes Code-Modell hinter einer OpenAI-kompatiblen API, angebunden an die vorhandenen Entwicklungswerkzeuge. Kleineres Modell für Autovervollständigung (Latenz kritisch), größeres Modell für Chat und Refactoring.

Use Case 4 – Hybrides Routing im Kundenservice

Branche: Versicherung, Energieversorgung, Handel

Ausgangslage: Hohes Anfragevolumen, überwiegend Standardfälle, aber gelegentlich komplexe Sachverhalte.

Lösung: Ein Gateway klassifiziert eingehende Anfragen. Standardfälle beantwortet ein lokales Modell aus dem Wissensbestand. Komplexe Fälle werden – nach Pseudonymisierung personenbezogener Daten – an ein Cloud-Spitzenmodell weitergeleitet. Alles wird protokolliert und ausgewertet.

6 Lokale Sprachmodelle im Unternehmen einführen: Konkrete Handlungsempfehlungen

6.1 Local LLMs im Unternehmen: Typische Fehler vermeiden

Fehler	Konsequenz	Besserer Weg
Technologie vor Anwendungsfall wählen	Infrastruktur ohne Nutzen	Mit zwei bis drei messbaren Anwendungsfällen starten
Anwendung direkt gegen einen Anbieter-SDK entwickeln	Lock-in auf Code-Ebene	Von Beginn an gegen eine abstrahierende Gateway-Schicht entwickeln
Ohne Evaluationsset starten	Kein belastbarer Qualitätsnachweis, Modellwechsel unmöglich	Gold-Set **vor** der ersten Implementierung erstellen
Größtes verfügbares Modell wählen	Hohe Kosten, hohe Latenz, kaum Mehrwert	Kleinstes Modell wählen, das die Gütekriterien erfüllt
KV-Cache und Parallelität ignorieren	Kapazitätsplanung bricht unter Last zusammen	Lasttest mit realistischer Kontextlänge und Nutzerzahl
Betriebspersonal nicht einplanen	Pilot läuft, Produktivbetrieb scheitert	Betriebsmodell und Verantwortlichkeiten vor dem Pilot klären
Modelllizenzen nicht prüfen	Rechtliches Risiko bei kommerzieller Nutzung	Lizenzprüfung als fester Schritt in der Modell-Registry
Berechtigungen erst in der Antwort filtern	Datenschutzvorfall	Zugriffskontrolle im Retrieval durchsetzen
Datenschutz und Betriebsrat spät einbeziehen	Rollout-Stopp kurz vor Produktivsetzung	Beteiligung ab Woche 1

6.2 Local LLMs im Unternehmen: Entscheidungshilfe für die Führungsebene

Sieben Fragen für das nächste Management-Meeting:

1. Welche unserer KI-Anwendungsfälle verarbeiten Daten, die das Haus nicht verlassen dürfen – und wissen wir das sicher?
2. Was zahlen wir heute für KI-Dienste, und wie entwickelt sich das bei zehnfachem Nutzungsvolumen?
3. Könnten wir unseren Modellanbieter innerhalb von vier Wochen wechseln? Was würde das kosten?
4. Können wir messen, ob unsere KI-Anwendungen besser oder schlechter werden?
5. Wer ist im Haus für den Betrieb von KI-Infrastruktur verantwortlich – namentlich?
6. Sind unsere KI-Anwendungen unter dem EU AI Act korrekt eingestuft und dokumentiert?
7. Existiert Schatten-IT, also nicht freigegebene KI-Nutzung in den Fachbereichen?

Wer mehr als zwei dieser Fragen nicht klar beantworten kann, hat Handlungsbedarf – unabhängig von der Betriebsform.

7 Wie HMS Analytical Software bei der Umsetzung unterstützt

HMS berät nicht technologiegetrieben, sondern anwendungsfallgetrieben. Die Frage „lokal oder Cloud" beantworten wir nicht vorab, sondern als Ergebnis einer Bewertung: Welches Modell erfüllt die fachlichen Gütekriterien, zu welchen Kosten, unter welchen regulatorischen Rahmenbedingungen?

Diese Haltung ist möglich, weil wir das gesamte Spektrum abdecken – von klassischem Machine Learning über Generative AI bis zu agentenbasierten Systemen – und als Partner mehrerer Hyperscaler und Plattformanbieter herstellerneutral beraten können.

8 Fazit

Der Vorsprung proprietärer Cloud-Modelle ist real, aber er ist kleiner geworden und er ist nicht überall entscheidungsrelevant. Für einen erheblichen Teil der unternehmensinternen Anwendungsfälle– Klassifikation, Extraktion, Zusammenfassung, wissensgestützte Antworten – liefern lokal betreibbare Modelle heute ausreichende Qualität bei besserer Kostenvorhersehbarkeit und vollständiger Datenkontrolle.

Für Unternehmen mit einem konkreten Anwendungsfall für lokale LLMs lautet die Empfehlung daher nicht Lagerwechsel, sondern **Wahlfreiheit**: eine Architektur, in der die Modellentscheidung je Workload getroffen und jederzeit revidiert werden kann. Wer diese Entkopplung herstellt, profitiert von jedem künftigen Fortschritt – auf beiden Seiten. Für Unternehmen ohne solchen Anwendungsfall ist lokaler Betrieb kein Muss; die Cloud bleibt dort eine tragfähige Lösung.

9 Quellen und weiterführende Literatur

Leistungsvergleiche und Marktbeobachtung

- Epoch AI – Analysen zum Abstand zwischen offenen und proprietären Modellen: https://epoch.ai/
- Artificial Analysis – unabhängige Benchmarks zu Qualität, Latenz und Preis: https://artificialanalysis.ai/
- LMArena – nutzerbasierte Modellvergleiche: https://lmarena.ai/
- Stanford HAI, AI Index Report: https://aiindex.stanford.edu/report/

Modelle und Anbieter (offene Gewichte)

- Meta AI – Llama: https://ai.meta.com/llama/
- Alibaba – Qwen: https://qwenlm.github.io/blog/
- Mistral AI: https://mistral.ai/news/
- Google – Gemma: https://ai.google.dev/gemma
- Microsoft – Phi: https://azure.microsoft.com/en-us/products/phi
- OpenAI – gpt-oss: https://openai.com/index/introducing-gpt-oss/
- DeepSeek: https://www.deepseek.com/
- Hugging Face Model Hub: https://huggingface.co/models

Betrieb und Werkzeuge

- vLLM: https://docs.vllm.ai/
- llama.cpp: https://github.com/ggml-org/llama.cp
- Ollama: https://ollama.com/
- Hugging Face Text Generation Inference: https://huggingface.co/docs/text-generation-inference
- NVIDIA TensorRT-LLM: https://github.com/NVIDIA/TensorRT-LLM
- MLCommons MLPerf Inference: https://mlcommons.org/benchmarks/inference-datacenter/

Sicherheit

- OWASP Top 10 for LLM Applications: https://owasp.org/www-project-top-10-for-large-language-model-applications/
- NIST AI 100-2, Adversarial Machine Learning: https://csrc.nist.gov/pubs/ai/100/2/e2025/fina
- NIST AI Risk Management Framework: https://www.nist.gov/itl/ai-risk-management-framework
- BSI – Generative KI-Modelle, Chancen und Risiken: https://www.bsi.bund.de

Recht und Regulierung

- Verordnung (EU) 2024/1689 (AI-Act), Volltext: https://eur-lex.europa.eu/eli/reg/2024/1689/oj
- Europäische Kommission, AI Act Übersicht: https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai
- Verordnung (EU) 2016/679 (DSGVO): https://eur-lex.europa.eu/eli/reg/2016/679/oj
- Richtlinie (EU) 2022/2555 (NIS2): https://eur-lex.europa.eu/eli/dir/2022/2555/oj
- Verordnung (EU) 2022/2554 (DORA): https://eur-lex.europa.eu/eli/reg/2022/2554/oj