Bundesministerium
Innovation, Mobilität
und Infrastruktur

F&E Dienstleistung „Wissenschaftliche Vorbereitung eines LLMs für die öffentliche Verwaltung“

Wien, 2026

Impressum

Medieninhaber, Verleger und Herausgeber:
Bundesministerium für Innovation, Mobilität und Infrastruktur,
Radetzkystraße 2, 1030 Wien
Autorinnen und Autoren: Brigitte Krenn (OFAI), Bernhard Moser (SCCH), Theodorich Kopetzky (SCCH), Alexandra Ciarnau (DORDA Rechtsanwälte), Arthur Erdem (IPEF), Stephanie Gross (OFAI), Andreas Gruber (Trustifai, TÜV Austria), Sophie Kieselbach (IPEF), Tamas Madl, Johann Petrak (OFAI), Simon Schmid (SCCH), Eugenia Stamboliev (Women in AI)
Projektkoordination: Brigitte Krenn
Wien, 2026.

Hinweis zum Inhalt:
Da es sich bei der Entwicklung von LLMs um einen äußerst dynamischen Forschungs- und Entwicklungsbereich handelt, sind die Ausführungen im Dokument entsprechend als Zeitaufnahme zu betrachten. Im vorliegenden Dokument wurden Entwicklungen bis inklusive April 2026 berücksichtigt.

Rückmeldungen: Ihre Überlegungen zu vorliegender Publikation übermitteln Sie bitte an iii5@bmimi.gv.at.

Inhalt

Einleitung

Der folgende Bericht ist die Zusammenfassung der Ergebnisse des Projektes „Wissenschaftliche Vorbereitung eines LLMs für die öffentliche Verwaltung“. Das Projekt wurde im Rahmen des Calls „AI Ökosysteme2025: AI for Tech & AI for Green“ Förderprojekt („F&E Dienstleistung“) auf Initiative von und finanziert durch das Bundesministerium für Innovation, Mobilität und Infrastruktur (BMIMI) über die Österreichische Forschungsförderungsgesellschaft (FFG) ausgeschrieben. Die Auswahl erfolgte durch eine internationale Fachjury.

Ziel der F&E-Dienstleistung war es, „die Voraussetzungen zu schaffen, damit externe Prozesse sachlich fundiert vorbereitet und bei Bedarf angestoßen werden können. Mit dieser Forschungs- und Entwicklungsdienstleistung soll eine tragfähige Wissensbasis geschaffen werden, die öffentliche Stellen in ihrer Entscheidungsfindung zu einem möglichen Bundes-LLM fachlich unterstützt. Sie dient der konzeptionellen Vorbereitung eines potenziellen Anwendungsbereichs von Künstlicher Intelligenz im öffentlichen Sektor, wobei insbesondere das Zusammenspiel mit öffentlich verfügbaren Daten im Mittelpunkt steht.“ (cf. Ausschreibungsleitfaden[1])

Im Rahmen des Projektes befasste sich ein Konsortium von Expert:innen aus KI-Forschung und -anwendung, Recht, Ethik und Umweltverträglichkeitsprüfung mit der Analyse der Anforderungen und Erwartungen an den Einsatz von KI in der öffentlichen Verwaltung und damit verbundenen technischen und strategischen Realisierungsperspektiven von KI-basierten Systemen. Die Analysen reichten von der Einbindung bestehender LLMs in größere Anwendungssysteme über Modelladaptierung und Wissensdestillation bis hin zum Training eines eigenen österreichischen LLMs von Grund auf (Foundation Model Training). Dabei wurden rechtliche, ethische und ökologische Aspekte diskutiert, in Ergänzung zu den technischen Realisierungsmöglichkeiten, den benötigten Daten und Expertisen, sowie den erforderlichen Human-, Soft- und Hardwareressourcen für Entwicklung und Betrieb der Systeme.

Das AI Policy Forum, eine interministerielle Arbeitsgruppe zur Umsetzung der nationalen KI Strategie (besetzt mit Vertreter:innen aus allen Resorts, unter dem Vorsitz von Bundeskanzleramt (BKA) und BMIMI) diente gemeinsam mit der CDO-Taskforce, dem Zusammenschluss

[1] fdoc.ffg.at/s/vdb/public/node/content/WFTwdy01RnqU52H-ClLOjQ/1.0?a=true

der Chief Digital Officers aller Ministerien, als Anlaufstelle und Multiplikator für eine umfangreiche, im Projekt durchgeführte Umfrage hinsichtlich der Anforderungen und Erwartungen an den Einsatz von KI in der öffentlichen Verwaltung. Ergebnisse aus den im Projekt durchgeführten Analysen wurden wiederum an die Stakeholder-Community in einem Workshop zur Kommentierung zurückgespielt.

Basierend auf den Ergebnissen der Auswertung des Rücklaufs der Befragung zum Einsatz von LLM-basierten Systemen in den jeweiligen Ministerien, den Inputs verschiedener Stakeholdergruppen[2] und dem State-of-the-Art in der Forschung zu KI-basierten Applikationen, lassen sich folgende drei Kernbotschaften ableiten:

Kernbotschaften

Vielfalt statt Monokultur: Synthese als Leitprinzip für öffentliche LLM-Architekturen

Die vielfältigen Anforderungen an AI-Technologie in der öffentlichen Verwaltung erfordern den Einsatz unterschiedlicher Large Language Models (LLMs) sowie kleinerer, spezialisierter Modelle (Small Language Models SLMs). Dazu braucht es, statt einer Monokultur, eine national abgestimmte Referenzarchitektur, die Interoperabilität und Transparenz fördert.

Kontinuierliche Qualitätssicherung im KI-Lebenszyklus

Essenziell für belastbare und adaptierbare KI-Infrastrukturen in der Verwaltung ist ein erweiterbarer, lebenszyklusweiter Test- und Validierungsrahmen, der Data Drift, Modell-Updates, AI-Stack-Änderungen und domänenspezifische Risiken adressiert. Dadurch können auch ethische Anforderungen, wie Inklusivität, Oversight und Transparenz, im Life-Cycle des jeweiligen Modells besser gewährleistet werden.

[2] Inkl.: AI Policy Forum, CDO-Taskforce, dem Bundesrechenzentrum (BRZ), der AI Factory (AI:AT), der APA als Institution mit Zugang zu großen Mengen österreichischer Text- und Sprachdaten, sowie Vertreter:innen aus Entwicklung und Betrieb von GPT-NL, dem nationalen niederländischen Foundation Model

Digitale Souveränität durch PPP-Strukturen

Schlüssel für digitale Souveränität und Effizienz im öffentlichen Sektor: PPP-Strukturen (i) zur Bündelung österreichischer KI-Kompetenzen sowie (ii) zur Verknüpfung mit europäischen LLM-Initiativen.

Diese drei Themen werden in den folgenden drei Kapiteln „Synthese als Leitprinzip für öffentliche LLM-Architekturen“, „Kontinuierliche Qualitätssicherung“ und „Digitale Souveränität durch PPP-Strukturen“ diskutiert. Im Abschlusskapitel „Synthese und strategische Empfehlungen“ wird eine phasen-orientierte Roadmap für die Entwicklung und Einführung KI-basierter Systeme für die öffentliche Verwaltung skizziert, Beispielarchitekturen werden vorgestellt und das Fazit der Studie wird zusammengefasst.

Synthese als Leitprinzip für öffentliche LLM-Architekturen

Der Einsatz von LLMs im öffentlichen Bereich oder auch im Firmenkontext ist sinnvollerweise aus der Anwendungsperspektive und der **Einbindung eines oder mehrerer LLMs in einen größeren Systemkontext** zu betrachten. Entsprechend diskutieren wir diese Realisierungsvariante als erstes und gehen dann über zur **Adaptierung existierender Modelle für spezifische Nutzer:innenanforderungen**. Dabei beschäftigen wir uns insbesondere mit Wissensdestillation (Knowledge Distillation) und Fine-Tuning. In weiterer Folge diskutieren wir das **Training von Foundation Models** und beziehen uns dabei exemplarisch auf die Entwicklung nationaler Modelle wie dem Schweizer Foundation Model Apertus[3] und dem Niederländischen Foundation Model GPT-NL[4].

> Alle drei Realisierungsvarianten sind jeweils mit rechtlichen Fragestellungen verbunden.

Die rechtliche Bewertung hängt dabei maßgeblich vom konkreten Anwendungsfall ab. Da die möglichen Einsatzszenarien von LLM-basierten Systemen sehr vielfältig sind und eine belastbare rechtliche Einschätzung stets den konkreten Sachverhalt – einschließlich der verwendeten Daten, der Funktionalität sowie der produkt- und kontextbezogenen Einbindung des LLM – voraussetzt, werden im Folgenden die allgemeinen rechtlichen Rahmenbedingungen für eine erste strategische Entscheidungsbasis dargestellt. “KI-Recht” ist weiters kein in sich geschlossenes Rechtsgebiet, sondern eine Querschnittsmaterie. Für Entwicklung, Training, Integration und Betrieb des Bundes-LLM sind unterschiedliche Rechtsnormen parallel zu beachten. Welche Anforderungen jeweils gelten, hängt maßgeblich von der Projektphase, der technischen Implementierungs-variante und dem konkreten Einsatzszenario ab.

[3] swiss-ai.org/apertus

[4] gpt-nl.nl/ Seite auf holländisch, appl.ai/projects/gpt-nl, nldigitalgovernment.nl/news/dutch-ai-model-gpt-nl-to-start-trial-with-virtual-assistant-gem/, aic4nl.nl/en/

Ethische Kernbotschaften (für alle Modellvarianten)

Die Wahl der Modellarchitektur stellt nicht nur eine technische Entscheidung dar, sondern ist als grundlegende ethische Weichenstellung für das Organisieren von Wissen in Administration und Verwaltung zu verstehen. Sie bestimmt, in welchem Ausmaß Kontrolle, Transparenz, Verantwortlichkeit und Kontextsensitivität im weiteren Systemverlauf überhaupt realisierbar sind.

Welches Modell ist am ethischsten?

Foundation-Modelle bieten hohe Flexibilität, sind jedoch aufgrund eingeschränkter Transparenz und Kontrollierbarkeit nur bedingt für kontextsensitive und verantwortungskritische Anwendungen geeignet. Fine-Tuning ermöglicht eine stärkere Steuerung des Modellverhaltens, geht jedoch mit einer Verfestigung normativer Entscheidungen und geringerer Anpassungsfähigkeit einher. RAG-Systeme stellen einen Mittelweg dar, verschieben zentrale ethische Fragestellungen jedoch in die Auswahl und Gewichtung von Wissensquellen, wodurch neue Formen dynamischer Verzerrung entstehen. Zentrale Spannungsfelder liegen nicht zwischen „guten“ und „schlechten“ Modellen, sondern entlang struktureller Zielkonflikte:

- zwischen Leistungsfähigkeit und Kontrollierbarkeit (und Souveränität)
- zwischen Effizienz und Verantwortungszuschreibung
- zwischen Standardisierung und kontextspezifischer Angemessenheit

Vor diesem Hintergrund lässt sich die Modellwahl nicht isoliert treffen, sondern muss im Zusammenhang mit einer übergeordneten Systemarchitektur betrachtet werden. Dies lässt sich mit Hilfe eines **ethischen Life-Cycle-Ansatzes (ELCA)** erreichen. Ein solcher „Life-Cycle“-Ethikansatz geht davon aus, dass ethische Anforderungen wie Transparenz, Verantwortlichkeit, Fairness und Datenschutz nicht isoliert auf einzelne Phasen oder technische Komponenten beschränkt werden können. Dieser Ansatz orientiert sich an Fragen und Schritten aus der EU-Richtlinie “Ethics by Design”, die auf die Modellvarianten angewendet werden.

Es ist essentiell, einen **Überblick über den gesamten Lifecycle** – von der Datenauswahl über statistische Lernprozesse bis hin zur normativen Ausrichtung und institutionellen Anwendung – im Blick zu haben. Dabei ist es hilfreich zu sehen, wo und wie ethische Fragen zu stellen sind und welche Schwachstellen sie beleuchten. Empfehlungen und Lösungen sind

jedoch immer kontextbezogen und nur in Kooperation mit technischen und administrativen Entscheidungen zu treffen.

Vorteile des Ethischen Life-Cycle-Ansatzes sind:

- Transparenz und Identifikation von ethischen Entscheidungen und Hürden
- die Schritt-für-Schritt-Koordinierung von ethischen Dimensionen anhand des technischen Designs.

Im Folgenden werden die unterschiedlichen Realisierungsvarianten LLM-Einbindung, Modelloptimierung und Foundation Model Training anhand verschiedener Kriterien wie Entwicklungszeit, Souveränität und Betriebsmodelle diskutiert.

LLM-Einbindung am Beispiel Retrieval-Augmented Generation (RAG) Systeme

Was sind RAG-Systeme?

RAG-Systeme (Lewis et al., 2020) sind Softwaresysteme, die Information Retrieval Techniken mit LLMs kombinieren. Somit können Wissensquellen, wie z.B. behörden- oder firmeninterne Dokumente, die nicht in den Trainingsdaten des verwendeten LLMs enthalten sind, dem LLM zugänglich gemacht werden und für die Antwortgenerierung herangezogen werden, inklusive der mit den internen Dokumenten verknüpften Metadaten. RAG kann Halluzinationen reduzieren, setzt dafür aber qualitätsgesicherte Quellen, korrektes Retrieval, Zugriffskontrolle und Quellenbindung in der Antwortgenerierung voraus.

RAG-Systeme bestehen oft aus verschiedenen Komponenten: dem Retriever, einem Reranker und dem Generator. Der Retriever durchsucht eine Wissensbasis – z.B. eine Vektordatenbank - nach Dokumenten, die semantisch am besten zu einer gestellten Benutzer:innenanfrage passen. Diese Dokumente (beziehungsweise Dokumentteile) werden dann mit einem Reranker neu gereiht. Dabei wird jedes Schnipsel gemeinsam mit der Anfrage von einem LLM (oft ein speziell für Reranking trainiertes LLM) neu bewertet. Diese neue Bewertung führt zu einer neuen und besseren Reihung der Dokumente. Die Dokumente werden dann gemeinsam mit der Anfrage als erweiterter Kontext an den Generator, ein vortrainiertes Sprachmodell (LLM), übergeben. Dieser erzeugt basierend auf diesen Daten die finale

Antwort, wobei so externe Wissensquellen mit dem internen Wissen des Modells verbunden werden.

Umsetzbarkeit

Wann kann ein RAG-System verfügbar sein?

- Keine lange Vorlaufzeit nötig
- Zeitplan für einen On-Premise Piloten mit klar begrenzten Use Cases: ca. 6-18 Wochen
- Zeitplan für eine zentrale, integrierte Plattform mit RAG, mit klar definierter Benutzer:innen- und Rechteverteilung, mit Service Level Agreement (SLA) Fähigkeit: ca. 6-12 Monate

Souveränität

Ist ein RAG-System für sensitive Anwendungsfälle geeignet?

Für den Aufbau eines souveränen RAG-Systems in der öffentlichen Verwaltung ist ein hybrider Ansatz aus On-Premise-Hardware und Open-Source-Software ideal. Dies garantiert Datenhoheit und Unabhängigkeit von außereuropäischen Cloud-Anbietern und reduziert datenschutzrechtliche Fragen zum internationalen Datentransfer.

Ein strikter On-Premise Betrieb ist möglich, sofern das oder die gewählten LLMs lokal betrieben werden können und betreiberseitig Sicherheitsmaßnahmen gesetzt werden, um sicherzustellen, dass es keine Schwachstellen in der lokalen IT-Infrastruktur gibt und auch Angriffe auf das Modell selbst möglichst abgeblockt werden. Dadurch wird auch datenschutzrechtlichen Sicherheitsbedenken Rechnung getragen.

Die Kontrolle über Trainingsdaten des verwendeten Foundation Models ist beschränkt: Die Offenheit eines Modells ist getrennt zu prüfen: Modellgewichte, Lizenz, Trainingsdaten-Transparenz, Trainingscode, Datenprovenienz und Nutzungsrestriktionen. Offene Gewichte allein bedeuten keine volle Kontrolle über Trainingsdaten.

Betriebsmodelle

Für welche Betriebsmodelle ist ein RAG-System geeignet (isolierter vs. zentral geregelter Betrieb)?

Mit einem RAG-System können verschiedene Betriebsmodelle und Applikationen umgesetzt werden. Die Abwägung ob Cloud, On-Premise oder hybride Lösung hängt stark von der Useranzahl und der langfristigen Nutzungsstrategie ab. Nachfolgend finden Sie Kostenmodelle für eine On-Premise vs. Cloud-Lösung.

Tabelle 1 Realisierungsvariante LLM-Einbindung: Kostenfaktor je nach Betriebsmodell

Kostenfaktor	On-Premisse (eigene Hardware)	Souveräne Cloud (z.B.: STACKIT/OTC
Investition (CAPEX)	Hoch (ca. 40.000 € – 150.000 € initial)	Gering (Pay-as-you-go)
Laufende Kosten (OPEX)	Wartung, Strom, Kühlung, IT-Personal	Monatliche Gebühren
Skalierbarkeit	Begrenzt durch physische Hardware	Hoch und flexibel auf Knopfdruck
Wirtschaftlichkeit	Amortisation oft nach 12–24 Monaten	Günstiger für Prototypen & kleine Teams

Kompetenzen und Rollen

Welche Kompetenzen und Rollen sind für die Umsetzung bzw. den Betrieb eines RAG-Technologie-Stacks und die darauf basierenden Fach-Applikationen notwendig?

Tabelle 2 Realisierungsvariante LLM-Einbindung: Kompetenzen und Rollen

Bereich	Rolle	Kernkompetenzen & Aufgaben
KI-Entwicklung	AI/ML Engineer / Architekt	LLM-Auswahl, Embedding-Modelle, Orchestrierung (LangChain/LlamaIndex), Prompt Engineering.
Daten-Infrastruktur	Data & Search Engineer	Management von Vektordatenbanken, ETL-Prozesse, Chunking-Strategien, Hybrid Search (Vektor + Keyword).
IT-Betrieb	DevOps/LLMOps Engineer	Cloud-Infrastruktur (AWS/Azure), GPU-Provisionierung, Monitoring von Latenz, Kosten und Modell-Drift.
Qualitätssicherung	AI Evaluation Specialist	Messung von Metriken (Faithfulness, Relevance), Validierung gegen Halluzinationen, Testing.
Fachbereich	Domain Expert	Kuratierung der Wissensbasis (Ground Truth), fachliche Abnahme der Antworten, Definition von Fachbegriffen.

Bereich	Rolle	Kernkompetenzen & Aufgaben
Governance	Legal, Ethics & Compliance Expert	Daten- und Informationsverwendung (insbesondere DSGVO, DSG, IWG, IFG, Data Governance Act, Data Act, UrhG), Produkt-, Prozess- und Datensicherheit (insbesondere CRA, NIS 2-RL), Haftungsfragen (insbesondere AHG), unverbindliche, richtungsweisende Leitlinien und Verhaltenskodizes, ethische Standards, Prüfung von Zugriffsrechten auf Dokumentebene.
Sicherheit & Betrieb	Security / Service Owner	Zugriffskontrolle, Mandantentrennung, Prompt-Injection-Schutz, Audit-Logs, Lösch-/Re-Indexierungsprozesse, SLAs und Betriebsfreigaben.

Soft- und Hardwareausstattung

Welche Soft- und Hardwareausstattung ist für die Umsetzung bzw. für den Betrieb eines RAG-Technologiestacks und die darauf basierenden Fach-Applikationen notwendig?

Die folgende Tabelle gibt ein Beispiel für einen souveränen RAG-Software-Stack (Open Source & On-Premise).

Tabelle 3 Realisierungsvariante LLM-Einbindung: Soft- und Hardwareausstattung

Komponente	Technologie-Beispiele	Fokus & Besonderheiten
Large Language Models (LLMs)	Llama 3 / 4, Mistral / Mixtral, Teuken-7B, DiscoLM German	Lokale Ausführung mit offenen Lizenzen; Mistral-Varianten bieten starke europäische/deutsche Sprachoptimierung.
Embedding-Modelle	Qwen3-Embedding, BGE-M3, multilingual-e5-large-instruct, jina-embeddings-v3	Bestimmen die Retrieval-Qualität; mehrsprachige Modelle für deutschsprachige Verwaltungstexte mit englischen Fachbegriffen empfohlen. Auswahl task-spezifisch validieren (MTEB-Benchmark als Orientierung).
RAG-Frameworks	LangChain, LlamaIndex, Haystack	Orchestrierung von Datenflüssen; Haystack als modulare Enterprise-Lösung aus Europa.
Vektordatenbanken	Qdrant, Weaviate, ChromaDB	Speicherung von Embeddings; Qdrant & Weaviate (Berlin) sind führend für den On-Premise-Betrieb.
Inferenz-Server	vLLM, TGI (Text Generation Inference), Ollama	Bereitstellung auf eigener Hardware; vLLM/TGI für Hochleistung, Ollama für schnelles Prototyping.

Tabelle 4 Realisierungsvariante LLM-Einbindung: Hardware-Anforderungen für den lokalen Betrieb eines RAG. Ein wesentlicher kritischer Faktor ist der Grafikspeicher (VRAM) der GPU

Komponente	Hardware-Fokus	Mindestanforderung (Pilot/Test)	Empfehlung (Produktivbetrieb)
LLMs (z. B. Llama 3 8B / Mistral 7B)	GPU (VRAM)	1x NVIDIA RTX 3060/4060 (12GB VRAM)	1x NVIDIA A100 / H100 (80GB) oder L40S (48GB)
Große LLMs (z. B. Llama 3 70B / Mixtral)	GPU (VRAM)	2x RTX 3090/4090 (je 24GB) mit Quantisierung	Multi-GPU Cluster (z. B. 4x - 8x A100/H100)
Vektordatenbanken (Qdrant, Weaviate)	RAM & SSD	16GB RAM, schnelle NVMe SSD	64GB - 128GB RAM (High-Speed RAM für In-Memory Indexing)
RAG-Frameworks & Inferenz (vLLM / TGI)	CPU & Durchsatz	Aktuelle Multi-Core CPU (min. 8 Kerne)	Server-CPUs (AMD EPYC / Intel Xeon) + High-Speed Interconnect (NVLink)
Embedding-Modelle (lokal)	GPU / CPU	Läuft oft parallel auf der LLM-GPU (benötigt ca. 2-4GB VRAM)	Dedizierte kleine GPU-Ressource für parallele Indexierung

Die Chunking-Strategie — wie Dokumente vor der Indexierung zerlegt werden — ist neben der Wahl des Embedding-Modells der wichtigste Qualitätsfaktor eines RAG-Systems. Für Verwaltungsdokumente sind strukturierte Verfahren (z.B. semantisches Chunking, hierarchisches Chunking nach Dokumentstruktur, oder spezialisierte Verfahren für lange PDFs und Tabellen) gegenüber naivem Fixed-Size-Chunking deutlich überlegen.

Wichtige Faustregeln für die Planung sind:

- Modellgröße & VRAM: Ein 7B-Parameter Modell benötigt (unkomprimiert) ca. 14GB VRAM. Durch Quantisierung (4-Bit) lässt es sich auf ca. 5-6GB reduzieren, was den Betrieb auf Consumer-Hardware ermöglicht.
- Latenz vs. Durchsatz: Für einen flüssigen Betrieb (Tokens pro Sekunde) ist die Speicherbandbreite der GPU wichtiger als die reine Rechenpower.
- Skalierung: Vektordatenbanken skalieren primär über den Arbeitsspeicher (RAM). Je mehr Dokumente, desto mehr RAM wird benötigt, um die Suchgeschwindigkeit hochzuhalten.

Vor produktiver Beschaffung ist ein Lasttest mit repräsentativen Dokumenten, Prompts, Nutzer:innenzahlen und aktivierter Zugriffskontrolle durchzuführen. Die Hardware gilt erst als geeignet, wenn Retrieval, Reranking, Antwortgenerierung, Logging und Monitoring die definierten SLAs erfüllen.

Rechtliche Rahmenbedingungen

Datenschutz

Die DSGVO verfolgt einen technologieneutralen Ansatz. Für die Anwendung ist es nicht maßgeblich, durch welches technische (Hilfs-)Mittel die Datenverarbeitung erfolgt. Daher ist es unerheblich, ob die Verarbeitung durch klassische IT-Systeme oder durch KI-Systeme erfolgt. Entscheidend ist lediglich, ob (i) personenbezogene Daten verarbeitet werden und (ii) sich durch den Einsatz von KI-Systemen Verarbeitungsprozesse ändern oder ergänzen und damit eine neue Beurteilung, insbesondere der Rechtsgrundlage, erforderlich ist. Während ersteres durch die EuGH-Entscheidung C-413/23 (curia.europa.eu/juris/document/document.jsf?docid=303863&doclang=DE) zum relativen Anonymitätsbezug die Datenübermittlung zwischen Verantwortliche/Auftragsverarbeiter deutlich vereinfacht, besteht zur Auswirkung von KI-Systemen auf die Verarbeitungsprozesse noch kaum Rechtsprechung nationaler Behörden und Gerichte oder des EuGH. Aus den spärlichen FAQs, Orientierungshilfen, Stellungnahmen und Leitlinien nationaler Datenschutzbehörden ist jedoch ableitbar, dass der Einsatz von RAG-Systemen einer Rechtsgrundlage bedarf.

So werden unter anderem personenbezogene Daten aus den Referenzdokumenten durch das Embedding-Modell verarbeitet und in einer Vektordatenbank gespeichert. Im Rahmen der Nutzung von generativen KI-Systemen und RAG-Methoden kommt es unzweifelhaft zu einer veränderten Datenverarbeitung (vgl obige technische Ausführungen). Fraglich ist jedoch, inwieweit sich der Verarbeitungszweck verändert bzw. eine Zweckkompatibilität nach Art 6 Abs 4 DSGVO vorliegt. Das ist je Anwendungsfall mit Blick auf die Differenzierung von Hoheits- und Privatwirtschaftsverwaltung zu prüfen, um auch eine allfällige Deckung mit gesetzlichen Rechtsgrundlagen zur Datenverarbeitung zu identifizieren. Behörden, die eine Verarbeitung in Erfüllung ihrer Aufgaben vornehmen, können sich schließlich nicht auf überwiegende berechtigte Interesse nach Art 6 Abs 1 lit f DSGVO berufen. Sie müssen vielmehr auf eine Rechtsgrundlage nach Art 6 Abs 1 lit c oder lit e DSGVO stützen. Im Zusammenhang mit der Rechtmäßigkeitsprüfung ist ein besonderes Augenmerk auf die automatisierte Entscheidung im Einzelfall nach Art 22 DSGVO zu legen. Art 22 Abs 1 DSGVO statuiert

kurz zusammengefasst das Recht jeder betroffenen Person, keiner ausschließlich auf automatisierter Verarbeitung – einschließlich Profiling – beruhenden Entscheidung unterworfen zu werden, die ihr gegenüber rechtliche Wirkung entfaltet oder sie in ähnlicher Weise erheblich beeinträchtigt. Das ist zwar unabhängig von RAG-Systemen beim Einsatz von KI-Systemen immer zu berücksichtigen, doch können RAG-Systeme auch risikomitigierende Maßnahmen darstellen und die Rechte der Betroffenen schützen. Für die Anwendbarkeit dieser Norm müssen kumulativ drei Voraussetzungen erfüllt sein:

- das Vorliegen einer Entscheidung;
- deren ausschließliche Automatisierung ohne menschliches Eingreifen; und
- eine rechtliche oder vergleichbar erhebliche Auswirkung auf die betroffene Person.

Die Rechtsprechung des EuGH, insbesondere die Entscheidung SCHUFA (C-634/21, eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:62021CJ0634), hat die Voraussetzungen weit ausgelegt. Bereits ein automatisiert erzeugter Score kann eine relevante Entscheidung darstellen, wenn er für eine nachgelagerte menschliche Entscheidung maßgeblich ist. Damit wird die bislang klare Trennlinie zwischen Entscheidungsunterstützung und automatisierter Entscheidungsfindung verwischt.

Als weitere Kernfrage steht eine Datenverarbeitung in Drittstaaten im Raum, wenn KI-Modelle Dritter angebunden und Daten abgerufen werden (z.B. über API-Schnittstellen). Jede Übermittlung personenbezogener Daten beim Training, Testen und Betrieb der KI-Funktion in Länder außerhalb des EWR muss nach Art 44 ff DSGVO gerechtfertigt sein. Dies kann in erster Linie auf der Grundlage eines Angemessenheitsbeschlusses oder durch geeignete Garantien gerechtfertigt werden. Wird der Datentransfer mangels Angemessenheitsbeschluss auf Standardvertragsklauseln gestützt, ist dabei ein Transfer Impact Assessment durchzuführen und die allfällig notwendige Vereinbarung zusätzlicher Garantien erforderlich. Erfolgt die Verarbeitung des Bundes-LLM beispielsweise On-Premise im BRZ und ist auch ein Datenzugriff aus Drittstaaten ausgeschlossen, sind keine Vorgaben an den Datentransfer zu beachten.

Zu beachten ist weiters, dass eine initiale rechtswidrige Datenverarbeitung nach aktueller nationaler Rechtsprechung zur Unzulässigkeit der weiteren, nachgelagerten Verarbeitungen führt. Anders sieht das aber die Deutsche Bundesbeauftragte für den Datenschutz und die Informationsfreiheit ("BfDI")[5]. Laut der Stellungnahme Handreichung "KI in Behörden"

[5] bfdi.bund.de/SharedDocs/Downloads/DE/DokumenteBfDI/Dokumente-allg/2025/Handreichung-KI.html

vom 22.12.2025 strahlt eine etwaige Rechtswidrigkeit früherer Verarbeitungsvorgänge nicht pauschal auf die spätere Verarbeitung aus. Demzufolge kann auch ein rechtswidrig entwickeltes KI-Modell rechtmäßig eingesetzt werden, sofern der Verantwortliche seinen Nachforschungspflichten vor dem Einsatz nachkommt. Generelle Klärung, ob z.B. im KI-Kontext aufgrund der großen Abhängigkeit etwas Abweichendes gelten soll, wird erst der EuGH bringen. Bis dahin besteht hier aber ein erhebliches Risiko.

Freilich sind auch die sonstigen Datenschutzpflichten zu berücksichtigen (z.B. Ergänzung des Verarbeitungsverzeichnisses, Durchführung einer Datenschutzfolgenabschätzung, Informationspflichten, Wahrung der Betroffenenrechte), auf die im Folgenden nicht näher eingegangen wird. Zu beachten ist, dass sich die oben genannten Datenschutzfragen unabhängig davon ergeben, ob ein kleines oder großes LLM eingesetzt werden.

Urheberrecht

Bei Verwendung von RAG-Systemen werden regelmäßig Werke bei Data Scraping und Nutzung des RAG-Ansatzes urheberrechtlich relevant verwertet, insbesondere inputseitig vervielfältigt und zur Generierung von Outputs ggfs. bearbeitet:

Ein Sprachmodell, dass Inhalte im KI-Wissenskorpus umgangssprachlich "ausliest" bzw. sucht und die Daten an den Generator übergibt, vervielfältigt diese Inhalte zumindest zu Beginn der Inferenz flüchtig, indem es Daten zur weiteren Verarbeitung in Tokens transformiert. Obwohl technisch gesehen mehrere Vervielfältigungen bei Nutzung eines RAG-Systems vorliegen, sind die Handlungen Teil eines einzigen vorübergehenden Vorgangs und können daher zusammen betrachtet werden. Erstellt eine generative KI-Anwendung einen synthetischen Text, der den wesentlichen schöpferischen Elementen des Originaltextes gleicht, kann eine Vervielfältigung angenommen werden. Die Nutzungshandlungen sind nur zulässig, wenn eine freie Werknutzung (z.B. § 7 UrhG für amtliche Werke, § 42h UrhG für Text- und Data-Mining, § 41 UrhG für flüchtige Vervielfältigung) oder eine Rechteeinräumung (Lizenz) besteht bzw. ableitbar ist.

Für den Aufbau der Wissensdaten ist es essentiell die Datenquellen und Informationen vorab zu definieren und auf die Einhaltung des Urheberrechts hin zu prüfen. Dadurch können sich gewisse Grenzen für den Anwendungsfall selbst ergeben, der Einsatz von RAG-Systemen ist jedoch generell denkbar.

Sonstige Datennutzung

Es ist weiters zu hinterfragen, welche Informationsquellen für den Aufbau der Wissensdatenbank sonst verwendet werden dürfen. Dafür gibt es zahlreiche Rechtsakte, die die Veröffentlichung und Nutzung von öffentlichen Daten regulieren (Stichwort Open Data Governance):

- data.gv.at bietet eine große, klare und risikofreie Datenquelle (CC BY 4.0);
- IFG-Informationen sind hingegen nicht automatisch frei nutzbar – fehlende Lizenzangaben schaffen Rechtsunsicherheit;
- Der DGA erleichtert die Weitergabe staatlicher Daten, verpflichtet aber nicht zur Veröffentlichung.

Der Data Act ermöglicht Behörden nur in Ausnahmesituationen Zugriff auf Unternehmensdaten – für das Bundes-LLM daher kaum nutzbar.

Produktsicherheit

Die KI-Verordnung (KI-VO) ist bei Umsetzung dieser Variante anwendbar. Die jeweilige öffentliche Stelle kann - je nach Ausgestaltung - Betreiber oder Anbieter des KI-Systems sein. Nutzt die öffentliche Stelle ein bestehendes KI-System und beschränkt deren Nutzung lediglich um interne Daten zur optimierten Ausgabe, handelt es als Betreiber. Ebenso als Betreiber agieren die öffentlichen Stellen, die ein KI-System einer anderen Einrichtung lizensieren und in eigener Verantwortung betreiben. Lizensiert die öffentliche Stelle hingegen ein Modell und bindet dieses in die eigene Systemlandschaft ein, handelt es als Anbieter des KI-Systems.

Die Risikoklassifizierung nach der KI-VO ist aus der Anwenderperspektive für die jeweiligen Einzelfälle zu prüfen. Diesbezüglich können keine allgemeinen Aussagen getroffen werden.

Die Rolle und das Risiko bestimmen die anwendbaren Produktsicherheitspflichten nach der KI-VO. Bei Lizensierung von KI-Systemen oder KI-Modellen ist eine vertragliche Absicherung für KI-spezifische IT-Projektrisiken ratsam (z.B. weite No-Training-Klausel, Zusicherung zur Beachtung verbotener KI-Praktiken, datenschutz- und urheberrechtliche Rechtmäßigkeit des KI-Modells, ergänzende Nebenleistungen zur Umsetzung der Pflichten als Downstream-Provider).

Der Strafrahmen der KI-VO beträgt maximal bis zu EUR 35 Mio. oder 7 % des gesamten weltweit erzielten Jahresumsatzes (je nachdem was höher ist). Eine Behördenausnahme ist in der KI-VO nicht verankert, könnte jedoch mit dem nationalen Umsetzungsgesetz erfolgen (noch kein Gesetzesentwurf veröffentlicht, Stand April 2026).

Cybersecurity

Cybersicherheit ist ein zentraler Faktor für die Widerstandsfähigkeit, Zuverlässigkeit und Sicherheit von KI-Systemen. Für das Bundes-LLM gilt dies in besonderem Maße, da es sensible Verwaltungsdaten verarbeiten, behördeninterne Abläufe unterstützen und potenziell in kritischen Sektoren eingesetzt werden kann. Ein angemessenes Cybersicherheitsniveau muss während des gesamten Lebenszyklus gewährleistet werden – von der Architekturplanung über das Training und Finetuning bis hin zum Betrieb. Dies entspricht auch den Anforderungen der KI-VO (Art 15 KI-VO), die für Hochrisiko-KI-Systeme spezifische Sicherheitsmaßnahmen vorgibt, etwa zum Schutz vor Datenvergiftung, Adversarial Attacks oder Manipulationen der Modelle. Neben der KI-VO schreibt die NIS-2-RL organisatorische Cybersicherheitsmaßnahmen (z.B. Risikomanagement, Meldungen von Vorfällen) vor. Allerdings umfasst NIS-2 keine unmittelbaren technischen Sicherheitsvorgaben. Diese Lücke schließt der CRA (gilt ab dem 11.12.2027), die Cyber- und IT-Sicherheitsstandards vorschreibt. NIS-2 und CRA stehen nicht in Konkurrenz zueinander, sondern ergänzen sich. Zudem fördern die im CRA enthaltenen Anforderungen zur IT-Sicherheit mittelbar auch den Datenschutz (ErwGr 32 CRA). Ergänzt werden die Sicherheitsmaßnahmen um Art 32 DSGVO zu technische und organisatorische Datensicherheitsmaßnahmen – beispielsweise im Hinblick auf Zugriffskontrollen, Verschlüsselung, Protokollierung etc.

Modelloptimierung: Finetuning und Modelladaption

Wir unterscheiden zwischen Finetuning-Ansätzen und Wissensdestillation (Knowledge Distillation). Beim Finetuning geht es um die Anpassung eines vortrainierten LLMs durch Training mit domänenspezifischen Daten (z. B. Verwaltungsanweisungen). Bei der Wissensdestillation werden kleinere, deploybare Modelle erstellt, die Eigenschaften bzw. Kapazitäten von größeren 'Teacher'-Modellen erben.

Fine-Tuning – Domänen Adaption eines existierenden Foundation Models

Das Verständnis der theoretischen Grundlagen von Finetuning ist essentiell für die Auswahl geeigneter Methoden und deren optimaler Konfiguration. Während im klassischen Finetuning alle trainierbaren Parameter verändert werden können, friert LoRA (Low-Rank Adaptation; Hu et al., 2021) die Gewichte des Basismodells ein und trainiert stattdessen kleine, niedrigrangige Adapter-Matrizen, die parallel zu den eingefrorenen Gewichten wirken. Die Anzahl der trainierbaren Parameter sinkt dadurch typischerweise auf unter 1 % des Originalmodells; LoRA ist entsprechend deutlich weniger ressourcenintensiv und für die Domänenanpassung gut geeignet. Für einen Vergleich klassisches Finetuning versus LoRA siehe die nachfolgende Tabelle.

Tabelle 5 Vergleich Klassisches Fine-Tuning versus Low Rank Adaptation

Feature	Klassisches Fine-Tuning	LoRA
Trainierbare Parameter	Alle (100 %)	Sehr wenige (< 1 %)
Rechenaufwand	Sehr hoch	Sehr gering
Speicherbedarf	Gigabyte/Terabyte (viele Checkpoints)	Megabyte (kleine Adapter)
"Vergessen" von Wissen	Hohes Risiko	Gering

LoRA eignet sich besonders gut für Anwendungen, bei denen das Vergessen von Pretraining-Wissen minimiert werden soll, z.B. Anpassung an österreichisches Verwaltungsdeutsch bei Erhalt allgemeiner Sprachfähigkeiten. Verschiedene LoRA- Erweiterungen und Varianten haben unterschiedliche Vorteile. Für Behörden mit vielfältigen Anwendungsfällen sind Methoden zur Kombination mehrerer spezialisierter Adapter von besonderer Relevanz, z.B. separate LoRA-Adapter für Rechtssprache, Bürgerkommunikation und interne Verwaltung können bei Bedarf dynamisch kombiniert werden.

Wissensdestillation

Wissensdestillation erstellt kleinere, deploybare Modelle, die die Leistungsfähigkeit von größeren 'Teacher'-Modellen erben. Das ist sowohl für die Modellentwicklung als auch für einen ressourceneffizienten Betrieb von Bedeutung. Der Aufwand für Wissensdestillation hängt stark vom Verfahren ab: Logit-basierte Distillation auf einem aufgabenspezifischen Korpus (einige Millionen Tokens) mit einem 8B Teacher zu einem 1B Student kann auf einer

A100 in wenigen Stunden bis zu einem Tag abgeschlossen werden. Capability-erhaltende Distillation, die einen breiteren Anteil der Teacher-Fähigkeiten überträgt, erfordert in der Regel mehrere Milliarden Trainings-Tokens und entsprechend Tage bis Wochen auf Multi-GPU-Setups.

Siehe Tabelle 6 für den Learner-zu-Teacher-Leistungserhalt in Abhängigkeit von der Aufgabe. Die Tabelle zeigt auch, dass unterschiedliche Aufgaben (Task-Type) unterschiedliche Student-Modell-Größen erfordern um einen guten Leistungserhalt zu gewährleisten.

Tabelle 6 Performanz von Student-Modellen je nach Modellgröße und Aufgabe

Task-Typ	Student-Größe	Performance Retention
Klassifikation, Kategorisierung	60-100M	95-99%
Question Answering	100-400M	85-96%
Komplexe Generierung	1-3B	75-95%

Anmerkung: M (EN: Million) steht für Million, B (EN: Billion) für Milliarde. Die Zahlen sind Orientierungswerte basierend auf veröffentlichten Distillationsstudien (u.a. DistilBERT, TinyBERT, MiniLM, MiniLLM) und variieren erheblich je nach Methode (Logit-, Feature-, Sequence-Level), Trainingsdaten und Teacher-Student-Architektur-Distanz. Für den Bundes-LLM-Kontext sind eigene Messungen am Zieltask erforderlich.

Ein anderes Beispiel für Wissensdestillation beschreibt die im Österreichischen Cluster of Excellence „Bilateral Artificial Intelligence“ entstandene Arbeit, in der die theoretischen Grundlagen für die (möglichst) verlustfreie Wissensdestillation von Transformer zu xLSTM Architektur entwickelt wurden und die Implementierung und Evaluierung eines Destillationsbeispiel von Llama 3.1.-8B (Teacher) zu einem gleich mächtigen aber verarbeitungseffizienteren Student-Modell (xLSTM-distill-8B) beschrieben sind (Hauzenberger et al., 2026).

Umsetzbarkeit

Wann kann das adaptierte Modell verfügbar sein?

- Zeitrahmen: ca. 12-18 Monate
- Wahl des Basis Modells, das adaptiert wird, beeinflusst die Souveränität:
  - open-source vs. open-weight Modelle

  - europäische vs. nicht-europäische Modelle
- Wissensdestillation und Modelladaption sind dynamische Bereiche in der Grundlagenforschung, daher ist der Entwicklungserfolg von der Einbindung entsprechender Expert:innen abhängig
- Entwicklungszeit von konkreten Applikationen kommt noch dazu
- Die adaptierten Modelle eignen sich für spezialisierte Applikationsworkflows

Empfohlener Workflow für die Modelladaption

1. Fine-tune Teacher Modell (z.B. Mistral-7B) mit LoRA auf Domain-Daten
2. Verwende LoRA-adaptierten Teacher für Knowledge Distillation
3. Destilliere zu kleinerem Student (z.B. 1-3B Parameter)
4. Optional: Wende LoRA auf Student für zusätzliches Fine-Tuning an

Souveränität

Ist ein adaptiertes Modell für sensitive Anwendungsfälle geeignet?

- Ja, da strikter On-Premise Betrieb möglich ist.

Hinsichtlich Kontrolle über Trainingsdaten gilt für das Teacher-Modell dasselbe wie für Foundation Modelle: Offenheit ist getrennt nach Gewichten, Lizenz, Trainingsdaten-Transparenz, Datenprovenienz und Nutzungsrestriktionen zu prüfen. Offene Gewichte bedeuten keine volle Kontrolle über Trainingsdaten. Die Auswahl der Trainingsdaten für das Finetuning bzw. für LoRA liegt in der Kontrolle jener, die das Finetuning durchführen.

Betriebsmodelle

Für welche Betriebsmodelle ist ein adaptiertes Modell geeignet (isolierter vs. zentral geregelter Betrieb)?

- Wie mit einem Foundation Model können auch mit einem adaptierten Modell verschiedenen Varianten umgesetzt werden.
- Für den Fall eines isolierten Betriebs muss auch das adaptierte Modell selbst zentral gewartet werden.

Kompetenzen und Rollen

Tabelle 7 gibt eine Übersicht über benötigte Kompetenzen für das Fine-Tuning mit LoRA. Für lokales Finetuning werden darüber hinaus Hardwaremanagement- und Ressourcenoptimierungskompetenzen gebraucht (siehe Tabelle 8).

Tabelle 7 Benötigte Kompetenzen für das Fine-Tuning mit LoRA

Kategorie	Kompetenz	Beschreibung
Programmierung	Python	Sicherer Umgang mit Skripten, APIs und Umgebungen (z.B. venv, Conda).
	Hugging Face Stack	Anwendung von Transformern, PEFT (LoRA-Logik) und Datasets.
	Deep Learning Frameworks	Grundverständnis von PyTorch (Tensors, Optimizer, Loss-Functions).
Daten (Data Engineering)	Data Curation	Erstellung und Bereinigung hochwertiger Trainingsdaten (z.B. Instruction-Tuning).
	Tokenisierung	Verständnis von Token-Limits und sprach- bzw. modellspezifischen Tokenizern.
	Formatierung	Transformation von Rohdaten in Formate wie z.B. jsonl oder Parquet.
Theorie & Mathematik	LoRA-Parameter	Wissen über den Einfluss von Hyperparametern wie Rank und Alpha sowie Ziel-Module.
	Modell-Training	Verständnis von Overfitting, Learning Rates und Batch-Größen.
	Quantisierung (QLoRA)	Kenntnisse in 4-Bit/8-Bit Kompression zur VRAM-Ersparnis. (Dettmers et al., 2023)
Infrastruktur	GPU-Management	Überwachung von VRAM (CUDA) und effiziente Speichernutzung.
	Cloud & Tools	Nutzung von Plattformen (Colab, RunPod) und Logging (Weights & Biases).

Tabelle 8 Hardwaremanagement- und der Ressourcenoptimierungskompetenzen für lokales Fine-Tuning

Bereich	Kompetenz	Praktische Anwendung
System-Ebene	Linux / WSL2	Sicherer Umgang mit Terminal, Dateiberechtigungen und Pfaden.
Treiber	CUDA-Setup	Installation von NVIDIA-Treibern, CUDA Toolkit und cuDNN.
Optimierung	VRAM-Management	Konfiguration von Gradient Checkpointing und bitsandbytes.
Monitoring	Hardware-Metriken	Überwachung von GPU-Temperatur und Speicher (z. B. nvitop).
Tooling	Local Trainers	Einrichtung von Frameworks wie Axolotl, Unsloth oder QLoRA.

Soft- und Hardwareausstattung

Im Folgenden wird die Software-Landschaft für das LoRA-Fine-Tuning umrissen (Tabelle 9 bis Tabelle 12): Es werden Beispiele für Frameworks, technische Basisbibliotheken, Infrastruktursoftware und Tools für das Tracking des Machinelearningfortschritts aufgelistet. Ready-to-use Frameworks bündeln die für das Finetuning erforderlichen komplexen Prozesse.

Für die Abschätzung des Hardwarebedarfs für LoRA-basiertes Finetuning ist der VRAM-Bedarf ausschlaggebend. Dieser setzt sich aus den Modellgewichten (ca. 0,5 GB pro 1 Mrd. Parameter bei 4-Bit-Quantisierung) plus dem Trainings-Overhead (Aktivierungen, Gradienten und Optimizer-States) zusammen. Siehe Tabelle 13 für Beispiele bezüglich des Hardwarebedarfs (VRAM x Modellgröße) für LoRA-basiertes Fine-Tuning.

Tabelle 9 Software für Fine-Tuning

Software-Typ	Name	Hauptvorteil	Fokus-Zielgruppe
All-in-One	Unsloth	Extrem schnell (2x), spart 70% VRAM.	Einsteiger & Profis (lokal/Colab)
Config-basiert	Axolotl	Steuerung via YAML-Datei (kaum Code).	Power-User & Cloud-Training
Web-UI	Oobabooga	Grafische Oberfläche (Klick-Menüs).	Nutzer:innen ohne Programmierwunsch

Tabelle 10 Technische Basis-Bibliotheken (aus dem "Hugging Face Stack")

Bibliothek	Funktion	Verwendung
PEFT	LoRA-Logik	Enthält die Algorithmen für die Adapter-Schichten.
transformers	Modell-Handling	Lädt das Teacher-Modell und den Tokenizer.
bitsandbytes	Quantisierung	Wrapper für CUDA Funktionen; ermöglicht QLoRA (4-Bit), um VRAM zu sparen.
TRL	Trainer-Interface	Bietet den SFT (Supervised Fine Tuning) Trainer für einfaches Instruktions-Training.
Accelerate	Hardware-Optimierung	Verteilt das Training effizient auf die GPU(s).

Tabelle 11 Infrastruktur-Software

Komponente	Zweck	Anmerkung
Python 3.10+	Programmiersprache	Basis für alle KI-Bibliotheken.
PyTorch	Deep Learning Backend	Muss exakt zur CUDA-Version passen.
CUDA Toolkit	GPU-Beschleunigung	NVIDIA-Schnittstelle für Berechnungen.

Tabelle 12 Software für das Tracking von Fortschritt und Fehlern im Machinelearningprozess (Loss-Kurven)

Tool	Typ	Hauptvorteil	Besonderheit
Weights & Biases (W&B)	Cloud / Hybrid	Industriestandard; extrem einfach zu integrieren.	Beste Visualisierung und Team-Kollaboration.
TensorBoard	Lokal / Open Source	Kostenlos; keine Cloud-Anbindung nötig.	Standard-Wahl für lokale Projekte; sehr performant bei einfachen Kurven.
MLflow	Open Source	Deckt den gesamten ML-Lebenszyklus ab.	Stark im Management von Modellversionen und Artefakten.
ClearML	Open Source / Enterprise	Automatisiertes Orchestrieren von Trainings-Jobs.	Kombiniert Tracking mit Ressourcen-Management.
Neptune.ai	Cloud	Fokus auf Skalierbarkeit für große Teams.	Bekannt für exzellenten Support und Übersichtlichkeit.

Tabelle 13 Hardwarebedarf für LoRA-basiertes Fine-Tuning; * B steht für EN Billion (= DE Milliarde)

Verfügbarer VRAM	Max. Modellgröße*	Beispiel-Modelle
8 GB	bis 3B - 7B	Phi-3 Mini, Mistral 7B (sehr knapp)
12 GB	bis 8B - 11B	Llama 3.1 (8B), Mistral 7B, Gemma 2 (9B)
16 GB	bis 14B	Mistral NeMo (12B), Qwen 2.5 (14B)
24 GB	bis 32B	Llama 3.1 (8B) mit hohem Kontext oder Qwen (32B)
48 GB (2x 24GB)	bis 70B	Llama 3.1 (70B) – knapp, benötigt Optimierung
80 GB (A100/H100)	bis 70B+	Llama 3.1 (70B) komfortabel mit viel Kontext

Rechtliche Rahmenbedingungen

Die im Zusammenhang mit dem RAG-System erläuterten Regularien sind auch bei dieser Umsetzungsvariante relevant. Daher gehen wir im Folgenden nur auf die Abweichungen und Besonderheiten ein:

Datenschutz

Finetuning und Model Adaption begründen einen gänzlich neuen Verarbeitungszweck und Verarbeitungsprozesse. Dafür ist, soweit ersichtlich, keine für die Hoheitsverwaltung relevante, gesetzliche Grundlage vorhanden. Das stellt hiermit (derzeit) eine echte Grenze dar. Im Rahmen der Privatwirtschaftsverwaltung kommen hingegen - je nach verwendeten Daten, ergriffenen risikomitigierenden Maßnahmen, Transparenz und Interessen der Betroffenen - auch überwiegende berechtigte Interessen nach Art 6 Abs 1 lit f DSGVO in Frage. Weiters besteht bei dieser Umsetzungsvariante darauf zu achten, dass keine personenbezogenen Daten im KI-Modell memoriert werden.

Produktsicherheit - KI-Regulierung

GPAI-Modelle können weiter modifiziert werden (ErwGr 97 AI Act[6]). Fine-Tuning ist laut AI Office eine Möglichkeit der Modifizierung. Die Leitlinien zum Umfang der Verpflichtungen für Anbieter von KI-Modellen mit allgemeinem Verwendungszweck stellen klar, dass eine nachträgliche Modifikation dazu führen kann, dass aus einem einfachen AI-Modell ein GPAI-Modell und aus einem GPAI-Modell ein GPAI-Modell mit systemischem Risiko werden kann.

[6] eur-lex.europa.eu/eli/reg/2024/1689/oj, eur-lex.europa.eu/eli/reg/2024/2847/oj

Dementsprechend können nachgelagerte Akteure, die ein bestehendes KI-Modell ändern, nachgelagerter Anbieter des neuen GPAI-Modells werden (Downstream-Provider gemäß Art 3 Z 68 KI-VO). Die entsprechende öffentliche Stelle muss demnach zusätzlich die Anbieterpflichten für KI-Modelle mit allgemeinem Verwendungszweck nach Art 51 ff KI-VO einhalten. Dabei ist ein besonderes Augenmerk auf den Umfang der Modellanpassung zu legen, zumal die öffentliche Stelle auch Anbieter eines KI-Modells mit allgemeinem Verwendungszweck und systemischen Risiken werden, wodurch sich der Umsetzungsbedarf erweitert.

Urheberrecht

Im Rahmen der Modellanpassung kommt der freien Werknutzung des Text- und Data Mining eine größere Bedeutung zu als bei RAG-Systemen, weil es hier auch auf die Erkennung von Mustern und Korrelationen ankommt. Schließlich ist bei Verwendung von urheberrechtlich geschützten Werken darauf zu achten, dass keine Werke im Modell memoriert werden.

Foundation Modelle

Was sind Foundation Modelle?

Foundation Modelle sind Large Language Models (LLMs) die von Grund auf auf sehr großen Datenmengen trainiert werden.

Wie werden sie trainiert?

Foundation-Modell-Training erfolgt in 5 Hauptschritten (Phase 0 – Phase 4).

Phase 0: Daten & Design

Ziele, Use-Cases, Qualitätskriterien und Metriken definieren; Daten sammeln, bereinigen, filtern und aufbereiten; Modellarchitektur und Hardware planen. Legt das Fundament für alle nachfolgenden Phasen. Hier werden kritische Entscheidungen über Architektur, Tokenizer und Datensammlung getroffen, die den gesamten Trainingsprozess bestimmen.

Tokenizerdesign muss auf Sprache abgestimmt sein, z.B.: Komposita im Deutschen werden als ein zusammenhängendes Wort geschrieben, während im Englischen Komposita meist aus durch Leerzeichen getrennten Einzelwörtern bestehen (z.B. DE: *Interessensgemeinschaft* vs. EN: *special interest group*).

Tabelle 14 Verhältnis Modellgröße zu Trainingsdaten; B (En: Billion) steht für Milliarde; T (En: Trillion) steht für Billion

Modellgröße Parameter	**Chinchilla-Optimal (Hoffmann et al., 2022)**	**Moderne Praxis (Overtrained): 10-100x Chinchilla-Empfehlung**
7-9B	140B Tokens	• 15T Tokens (Apertus 8B) • 6T Tokens (Teuken 7B v0.6)
13-22B	260B Tokens	4T Tokens (EuroLLM 22B)
70B+	1.4T Tokens	• 15T Tokens (Apertus 70B) • ~40T Tokens (Llama 4 Scout 109B gesamt, 17B aktiv)

Bei MoE-Modellen wie Llama 4 Scout dürfen aktive Parameter nicht mit Speicherbedarf oder Deployment-Komplexität gleichgesetzt werden; dafür sind auch Gesamtparameter und Serving-Architektur relevant.

Phase 1: Pretraining

Großskaliges selbstüberwachtes Training auf vielen, überwiegend unbeaufsichtigten Daten, um allgemeine Sprach- und Weltrepräsentationen zu lernen. Das ist die rechenintensivste Phase und erfordert sorgfältige Planung bezüglich Scaling Laws, Trainingsstabilität und verteilter Infrastruktur.

Phase 2: Enhancement

Veredelung des Basismodells durch Supervised Fine-Tuning, RLHF/RLAIF, Tool-Use-Training und andere Verfahren zur gezielten Fähigkeitsverbesserung. Midtraining für Domain-Adaptation und Long-Context Extension, um das Basismodell für spezifische Anwendungsfälle zu optimieren.

Phase 3: Post-Training
Umfassende Evaluation mit Benchmarks, internen Tests und Red-Teaming; anschließende Anpassung von Gewichten, Adaptern und System-Prompts. Post-Training transformiert das Basismodell in einen hilfreichen, harmlosen und ehrlichen Assistenten. Es gibt mehrere Pfade: den Instruct Path für Standard-Assistenten, den Thinking Path für Reasoning-Modelle, den Agentic Path für Tool-Use und den RL Zero Path für pure RL-basierte Reasoning-Entwicklung.

Phase 4: Safety, Optimierung & Deployment
Implementierung von Safety-Mechanismen, Performance-Optimierung (z.B. Quantisierung), Monitoring sowie kontinuierlicher Verbesserungs- und Retraining-Prozess im Betrieb. Das ist die finale Phase umfasst kontinuierliche Evaluation, Safety Alignment, optionale Distillation und Quantization sowie das eigentliche Deployment.

Des Weiteren ist zu beachten, dass LLMs regelmäßig weitertrainiert werden müssen, um am aktuellen Wissenstand zu bleiben. Diese Aufgabe ist selbst wieder ressourcenintensiv und bedarf Expertise in LLM-Training.

Umsetzbarkeit
Wann kann ein nationales Foundation Modell produktiv verfügbar sein?

- Zeitrahmen: >= 24 Monate mit einem erfahrenen Team.
  - Die Entwicklungszeit für das erste verfügbare Apertus Model: ca. 2 Jahre
  - Geplante Entwicklungszeit für GPT-NL: 2,5 Jahre
  - Die Entwicklungszeit für konkrete Applikationen kommt noch dazu

Ein national trainiertes Foundation Model führt zu maximaler Souveränität und Kontrolle, und zu langfristiger Unabhängigkeit von externen Anbietern. Es hat in seiner Entwicklung aber eine lange Vorlaufzeit und sehr hohe Kosten. Mit dem Trainieren eines Foundation Models ist es nicht getan, denn es muss betrieben, gewartet und regelmäßig aktualisiert werden.

Die Entwicklung eines Foundation Models ist ressourcenaufwändig (Daten, Speicherplatz, Rechenleistung) und entsprechend kostspielig. In Tabelle 15 finden sich Kostenbeispiele für

die Entwicklung Europäischer Open-Source-Modelle und in Tabelle 16 ein Kostenvergleich von Foundation Model Training bis hin zu LoRA-basiertem Fine-Tuning.

Tabelle 15 Kostenvergleich Europäische Beispiele

Projekt	Modellgröße	Budget	Offenheit
Teuken-7B	7B	€14M Projekt	Voll offen (Apache 2.0)
OpenEuroLLM	1B/9B/22B	€37-56M	Voll offen geplant
GPT-NL	26B	€13,5M, davon €4M Pretraining	Offen (Niederländische Behörden)
Apertus	8B/70B	CHF 27M, davon 7M Pretraining	Voll offen (Daten, Gewichte, Trainingdetails) Apache 2.0
EuroLLM	22B	nicht beziffert	Voll (Apache 2.0)

Tabelle 16 Kostenvergleich Fine-Tuning vs Foundation-Modell-Training (von Grund auf)

Ansatz	Kosten (13B)	Zeit bis Deployment	Souveränität
From Scratch	€14-30M und mehr	18-24 Monate und mehr	Voll
Continued Pre-training	€2-5M	6-12 Monate	Hoch
Fine-tuning (LoRA)	€100-400K	2-4 Monate	Moderat
QLoRA (7B)	€50-500/Run	Tage	Moderat

Souveränität

Ist ein Foundation Model für sensitive Anwendungsfälle geeignet?

- Ja, da strikter On-Premise-Betrieb möglich ist, wobei die Anforderungen dafür sehr hoch sind.
- Unabhängigkeit von außereuropäischen oder europäischen Anbietern.
- Ausreichend IT-Infrastruktur und Personal für das Training, die Wartung und den Betrieb müssen vorhanden sein.
- Ethisch betrachtet wichtig: Datensouveränität heißt transparente.

- Dokumentation, Gestaltung und Kuration, um Verzerrungen oder Exklusionen zu vermeiden.

Betriebsmodelle

Für welche Betriebsmodelle ist ein Foundation Model geeignet (isolierter vs. zentral geregelter Betrieb)?

- Mit einem Foundation Model können verschiedene Varianten umgesetzt werden: ein Service für alle, zentral gemanagter Plattform Betrieb mit lokalen Erweiterungen, isolierte Anwendungen die zentral gemanagt werden, bzw. jedes Ressort betreibt das Modell selbst.
- Auch wenn ein Foundation Model mit einer bestimmten Anwendung isoliert betrieben wird, muss dennoch das Foundation Model selbst zentral gewartet werden.

Kompetenzen und Rollen

Foundation Model Training

Das Training eines Foundation Models (insbesondere beim Continued Pretraining oder Full Pretraining) erfordert ein interdisziplinäres Team, das tiefgreifende technische Aufgaben theoretisch lösen und praktisch umsetzen kann. In Tabelle 17 werden die verschiedenen Personalanforderungen für das Foundation Modell Training zusammengefasst. Um ein Foundation-Model-Training-Projekt erfolgreich On-Premise umzusetzen, muss das Team eine Reihe von weiteren Kompetenzfeldern abdecken. Dazu gehören folgende Kompetenzfelder und Rollen:

Kompetenzfelder

- Distributed Training: Da Foundation Models meist auf mehreren GPUs gleichzeitig trainiert werden, sind Kenntnisse in DeepSpeed, FSDP (Fully Sharded Data Parallel) oder Megatron-LM erforderlich.
- Data Curation & Filtering: Das Modell ist nur so gut wie die Daten. Kompetenz in der automatisierten Filterung von "Toxic Content", Dubletten und minderwertigem Text (Heuristiken, Klassifikatoren) ist entscheidend.

- Hardwarenahe Optimierung: Verständnis von CUDA, Kernel-Optimierung und Quantisierungstechniken, um das Maximum aus der Hardware herauszuholen.
- Evaluation Frameworks: Aufbau eigener Test-Sets (Benchmarks), um zu messen, ob das Modell durch das zusätzliche Training tatsächlich besser geworden ist (und nicht unter Catastrophic Forgetting leidet).

Zusätzliche organisatorische Rollen

- Legal & Compliance Officer: Klärung von Urheberrechtsfragen der Trainingsdaten (besonders wichtig bei On-Premise/proprietären Daten).
- Project Lead / Product Owner: Koordination zwischen technischer Machbarkeit und dem geschäftlichen Nutzen (return of investment).

Tabelle 17 Foundation Model Training Personalanforderungen: Rollen, Aufgaben, erforderliche Kompetenzen

Rolle	Hauptaufgabe	Wichtigste Kompetenz
Data Engineer	Aufbau der Daten-Pipeline (Scraping, Cleaning, Deduplizierung).	Spark, Python, SQL, Umgang mit Terabytes an Textdaten.
ML / NLP Engineer	Auswahl der Modellarchitektur, Tokenisierung und Hyperparameter-Tuning.	PyTorch, Deep Learning Frameworks, Hugging Face Ökosystem.
MLOps Engineer	Infrastruktur-Setup (GPU-Cluster), Orchestrierung und Monitoring.	Kubernetes, Docker, Slurm, NVIDIA Triton/vLLM.
Research Scientist	Wissenschaftliche Begleitung, Validierung der Modellgüte, Benchmarking. Ethische und rechtliche Expertise einbeziehen.	Evaluation von Sprachmodellen, Verständnis von Loss-Kurven.
Domain Expert	Fachliche Prüfung der Daten und Ergebnisse (z. B. Mediziner oder Jurist).	Tiefes Verständnis der Zielsprache/Fachterminologie.

Foundation Model Inference

Für den reinen Betrieb (Inference/Bereitstellung) eines Foundation Models verschiebt sich der Fokus weg von der Forschung (R&D) hin zur IT-Infrastruktur und Software-Architektur. Es werden weniger Data Scientists und dafür mehr Engineers benötigt. Siehe Tabelle 18 für eine Übersicht benötigter Rollen und Kompetenzen für den lokalen Betrieb eines LLMs. Des

Weiteren werden folgende spezifische technische Kompetenzen und unterstützende Rollen benötigt.

Spezifische technische Kompetenzfelder

- Inference Optimization: Wissen, wie man Modelle mittels Quantisierung (AWQ, GPTQ, GGUF) schrumpft, um Kosten zu sparen, aber die Modellqualität für die Nutzer:innen zu erhalten.
- Observability & Monitoring: Einrichtung von Dashboards (z. B. Prometheus/Grafana), um Latenz (Time to First Token), Durchsatz und GPU-Auslastung in Echtzeit zu sehen.
- Security & Identity: Implementierung von Zugriffskontrollen (Identity & Access Management), damit nicht jede Nutzer:in unbegrenzt GPU-Ressourcen verbrauchen kann.
- Prompt Engineering / RAG-Expertise: Kompetenz darin, das Modell über externe Datenquellen (Vector Databases) aktuell zu halten, ohne es ständig neu trainieren zu müssen.

Unterstützende Rollen

- Compliance/IT-Security Officer: Sicherstellung, dass die Datenverarbeitung auf dem On-Premise-Server den internen Richtlinien entspricht (besonders bei personenbezogenen Daten).
- Product Owner (AI): Definiert die Service Level Agreements (SLAs) – wie schnell und wie verfügbar muss die KI für die Fachabteilungen sein?

Zusammenfassend

Während für das Training Mathematiker:innen und KI-Forscher:innen erforderlich sind, werden für den Betrieb klassische IT-Spezialist:innen mit KI-Zusatzwissen gebraucht. Besonders das Wissen, wie man Hochverfügbarkeit auf GPUs sicherstellt, ist erforderlich.

Tabelle 18 Rollen und Kompetenzen für einen stabilen On-Premise-Betrieb eines LLMs

Rolle	Hauptaufgabe	Fokus-Kompetenz
MLOps Engineer	Automatisierung des Deployments, Versionierung der Modell-Weights.	Kubernetes, Docker, NVIDIA Container Toolkit, CI/CD-Pipelines.
System/Infrastruktur Admin	Überwachung der Hardware (GPUs), Kühlung und Stromversorgung.	Linux-Serveradministration, GPU-Monitoring (DCGM), Hardware-Troubleshooting.
Backend Developer	Integration des Modells in bestehende Anwendungen via API (REST/gRPC).	Python, FastAPI, Anbindung an Datenbanken oder RAG-Systeme.
AI Architect	Auswahl der optimalen Inferenz-Strategie (Quantisierung vs. Speed).	Verständnis von vLLM, TGI, TensorRT-LLM und Modell-Architekturen.

Soft- und Hardwareausstattung

Im Folgenden werden Beispiele für Soft- und Hardwareausstattungen gegeben: In Tabelle 19 werden benötigte Rechenleistung (GPU) und Trainingszeiten für den finalen Trainingsrun ausgewählter Europäischer Foundation Modelle aufgelistet.

Hardware für Modelltraining: finaler Trainingsrun

Tabelle 19 Foundation Model finaler Trainingsrun: benötigte Hardware und Trainingszeit

Modell	GPUs	Trainingszeit
GPT-NL 26B	88 NVIDIA H100 Nodes	4 Monate für finalen Training Run
Apertus	4000 NVIDIA GH200 Grace-Hopper-GPUs	~3 Monate
Teuken 7B	946 Nodes mit jeweils 4 NVIDIA A100 GPUs (40 GB Speicher) standen zur Verfügung; für das Training wurden 32 bis 1024 A100 GPUs verwendet	nicht beziffert
EuroLLM 22B	400 NVIDIA H100 GPUs	nicht beziffert

Hardware & Infrastrukturanforderungen für den LLM-Betrieb

Für den Betrieb von Foundation Models on-premise variieren die Hardware-Anforderungen stark je nach Modellgröße (Anzahl der Parameter) und der verwendeten Präzision (Quantisierung). Tabelle 20 gibt eine Übersicht über die grundlegenden Komponenten für den lokalen Betrieb eines LLM. Tabelle 21 illustriert die Hardware-Voraussetzungen für die drei gängigsten Modellgrößenordnungen im On-Premise-Betrieb.

Tabelle 20 Foundation Model Betrieb: grundlegende Hardware- & Infrastrukturanforderungen

Komponente	Fokus	Detail / Empfehlung
GPU-Speicher (VRAM)	Kapazität	Entscheidend für die Modellgröße (z. B. 70B Modell ca. 40GB VRAM bei 4-Bit Quantisierung).
Rechenleistung	Chip-Typ	NVIDIA A100/H100 (Enterprise) oder RTX 4090 (Consumer/Prosumer) für kleinere Setups.
Interconnect	Datendurchsatz	NVLink für schnelle Kommunikation zwischen mehreren GPUs zur Vermeidung von Flaschenhälsen.
Kühlung/Strom	Betriebssicherheit	Auslegung auf 300W–700W pro GPU; Server-Racks benötigen aktive Klimatisierung.

Tabelle 21 Foundation Model Betrieb: Hardware-Anforderungen nach Modellgröße

Modellgröße	RAM/VRAM Bedarf (ca.)	Empfohlene GPU-Klasse	Typische Anwendungsbereiche
7B - 8B (z.B. Llama 3)	8 – 16 GB	Entry-Level: RTX 3060/4060, RTX 4070 (12-16GB)	Schnelle Antwortzeiten, einfache Chat-Aufgaben, RAG-Systeme.
13B - 15B (z.B. Mistral)	16 – 32 GB	Mid-Range: RTX 3090/4090 (24GB); für quantisierte Modelle auch RTX 4070 Ti Super 16GB	Höhere Kohärenz, komplexere Instruktionen, gute Balance zwischen Speed & Qualität.
70B+ (z.B. Llama 3 70B)	48 GB – 128+ GB	Enterprise: Multi-GPU (2x 4090) oder NVIDIA A100/H100 (80GB)	Komplexe logische Schlussfolgerungen, tiefes Fachwissen, anspruchsvolle Analysen.

Die **Quantisierung** reduziert die Bit-Präzision der Modellgewichte, um den Speicherbedarf massiv zu senken. Für den produktiven On-Premise-Einsatz gelten folgende Richtwerte:

- FP16 (Unkomprimiert): Benötigt ca. 2 GB VRAM pro 1 Mrd. Parameter. Ein 70B-Modell benötigt hier etwa 140-150 GB VRAM.
- INT8 (8-Bit): Halbiert den Bedarf auf ca. 1 GB pro 1 Mrd. Parameter (z.B. 8 GB für ein 8B-Modell).
- 4-Bit (Q4/INT4): Ist der "Sweet Spot" für den On-Premise-Betrieb. Ein 70B-Modell passt hier mit ca. 40 GB VRAM auf eine einzelne High-End-GPU oder zwei Consumer-GPUs.

System-RAM: Unabhängig von der GPU sollten mindestens 32 GB bis 64 GB Systemspeicher vorhanden sein, um Modelle effizient laden und verwalten zu können.

CPU: Ein moderner Prozessor mit mindestens 8-16 Kernen (z.B. AMD Threadripper oder Intel Xeon) wird empfohlen, um die Datenvorbereitung für die GPU nicht zu bremsen.

Speicher: Es sollten NVMe SSDs für die Modellgewichte verwendet werden, um die Ladezeiten beim Starten oder Austauschen von Modellen gering zu halten.

Wenn sowohl viele gleichzeitige Nutzer:innen (Durchsatz) als auch maximale Geschwindigkeit (Latenz) für die/den Einzelne/n abgedeckt werden soll, verschiebt sich der Fokus von der reinen VRAM-Kapazität hin zur Speicherbandbreite und Software-Optimierung. In Tabelle 22 werden anhand dreier Modellgrößen Beispiele für die Szenarien hoher Durchsatz (High Throughput) und geringe Latenz (Low Latency) gegeben.

Tabelle 22 Foundation Model Betrieb: Hardware- & Setup-Strategie nach Modellgröße

Modellgröße	Fokus: Viele Nutzer:innen (Throughput)	Fokus: Geschwindigkeit (Latency)	Empfohlene Hardware-Konfiguration
7B - 8B	Continuous Batching: Kann hunderte Anfragen parallel auf einer Karte verarbeiten.	FP16 / Unquantized: 16 Bit Repräsentation von Zahlen.	1x RTX 4090 (24GB) oder 1x L40S (48GB) für hohe Nutzer:innenzahlen.
13B - 15B	KV-Cache Management: Benötigt effizienten Speicher für viele offene Konversionen.	Speculative Decoding: Ein kleineres Modell (z.B. 1B) rät Token, das 13B bestätigt.	1x A100 (80GB) oder 2x RTX 4090 (verbunden via NVLink/PCIe Gen4+).

Modellgröße	Fokus: Viele Nutzer:innen (Throughput)	Fokus: Geschwindigkeit (Latency)	Empfohlene Hardware-Konfiguration
70B+	Tensor Parallelism: Modell wird auf mehrere GPUs aufgeteilt, um Last zu verteilen.	Flash Attention 2: Minimiert Berechnungszeit pro Token massiv.	Min. 2x H100 (80GB) oder ein Verbund aus 4x A100 (für 4-Bit Quantisierung).

Wesentliche Stellschrauben beim On-Premise-Betrieb mit vielen Nutzer:innen sind die Engine für die Bereitstellung des/der LLMs und die Quantisierung, sowie entsprechende Multi-GPU-Strategien: vLLM ist eine Standardengine für Model Inference. Es nutzt PagedAttention, was den Speicher für Anfragen so effizient verwaltet wie das RAM in einem Betriebssystem (Kwon et al., 2023). Das verhindert, dass das System bei vielen Nutzer:innen "abstürzt". Activation-aware Weight Quantization (AWQ) oder das neuere FP8 (auf H100 Karten) sind schneller als 4-Bit (GGUF), behalten aber die Genauigkeit fast vollständig bei und sparen genug VRAM, um Platz für den "Chat-Verlauf" (KV-Cache) vieler Nutzer:innen zu lassen. Bei den Multi-GPU-Strategien lassen sich Data Parallelismus und Tensor Parallelismus unterscheiden. Beim Data Parallelismus liegt das gleiche Modell auf 2 GPUs, wobei Nutzer:in A auf GPU 1 und Nutzer:in B auf GPU 2 geht. Das ist ideal bei vielen Nutzer:innen. Beim Tensor Parallelismus wird ein Modell quasi zerschnitten und auf 2 GPUs verteilt. Das ist ideal für maximale Geschwindigkeit bei größeren Modellen, z.B. 70B-Modellen. In Tabelle 23 werden die Anforderungen für den Mischbetrieb umrissen, wobei das LLM in einem Rechenzentrum betrieben wird.

Tabelle 23 Foundation Model im Mischbetrieb: das Modell wird in einem Rechenzentrum betrieben

Anforderung	Umsetzung im Rechenzentrum
Netzwerk	10GbE oder höher intern, damit die Daten schnell vom Client zum Server fließen.
VRAM-Puffer	Einplanen von 20-30% extra VRAM, der nicht vom Modell belegt ist, sondern als "Arbeitsspeicher" für die Nutzer:innenanfragen (KV-Cache) dient.
Software	Deployment via Docker mit NVIDIA Triton Inference Server oder vLLM API.

Softwareanforderungen für die Bereitstellung eines LLM

Tabelle 24 gibt Software- bzw. Toolbeispiele für den Betrieb eines LLM entlang der Bereitstellungsbereiche Inference Engine, Optimierung, Containerisierung und Orchestrierung. In Tabelle 25 werden Maßnahmen für die Betriebssicherheit beim Einsatz eines LLMs hinsichtlich der Aspekte Modellisolation, Zugriffskontrolle, Überwachung und Lifecycle-Management zusammengefasst.

Tabelle 24 Foundation Model Betrieb: Software für die Bereitstellung des LLM

Bereich	Tool / Konzept	Nutzen
Inference Engine	vLLM, TGI, Triton	Optimiert den Durchsatz (Tokens pro Sekunde) und die Speichernutzung.
Optimierung	Quantisierung	Reduziert den VRAM-Bedarf massiv (z.B. von FP16 auf INT4) bei geringem Qualitätsverlust.
Container	Docker / K8s	Standardisierung des Deployments; Nutzung des NVIDIA Container Toolkits.
Orchestrierung	Ray / SkyPilot	Hilft bei der Verteilung von Lasten auf mehrere Knoten oder GPUs.

Tabelle 25 Foundation Model Betrieb: Sicherheit

Aspekt	Maßnahme	Ziel
Isolation	Air-Gapping / Firewall	Verhindert ungewollten Datenabfluss (Telemetrie) ins Internet.
Access Control	API-Gateway (z.B. Kong)	Regelt, welche internen Nutzer:innen oder Apps auf das Modell zugreifen dürfen.
Monitoring	Prometheus / Grafana	Überwachung von Latenz, GPU-Temperatur und Fehlerraten.
Lifecycle	Model Registry	Strukturierte Versionierung und Updates der Modellgewichte (Weights).

On-Premise Software-Stack für den Betrieb eines LLMs

Ein typischer On-Premise Software-Stack für den Betrieb (Inference) ist modular aufgebaut. Er trennt die Hardware-Ansteuerung von der eigentlichen KI-Logik und der Benutzer:innen-oberfläche. Tabelle 26 gibt einen Überblick über KI-Inferenz von der Basisinfrastruktur bis zum User-Interface. Die 3 wichtigsten Werkzeug-Gruppen für einen On-Premise-Betrieb sind:

- Modell-Management (Model Registry): Modelle werden lokal gespeichert (z. B. auf einem MinIO-Speicher).
- Vektordatenbanken (für RAG): z.B. Qdrant, Milvus oder Weaviate; wichtig ist, dass die Daten gut kuratiert und die Datenbanken kontinuierlich gepflegt werden.
- Monitoring & Observability: z.B. Prometheus & Grafana um Monitoren von GPUs oder Latenzzeiten relativ zu Nutzer:innenzahlen; LangSmith / Arize Phoenix um die Qualität der KI-Antworten zu loggen und zu verbessern.

Tabelle 26 Stack für KI-Inferenz von der Basisinfrastruktur bis zum User-Interface

Ebene	Technologie (Beispiele)	Funktion
7. User Interface	Open WebUI, Streamlit, Custom React App	Die Oberfläche, über die Nutzer:innen mit der KI chatten.
6. Application Logic	LangChain, LlamaIndex, Haystack	Steuert den Workflow (z. B. Dokumente suchen & zusammenfassen).
5. API Gateway	Kong, Traefik, Nginx	Regelt Authentifizierung, Rate-Limiting und Lastverteilung.
4. Inference Engine	vLLM, NVIDIA Triton, TGI (Hugging Face)	Lädt das Modell (7B, 70B) und generiert die Tokens (Antworten).
3. Orchestrierung	Kubernetes (K8s), Docker Compose	Verwaltet Container und weist ihnen GPUs zu.
2. Acceleration Libs	CUDA, NCCL, FlashAttention 2	Mathematische Bibliotheken zur GPU-Beschleunigung.
1. OS & Treiber	Ubuntu Server, NVIDIA Driver, Docker Runtime	Die Basisverbindung zwischen Hardware und Software.

Rechtliche Rahmenbedingungen

Die im Zusammenhang mit Fine-Tuning und Model Adaption erläuternden Besonderheiten sind auch bei dieser Umsetzungsvariante sinngemäß anwendbar.

Ökologische Befunde

Die in diesem Kapitel dargelegten ökologischen Befunde basieren auf folgender technologischen Variantenlogik:

LLM-Einbindung (V1) differenziert nach geringer, mittlerer, hoher und sehr hoher Komplexität. Für die ökologische Bewertung ist deshalb nicht nur das Vorhandensein eines LLM relevant, sondern das konkrete Architektur- und Prozessniveau des Zielsystems.

Model Adaptation (V2), es werden klassisches Fine-Tuning, LoRA/QLoRA und Knowledge Distillation unterschieden. Diese Untergliederung ist ökologisch relevant, weil Trainingsaufwand, Speicherbedarf und spätere Inferenzlast je nach Adaptionspfad deutlich variieren.

Foundation Model Training (V3) umfasst mehr als den finalen Trainingslauf. Dazu gehören auch Daten und Design, Pretraining, Enhancement, Post-Training sowie Safety, Optimierung und Deployment. Die ökologische Last ergibt sich daher aus einer Kette aufeinander aufbauender Phasen.

Methodik der „Ökologische Bewertung“

Normativer Rahmen und methodische Einordnung

Methodisch folgt die Analyse den Grundprinzipien der ISO 14040 und ISO 14044: Ziel- und Untersuchungsrahmen, Sachbilanzmodell, Wirkungsabschätzung und Interpretation werden transparent getrennt dargestellt (ISO, 2006a,b). Zugleich entspricht die Studie bewusst dem Charakter einer „Ökologischen Bewertung“. Sie ist damit primär auf Hotspot-Erkennung, Variantensensitivität und robuste Entscheidungsunterstützung ausgerichtet. Dieser Zuschnitt ist auch deshalb sachgerecht, weil Delort et al. (2023) und Ligozat et al. (2022) ausdrücklich darauf hinweisen, dass KI-Systeme bislang häufig weder mehrkriteriell noch über den gesamten Lebenszyklus hinweg hinreichend bewertet werden.

Die vorliegende Bewertung ist im Kern attributional[7] angelegt, ergänzt jedoch um qualitative consequential[8] Hinweise dort, wo Rebound-Effekte, Nutzungsexpansion, Infrastrukturduplikation oder Behördenfragmentierung für die strategische Interpretation entscheidend sind. Gerade bei KI-Systemen können Effizienzgewinne durch steigende Nutzung, größere Modelle oder agentische Mehrschritt-Workflows teilweise oder vollständig kompensiert werden (Delort et al., 2023; Jiang et al., 2024; Desroches et al., 2025).

Funktionale Einheit, Referenzflüsse und Vergleichslogik

Als übergreifender Vergleichsrahmen wird - in Anlehnung an die Projektpräsentation - eine funktionale Einheit von 1.000 Prompts in einem administrativen Nutzungskontext verwendet. Die Projektpräsentation nennt hierfür als Arbeitsvorschlag 1.000 Prompts mit je 100 Tokens; diese Logik wird hier als Baseline übernommen.

Für die literaturbasierten Betriebswerte nach Desroches et al. (2025) werden zusätzlich die dort spezifizierten workflowabhängigen Annahmen zu Input-, Output- und Kontextumfang (Chat, RAG, Agenten; Low/Medium/High) beibehalten, weil nur so eine intern konsistente Skalierung der veröffentlichten Werte möglich ist. Neben der nutzungsbezogenen Vergleichseinheit wird für die Entwicklungsphase eine zweite Vergleichslogik verwendet: die einmalige Bereitstellung des jeweiligen technischen Pfads bis zu einem einsatzfähigen Zustand.

Systemgrenzen und Lebenszyklusphasen

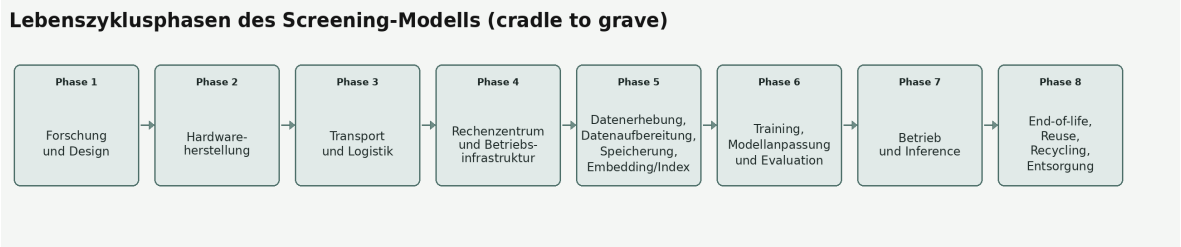
Die Systemgrenzen sind cradle to grave angelegt und orientieren sich an einer Acht-Phasen-Logik, wie sie in der LLM-Literatur und in den Projektunterlagen angelegt ist: Phase 1 Forschung und Design; Phase 2 Hardwareherstellung; Phase 3 Transport und Logistik; Phase 4 Rechenzentrumsbetrieb und Betriebsinfrastruktur; Phase 5 Datenerhebung, Datenaufbereitung, Speicherung sowie Embedding- und Index-Infrastruktur; Phase 6 Training,

[7] ‚Attributional' bezeichnet hier eine Betrachtungsweise, die die unmittelbar zurechenbaren Umweltwirkungen eines betrachteten Systems oder einer definierten Funktionseinheit erfasst, ohne weitergehende Markt-, Verlagerungs- oder Nutzungseffekte systematisch einzubeziehen.

[8] ‚Consequential' bezeichnet hier eine Betrachtungsweise, die nicht nur die direkt zurechenbaren Umweltwirkungen eines Systems erfasst, sondern auch solche zusätzlichen Effekte einbezieht, die sich aus seiner Einführung, Ausweitung oder veränderten Nutzung ergeben können, etwa durch Rebound-Effekte, steigenden Infrastrukturbedarf oder veränderte Organisationsstrukturen.

Modellanpassung und Evaluation; Phase 7 Betrieb und Inference; Phase 8 End-of-life, Wiederverwendung, Recycling und Entsorgung.

Abbildung 1 Lebenszyklusphasen der Screening-LCA



Verwendetes cradle-to-grave-Screeningmodell mit acht explizit benannten Lebenszyklusphasen

Inhaltliche Bedeutung im Screening-Modell:

1. Forschung, Design und Systemarchitektur; dazu zählen konzeptionelle Modellwahl, Architekturentscheidungen und frühe Entwicklungslogik.
2. Hardwareherstellung und Beschaffung; insbesondere Server, GPUs beziehungsweise andere Beschleuniger, Speicher- und Netzwerktechnik.
3. Transport und Logistik; hierzu gehören Lieferketten, Verbringung und vorgelagerte Distributionsprozesse der eingesetzten Infrastruktur.
4. Rechenzentrumsbetrieb und Betriebsinfrastruktur; insbesondere Stromversorgung, Kühlung, Flächenbetrieb und technische O&M-Strukturen.
5. Datenerhebung, Datenaufbereitung, Speicherung sowie Embedding- und Index-Infrastruktur; dazu zählen auch RAG-nahe Wissensbasis und Vektorstrukturen.
6. Training, Modellanpassung und Evaluation; hierzu gehören Pretraining, Fine-Tuning, LoRA/QLoRA, Distillation sowie Test- und Validierungsläufe.
7. Betrieb und Inference; also die laufende Nutzung des Systems im Verwaltungsalltag einschließlich Prompt-Verarbeitung und Antwortgenerierung.
8. End-of-life, Wiederverwendung, Refurbishment, Recycling und Entsorgung der relevanten Hardware- und Infrastrukturkomponenten.

Bewusst ausgeklammert oder nur qualitativ adressiert werden die Herstellung von Endnutzergeräten der Verwaltungsmitarbeitenden, allgemeiner Büroenergieverbrauch sowie Bauwerkswirkungen außerhalb der in der Literatur bereits pauschal mit

Rechenzentrumsindikatoren erfassten Infrastruktur. Diese Abgrenzung ist sachgerecht, weil die wesentlichen Entscheidungsunterschiede zwischen den Varianten auf Server- und Accelerator-Hardware, Datenhaltung, Training und Betriebslogik zurückgehen.

Indikatoren, Datenbasis und Unsicherheiten

Die quantitative bzw. semi-quantitative Bewertung umfasst vier Kernindikatoren: Endenergie (kWh), Klimawirkung (kg CO2e), Wasserverbrauch beziehungsweise wasserbezogene Deprivation (vereinfachend in L oder m3 berichtet) und Ressourcendepletion/Abiotic Resource Depletion (ADP; soweit Literaturwerte vorliegen, in mg Sb-eq dargestellt), cf. Ligozat et al. (2022), Li et al. (2023), Desroches et al. (2025). Ergänzend werden Rohstoffkritikalität, sozioökonomische Risiken, Transparenzdefizite und Governance-Fragen qualitativ bewertet.

Die Datenbasis ist hybrid: Die technischen Realisierungsvarianten definieren Scope und Entscheidungsobjekte; die verwendeten Studien von Jiang et al. (2024), Desroches et al. (2025), Delort et al. (2023) sowie Fernandez et al. (2025) liefern die Lebenszykluslogik, Training- und Inference-Größenordnungen, Wasser- und Embodied-Impact-Befunde sowie neuere Evidenz zur starken Sensitivität der Inferenzenergie gegenüber Input- und Output-Längen, Batchgrößen, Serving-Frameworks, GPU-Typen, Modellparallelismus und Dekodierungsstrategien; offizielle EU- und IEA-Quellen dienen der Einordnung von Rechenzentrumsindikatoren, Stromsystemkontext und kritischen Rohstoffen. Die Unsicherheit ist für V1 moderat, für V2 hoch und für V3 hoch bis sehr hoch, weil die Ergebnislage stark von Modellgröße, Datenmenge, Auslastung, Standort, PUE, WUE, Kühlkonzept und Scope der Emissionszuordnung abhängt.

Interpretationshinweis**:** Die numerischen Werte in dieser Unterlage sind nicht als exakte Prognose eines bereits festgelegten Bundes-LLM-Systems zu lesen. Sie sind Proxy- und Literaturwerte, die robuste Größenordnungen und Hebel sichtbar machen sollen. Gerade bei V3 sind veröffentlichte Werte unterschiedlicher Modelle nur bedingt direkt vergleichbar, weil Scope, Strommix, Datacenter-Effizienz, Idle-Anteile, Experimentieraufwand und Embodied-Accounting stark variieren.

„Ökologische Bewertung“ Variante V1 – LLM-Einbindung

Technische Charakterisierung

V1 integriert bestehende LLMs in operative Verwaltungssysteme. Die ökologische Signatur dieser Variante wird wesentlich weniger durch die Trainingsphase als durch Betriebslogik, Modellgröße, Prompt- und Kontextlänge, Retrieval-Tiefe, Agentenschritte, Serving-Frameworks, Batchgrößen, Dekodierungsstrategien und die Auslastung der zugrundeliegenden Infrastruktur bestimmt. Diese Variante ist deshalb ökologisch attraktiv, weil sie große Anfangslasten vermeidet; sie ist aber keineswegs 'leichtgewichtig', wenn sie als großvolumiger Service mit langen Kontexten oder agentischen Kontrollschleifen betrieben wird.

V1 reicht von Assistenz- und Chatfunktionen über wissensgestützte RAG-Systeme bis zu agentischen und strategisch-souveränen Systemen. Für die ökologische Bewertung ist daher nicht nur das 'Ob' der LLM-Einbindung, sondern vor allem das erreichte Komplexitätsniveau des Gesamtsystems ausschlaggebend.

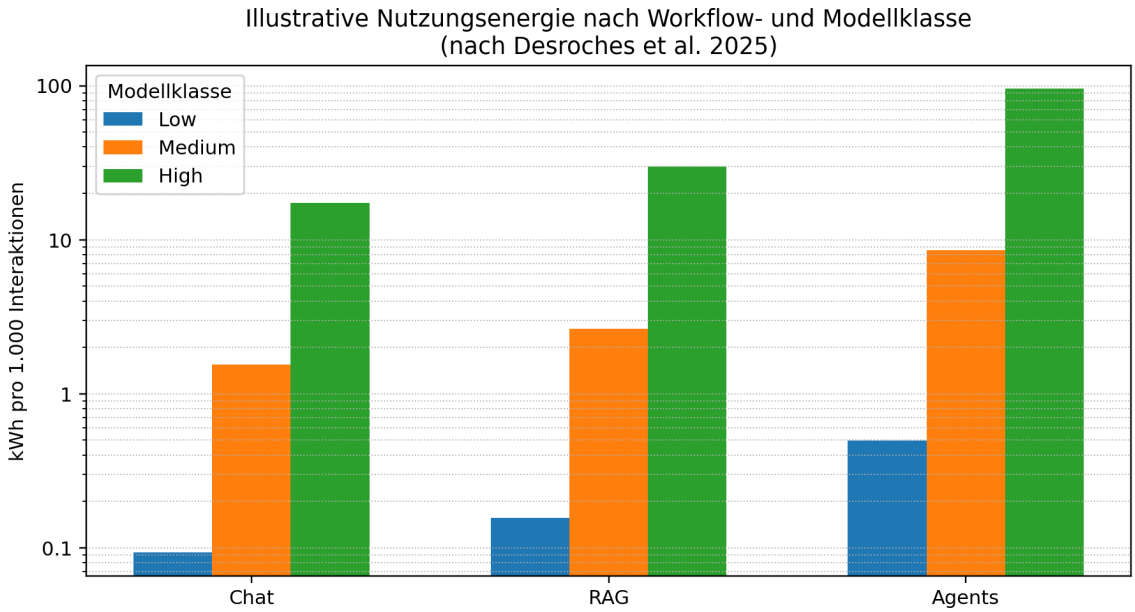
Tabelle 27 illustriert zwei robuste Muster. Erstens steigt der ökologische Aufwand mit der Modellklasse stark an. Zweitens ist Workflow-Komplexität ein eigenständiger Treiber: agentische Systeme liegen bei gleicher Modellklasse deutlich über Chat- und RAG-Konfigurationen (siehe auch Abbildung 2). Für die öffentliche Verwaltung folgt daraus, dass V1 nur dann ökologisch günstig bleibt, wenn der Systementwurf sparsame Standardpfade privilegiert und schwere agentische Ausnahmen eng begrenzt (Desroches et al., 2025).

Tabelle 27 Operative Screeningwerte für 1.000 Interaktionen

Workflow	Modellklasse	kWh / 1.000	kg CO2e / 1.000	L Wasser / 1.000	mg Sb-eq / 1.000
Chat	Low	0,093	0,059	3,8	0,018
	Medium	1,550	0,978	63,1	0,298
	High	17,300	10,879	704	3,330
RAG	Low	0,156	0,098	6,3	0,031
	Medium	2,640	1,668	107,4	0,508
	High	29,900	18,900	1.217	5,750
Agents	Low	0,497	0,313	20,2	0,100

Workflow	Modellklasse	kWh / 1.000	kg CO2e / 1.000	L Wasser / 1.000	mg Sb-eq / 1.000
	Medium	8,540	5,375	346,5	1,642
	High	95,800	60,400	3.890	18,450

Abbildung 2 Literaturbasierte Nutzungsenergie nach Workflow- und Modellklasse, auf 1.000 Interaktionen skaliert; logarithmische y-Achse (Werte nach Desroches et al., 2025)



Hotspots, Risiken und Interpretation

Dominant sind in V1 typischerweise die Lebenszyklusphase Betrieb und Inference sowie - bei RAG- und agentischen Systemen - die Lebenszyklusphase Datenspeicherung, Indexierung und Embedding-Infrastruktur. Bei hohem Nutzungsvolumen werden zusätzlich die zurechenbaren Beiträge aus Hardwareherstellung und Rechenzentrumsbetrieb relevanter.

Haupttreiber

Modellgröße, Promptlänge, Outputlänge, Retrievaltiefe, Agentenketten, Batchgrößen, Serving-Framework, Dekodierungsstrategie, Modellparallelismus, mangelnde Cache-Nutzung und niedrige Hardwareauslastung.

Methodischer Zusatzbefund

Fernandez et al. (2025) zeigen, dass naive FLOPs-basierte oder rein theoretische GPU-Auslastungsabschätzungen die reale Inferenzenergie systematisch unterschätzen können. Geeignete Inferenzoptimierungen – unter anderem optimierte Serving-Frameworks, Kernel- und Graph-Optimierungen, Continuous Batching sowie ein kontextabhängiger Einsatz von Dekodierungsverfahren – reduzierten in ihren realitätsnäheren Workloads den Energiebedarf gegenüber unoptimierten Baselines um bis zu 73 %.

Rebound-Risiko

Gerade bei scheinbar günstigen Chat- oder RAG-Setups kann eine massive Verbreitung im Verwaltungsalltag die kumulierten Betriebswirkungen stark erhöhen. Jiang et al. (2024) zeigen, dass die Servicephase großer Chatbots perspektivisch selbst den einmaligen Trainingslauf übersteigen kann.

Transparenzrisiko

Bei proprietären APIs fehlen häufig Primärdaten zu PUE, WUE, Auslastung, Modellrouting und embodied impacts. Das ist nicht nur ein Governance-, sondern auch ein LCA-Risiko (Delort et al., 2023; Jiang et al., 2024; Desroches et al., 2025).

Ökologischer Gesamtbefund

V1 ist der ökologisch bevorzugte Einstiegspfad für Bundes-LLM, sofern die Funktionalität mit vorhandenen oder moderat großen offenen Modellen, begrenzten Kontextfenstern und klar geregelten RAG-Workflows erreichbar ist. Die Variante verliert ihren Vorteil rasch, wenn V1 faktisch zu einem agentischen Multi-Step-System mit sehr großen Modellen und langen Kontexten ausgebaut wird. Die Designmaxime lautet daher: so einfach wie möglich, so groß wie fachlich nötig.

„Ökologische Bewertung“ Variante V2 – Model Adaptation

Technische Charakterisierung

V2 umfasst die Anpassung bereits vorhandener Modelle an spezifische Verwaltungsdomänen. Ökologisch ist diese Variante ambivalent: Einerseits vermeidet sie das vollständige Vortraining eines neuen Basismodells; andererseits erzeugt sie zusätzliche Trainingsläufe, Experimente, Datenvorbereitung und eventuell stakeholder-spezifische Modellfragmentierung. Entscheidend ist daher nicht allein, ob adaptiert wird, sondern wie Trainingslast, Speichernachfrage, Modellfragmentierung und spätere Betriebslogik sind wesentlich vom Realsierungspfad beeinflusst (siehe Tabelle 28).

Tabelle 28 Realsierungspfade und deren Screening-Interpretation

Subpfad	Entwicklungs-aufwand	Folgelast im Betrieb	Screening-Interpretation
LoRA / QLoRA	niedrig	gering bis moderat	Bevorzugte Standardoption, wenn Adaptation wirklich nötig ist
Volles Fine-Tuning	mittel bis hoch	oft unverändert hoch, da Basismodell weiter groß bleibt	Nur mit klar nachgewiesenem Performancegewinn vertretbar
Distillation	niedrig bis mittel	potenziell stark sinkend, wenn kleineres Student-Modell produktiv betrieben wird	Ökologisch attraktiv bei stabilen Hochvolumenanwendungen

Literaturbasierte Befunde

- Wang et al. (2023) zeigen für BERT-basierte NLP-Workloads, dass der Energieaufwand des Pretrainings - abhängig vom Datensatz - etwa 400 bis 45.000 Fine-Tuning-Läufen entsprechen kann. Das belegt den deutlichen Abstand zwischen V2 und einem echten Foundation-Training.
- Für die hier relevante Variantenentscheidung folgt daraus: Der ökologische Unterschied innerhalb von V2 hängt weniger an der bloßen Tatsache 'Adaption ja/nein' als an der konkreten Methode. Parameter-effiziente Verfahren reduzieren den Trainingsaufwand erheblich; ihr Vorteil wird aber nur dann voll wirksam, wenn der spätere Zielbetrieb tatsächlich effizienter wird (Wang et al., 2023).

- LoRA trainiert nur einen sehr kleinen Anteil der Modellparameter. Zweitens kann Distillation kleinere, produktiv betreibbare Student-Modelle erzeugen und damit gerade bei häufig genutzten Verwaltungsanwendungen die spätere Inferenzlast senken.
- Desroches et al. (2025) zeigen zugleich, dass der Betriebsaufwand stark mit Modellgröße und Workflow-Komplexität skaliert. Distillation oder schlanke Domänenmodelle sind daher ökologisch nur dann ein Vorteil, wenn sie wirklich zu kleineren produktiven Modellen, kürzeren Kontexten oder weniger Retrieval- und Agentenschritten führen.

Hotspots, Risiken und Interpretation

- Dominant sind in V2 typischerweise die Trainings- und Anpassungsphase sowie die spätere Betriebsphase. Im Gegensatz zu V1 ist der Anpassungsaufwand nicht mehr vernachlässigbar; im Gegensatz zu V3 bleibt er jedoch meist um Größenordnungen kleiner.
- Haupttreiber: Zahl der Experimente, stakeholder-spezifische Adapter-Vervielfachung, Datenaufbereitung, Modellgröße im Zielbetrieb, Sequenzlängen und Auslastung.
- Organisationsrisiko: Wenn jede Behörde eigene Adapter, eigene Testsets und eigene Modellpfade pflegt, entstehen vermeidbare Doppelarbeiten und zusätzliche ökologische Lasten. Aus Nachhaltigkeitssicht ist ein geteiltes Adapter- und Evaluationsregime klar vorzuziehen.
- Systemischer Hebel: Distillation und kleinere Domänenmodelle sind nur dann ökologisch vorteilhaft, wenn das resultierende kleinere Modell tatsächlich produktiv genutzt wird und nicht zusätzlich zum großen Basismodell dauerhaft parallel weiterläuft.

Ökologischer Gesamtbefund

V2 ist aus ökologischer Sicht eine selektiv sinnvolle Mitteloption. Die bevorzugte Reihenfolge innerhalb der Variante lautet: PEFT vor Full Fine-Tuning, Knowledge Distillation vor dauerhaftem Großmodellbetrieb. V2 ist dann besonders attraktiv, wenn ein dokumentierter Qualitätsgewinn erreicht wird, der entweder große RAG-Kontexte überflüssig macht oder den Einsatz kleinerer, effizienterer Betriebsmodelle ermöglicht.

„Ökologische Bewertung“ Variante V3 – Foundation Model Training

Technische Charakterisierung

V3 ist der infrastrukturell und ökologisch anspruchsvollste Pfad. Er umfasst - je nach Ausprägung - Datensammlung im großen Maßstab, Tokenizer- und Architekturdesign, Vortraining, mögliche Mid- und Post-Training-Schritte, Sicherheits- und Evaluationsphasen sowie den späteren Betrieb. Schon die Beschaffung und Vorhaltung der dafür benötigten Beschleunigerhardware verschiebt wesentliche Lasten in die Lebenszyklusphasen Hardwareherstellung, Rechenzentrumsbetrieb und Training beziehungsweise Anpassung.

V3 umfasst mindestens fünf Hauptphasen: Daten & Design, Pretraining, Enhancement, Post-Training sowie Safety, Optimierung und Deployment. Die ökologische Bewertung darf V3 deshalb nicht auf den finalen Trainingslauf verkürzen.

Literaturbasierte Ordnungsgrößen

Die Größenordnungen in Tabelle 29 sind bewusst nebeneinandergestellt, obwohl sie nicht 1:1 vergleichbar sind. Gerade dieser Vergleich ist lehrreich: Er zeigt, wie stark Ergebnisse von Scope, Datacenter-Effizienz, Strommix, Idle-Anteilen und Embodied-Accounting abhängen. Was jedoch invariant bleibt, ist die strukturelle Aussage: Foundation-Training verschiebt die ökologische Last in einen Bereich, der für V1 und V2 nicht annähernd erreicht wird (Patterson et al., 2021; Jiang et al., 2024; Luccioni et al., 2024; Faiz et al., 2024).

Tabelle 29 Beispiele von stromverbrauchsbedingter Klimawirkung beim LLM-Training

Referenz	Scope	Strom	Klimawirkung	Interpretationshinweis
BLOOM 176B	Gesamter modellbezogener CF inkl. embodied, idle, dynamic	—	50,5 t CO2e	niedriger Strommix; anderer Scope als reine Trainingsstudien
BLOOM 176B	Final training – dynamic only	—	24,7 t CO2e	nur dynamischer Trainingsstrom, ohne embodied/idle
GPT-3 175B	Final training run	1.287 MWh	552 t CO2e	publizierter Referenzwert für großes Dense-Modell

Referenz	Scope	Strom	Klimawirkung	Interpretationshinweis
GPT-4 (Schätzung)	Final training run	7.200 MWh	~3.088 t CO2e	Jiang et al. (2024) als Modellschätzung, kein Herstellerwert

Infrastruktur- und Rohstoffdimension

- Der Bedarf an sehr großen Hardwarekonfigurationen - etwa hunderte H100 oder sehr große GH200- beziehungsweise A100-Äquivalente über längere Trainingszeiträume – verdeutlicht die erhebliche Infrastrukturintensität von V3.
- Mit steigender Accelerator-Nachfrage nehmen ökologischen Auswirkungen (Embodied Impacts), Lieferkettenabhängigkeiten und Rohstoffkritikalitäten zu. Für Halbleiter-, Speicher-, Stromversorgungs- und Kühltechnik sind kritische Rohstoffketten relevant, die auf europäischer Ebene inzwischen ausdrücklich im Rahmen des Critical Raw Materials Act adressiert werden (EU,2024b, 2026a, 2026b).
- Neben der Hardwareproduktion ist Wasser ein wesentlicher Hotspot. Wasser wird sowohl in der Halbleiterfertigung als auch im Rechenzentrumsbetrieb benötigt; Li et al. zeigen, dass die wasserbezogene Wirkung je nach Standort, Kühlkonzept und Betriebsfenster stark variieren kann (Li et al., 2023).
- V3 ist auch unter Resilienzgesichtspunkten ambivalent: Eigene Trainingskapazitäten erhöhen potenziell die technologische Souveränität, können aber Stromnetz- und Kühlungsanforderungen lokal konzentrieren und erhöhen den Bedarf an spezialisierten Ersatzteilen sowie qualifiziertem Betriebspersonal.

Ökologischer Gesamtbefund

V3 ist aus ökologischer Sicht nur bedingt vertretbar und muss als strategische Reserveoption behandelt werden. Die Variante ist nur dann ernsthaft zu verfolgen, wenn ein substanzieller Souveränitäts- oder Sicherheitsgewinn nachgewiesen ist, eine europäische Ressourcenbündelung geprüft wurde und von Beginn an primärdatenfähiges Nachhaltigkeitsmonitoring vorgesehen ist. Ökologisch besonders problematisch wäre ein souveränes From-Scratch-Training, das bereits existierende europäische oder offene Modellfamilien dupliziert, ohne dass ein klarer Mehrwert in Qualität, Rechtssicherheit oder Sicherheitsarchitektur entsteht.

Querschnittsvergleich und Sensitivität gegenüber Implementierungsszenarien

Tabelle 30 fasst den Kern der Entscheidung zusammen: Der ökologische Grenznutzen sinkt deutlich, je weiter man von V1 über V2 zu V3 geht. Zugleich dürfen die qualitativen Souveränitätseinschätzungen nicht isoliert gelesen werden, sondern nur im Zusammenspiel mit den Infrastruktur-Szenarien Sovereignty-First (S1), Hybrid Flexibility (S2) und Ecosystem Partnership (S3), siehe Tabelle 32. Reale Souveränität hängt nicht allein vom technischen Pfad, sondern auch von Hosting, offenem Modellzugang, Lieferketten, Personalverfügbarkeit und Governance-Strukturen ab (EU, 2024a,b).

Tabelle 30 Variantenvergleich

Variante	Initiale Entwicklungs-belastung	Belastung im Zielbetrieb	Rohstoff-/Lieferketten-kritikalität	Souveränitäts-potenzial (abhängig von Betriebsform)	Ökologische Eignung als Startoption
V1 LLM-Einbindung	niedrig	niedrig bis mittel; bei agentischen Pfaden hoch	mittel	variabel; hosting- und modellzugangsabhängig	hoch
V2 Model Adaptation	niedrig bis mittel	niedrig bis mittel; mit Distillation potenziell niedrig	mittel	mittel bis hoch	mittel bis hoch
V3 Foundation Training	sehr hoch	mittel bis hoch	sehr hoch	hoch, aber aufwändig	niedrig als Startpfad

Einordnung der Infrastruktur-Szenarien S1-S3

Aus ökologischer Sicht ist das Partnerschaftsmodell S3 besonders relevant, weil es die zentrale Nachhaltigkeitsfrage von V3 adressiert: Muss ein großes Basismodell wirklich national doppelt aufgebaut werden, oder kann strategische Souveränität mit europäischer Arbeitsteilung, Transparenzauflagen und lokalen Betriebsdomänen kombiniert werden? Für V1 und V2 ist S2 häufig der pragmatischste Übergangspfad, solange Nachhaltigkeitsdaten vertraglich eingefordert werden. In der Kreuzung mit den technischen Varianten ergibt sich

screeningartig folgende Tendenz: V1 ist in allen drei Szenarien grundsätzlich darstellbar, bleibt in S1 aber nur bei hoher gemeinsamer Auslastung ökologisch plausibel; V2 ist vor allem in S2 und S3 gut begründbar; V3 ist aus ökologischer Sicht primär dann vertretbar, wenn S3 redundante Trainings- und Hardwarelasten tatsächlich vermeidet.

Tabelle 31 Ökologische Einordnung der Infrastrukturszenarien S1-S3

Szenario	Ökologische Chancen	Ökologische Risiken	Ökologische Einordnung des Szenarios
S1 Sovereignty-First / On-Prem	hohe Kontrolle über Daten und Technik; potenziell hohe lokale Resilienz	Gefahr zusätzlicher Hardwarebeschaffung, geringerer Auslastung, hoher lokaler Strom- und Kühlbedarf	Ökologisch nur dann günstig, wenn bestehende Infrastruktur intensiv gemeinsam genutzt, hohe Auslastung erreicht und zusätzliche Hardwareduplikation weitgehend vermieden wird.
S2 Hybrid Flexibility	Nutzung elastischer, teils effizienter externer Infrastruktur möglich; schnellere Skalierung	mehr Netzwerk- und Governance-Komplexität; ökologische Qualität hängt stark von Provider-Transparenz ab	Oft der pragmatisch günstigste kurzfristige Kompromiss, sofern belastbare Nachhaltigkeitsdaten und Steuerungsmöglichkeiten gegenüber den Providern vorliegen.
S3 Ecosystem Partnership	Vermeidung redundanter Modelltrainings und doppelter Hardwarebestände; Know-how- und Ressourcenteilen	geringeres Autonomieniveau; Nutzen hängt von Partnerreife und Allokationsregeln ab	Ökologisch besonders attraktiv, wenn echte Ressourcenteilung stattfindet und Doppelstrukturen konsequent vermieden werden.

Kritische Rohstoffe, sozioökonomische Aspekte und systemische Risiken

Für digitale Infrastrukturen sind die in Tabelle 32 aufgelisteten Dimensionen nicht nachrangig, sondern zentral: Halbleiter, Leiterplatten, Netzteile, Kühlsysteme und Speicherhardware binden hochgradig globalisierte Lieferketten, deren Materialbasis oft durch Importkonzentration, niedrige Recyclingraten und sozial-ökologische Konflikte geprägt ist (Delort et al., 2023; EU, 2024b, 2026a,b).

Tabelle 32 Risikodimensionen

Risikodimension	Betroffene Lebens-zyklusphasen	Besonders relevante Varianten	Screening-Rating	Minderungsansatz
Lieferketten-konzentration bei CRMs	Hardwareherstellung; End-of-life, Wiederver-wendung und Recycling	V2, V3 > V1	hoch	Beschaffungs-auflagen, Reparierbarkeit, Reuse, europäische Partnerschaften, längere Hardwarelebens-dauer
Wasserstress in Halbleiterferti-gung und Rechenzentren	Hardware-herstellung; Rechenzentrumsbetrieb; Training und Anpassung; Betrieb und Inference	alle, besonders V3	hoch	Standortwahl, WUE-Transparenz, Kühlkonzept, wasserarme Betriebsfenster
E-Waste und geringe Rückgewin-nungsraten	End-of-life, Wiederver-wendung und Recycling	alle	mittel bis hoch	Take-back, Refurbishment, modulare Beschaffung, Recovery-Pfade für CRM-haltige Komponenten
Rebound durch Nutzungs-expansion	Betrieb und Inference	V1, V2	hoch	Use-case-Governance, Nutzungsmonitor-ing, Standardpfade statt unbeschränkter Agentik
Intransparenz externer Providerdaten	Rechenzentrums-betrieb; Datenhaltung; Training und Anpassung; Betrieb und Inference	v.a. V1, V2	hoch	vertragliche KPI-Pflichten: PUE, WUE, REF, ERF, Modellrouting, Auslastung
Verteilungsgerechtigkeit öffentlicher Ressourcen	Training und Anpassung; Betrieb und Inference	V2, V3	mittel	Priorisierung von High-Value-Use-Cases, Shared Services statt Behördeninseln

Für die operative Nachhaltigkeitssteuerung ist relevant, dass die Delegierte Verordnung (EU) 2024/1364 unionsweit aggregierte Rechenzentrumsindikatoren wie PUE, WUE, ERF und REF adressiert. Für die Entwicklung und den Einsatz von KI-basierten Systemen sollten solche Kennzahlen nicht nur als regulatorisches Minimum, sondern als verbindliche Beschaffungs- und Monitoringkriterien für produktionsrelevante Infrastrukturen verstanden werden (EU, 2024a).

Ökologische und rohstoffkritische Einordnung der drei technischen Varianten im Überblick

Die nachfolgende Synopse (Tabelle 33) verdichtet die ökologische und rohstoffkritische Einordnung der drei technischen Varianten. Sie ist als wissenschaftlich fundierte Screening-Gegenüberstellung zu lesen und dient der schnellen Vergleichbarkeit zentraler Belastungs- und Risikodimensionen.

Tabelle 33 Ökologische und rohstoffkritische Einordnung der 3 technischen Realisierungsvarianten

Kriterium	V1 - LLM-Einbindung	V2 - Model Adaptation	V3 - Foundation Model Training
Technische Grundlogik	Nutzung bestehender Modelle, typischerweise Chat, RAG, gegebenenfalls agentische Workflows.	Anpassung vorhandener Modelle, z. B. LoRA, QLoRA, Fine-Tuning oder Distillation.	Eigenes Basismodell beziehungsweise großskaliges Pretraining/Continued Pretraining.
Initiale ökologische Last	niedrig	niedrig bis mittel	sehr hoch
Laufende ökologische Last im Betrieb	niedrig bis mittel; bei großen Modellen, tiefem Retrieval und agentischen Workflows auch hoch.	mittel; bei erfolgreicher Distillation potenziell niedriger als V1.	mittel bis hoch; zusätzlich hohe Vorlast aus Entwicklung, Training und Hardware.
Klimawirkung (Screening)	günstigster Einstiegspfad, solange Modelle möglichst rechtssized bleiben und Workflows schlank sind.	ökologisch sinnvoll, wenn Adaptation echte Effizienz- oder Qualitätsgewinne bringt.	mit Abstand höchste Klimawirkung wegen Training, Experimenten, Hardware und Infrastruktur.

Kriterium	V1 - LLM-Einbindung	V2 - Model Adaptation	V3 - Foundation Model Training
Wasserrelevanz	primär betriebsbedingt durch Rechenzentrumsbetrieb und Inference.	betriebsbedingt plus zusätzliche Anpassungsläufe.	besonders hoch, da Training sowie vorgelagerte Hardware- und Halbleiterproduktion wasserintensiv sind.
Ressourcenbedarf / Rohstoffintensität	mittel	mittel bis hoch	sehr hoch
Kritikalität von Rohstoffen	relevant durch Nutzung von GPU- und Server-Infrastruktur; ohne neue Trainingsgroßinfrastruktur oft begrenzter.	steigt bei zusätzlicher Adapter-, Trainings- und Hardwarevorhaltung.	am höchsten, da große Mengen spezialisierter Hardware und damit CRM-Abhängigkeiten relevant werden.
Embodied Impacts der Hardware	vorhanden, aber oft indirekt über genutzte Infrastruktur zugerechnet.	spürbar, wenn zusätzliche Hardware oder dedizierte Instanzen benötigt werden.	zentraler Hotspot, weil Hardwarebeschaffung und Rechenzentrumsvorhaltung stark ins Gewicht fallen.
Dominante ökologische Hotspots	Inference, Datenhaltung, Embeddings, Retrievaltiefe und Agentenschritte.	Anpassungsläufe plus späterer Zielbetrieb.	Hardwareproduktion, Datenhaltung, Training, Kühlung und Betrieb.
Rebound-Risiko	hoch, weil breite Nutzung im Verwaltungsalltag die Gesamtlast stark erhöhen kann.	mittel bis hoch, besonders wenn viele behördenspezifische Adapter parallel wachsen.	mittel, aber auf sehr hohem absoluten Lastniveau.
Transparenz / Bilanzierbarkeit	oft eingeschränkt bei proprietären APIs.	mittel; besser bei offenen Modellen, schwieriger bei gemischten Setups.	besser steuerbar bei eigener Infrastruktur, aber datenintensiv und aufwendig.
Ökologische und Kritikalitäts-Gesamtbewertung	ökologisch am günstigsten als Startpfad; Kritikalität insgesamt mittel.	ökologisch sinnvolle Mitteloption; Kritikalität mittel bis hoch.	nur als Reserve- beziehungsweise Ausnahmeoption vertretbar; Kritikalität sehr hoch.

Strategische Empfehlungen für KI-basierte Systeme in der öffentlichen Verwaltung

Tabelle 34 fasst die strategischen Empfehlungen für KI-basierte Systeme in der öffentlichen Verwaltung zusammen.

Tabelle 34 Empfehlungen

Nr.	Empfehlung	Begründung / Umsetzungshinweis
1	V1 als Default-Pfad festlegen	RAG/LLM-Einbindung mit passend dimensionierten offenen Modellen als ökologische Baseline; agentische Pfade nur auf Basis klarer Nutzenbelege freigeben
2	PEFT-first-Prinzip für V2	LoRA/QLoRA vor Full Fine-Tuning; Full Fine-Tuning nur mit dokumentierter fachlicher Notwendigkeit
3	Distillation für stabile Hochvolumenfälle	Nur dort, wo kleinere produktive Modelle die spätere Inference-Last tatsächlich senken
4	V3 als Reserveoption definieren	Kein From-Scratch-Training ohne Souveränitätsnachweis, Primärdatenplan und Prüfung europäischer Pooling-Optionen
5	Nachhaltigkeits-KPIs vertraglich verankern	Mindestens PUE, WUE, REF, ERF, Standort, Hardwarealter, Auslastung, Take-back/Refurbishment, Modellrouting
6	Kontext-, Agenten- und Inferenzbudgets einführen	Grenzen für Promptlängen, Outputlängen, Retrievaltiefe und Tool-Schritte; Caching, Reuse, passende Batch-Strategien, energieeffiziente Serving-Frameworks und zurückhaltenden Einsatz energieintensiver Dekodierungsverfahren systematisch nutzen
7	Gemeinsame Evaluations- und Adapter-Governance	Vermeidung behördenspezifischer Doppelentwicklungen, gemeinsamer Benchmark- und Modellpflegeprozess
8	Hardwarelebensdauer maximieren	Beschaffung modularer Systeme, Refurbishment, Second-Life-Strategien, standardisierte Rücknahmeverträge
9	Nutzungsgerechtigkeit operationalisieren	Compute-intensive Pfade nur für Anwendungsfälle mit hohem öffentlichen Mehrwert reservieren; Low-compute-Alternativen aktiv prüfen
10	Primärdatenfähige nächste Projektphase vorbereiten	Standortspezifische Messung von Strom, Wasser, Auslastung, Modelltelemetrie und End-of-life-Strömen

Priorisierte Entscheidungslinie

Aus ökologischer Sicht ist die robuste, wissenschaftlich begründbare Entscheidungslinie für die Entwicklung und den Einsatz von KI-basierten Systemen in der öffentlichen Verwaltung wie folgt: V1 zuerst, V2 selektiv zweitens, V3 nur unter strengen Bedingungen. Diese Priorisierung bleibt auch dann stabil, wenn man die drei Infrastruktur-Szenarien S1-S3 mitdenkt. Änderungen an dieser Reihenfolge wären nur dann sachlich begründbar, wenn neue Primärdaten oder sicherheits-bzw. souveränitätsbezogene Anforderungen einen klaren, dokumentierten Zielkonflikt erzeugen.

Empfohlene Maßnahmen

Gemeinsame Infrastruktur statt Behördeninseln, verbindliche Nachhaltigkeits-KPIs für Provider, begrenzte Kontextfenster, systematische Wiederverwendung von Embeddings und Ergebnissen, energieeffiziente Serving-Frameworks und passende Batch-Strategien, gezielte Knowledge Distillation für Daueranwendungen, längere Hardwarelebensdauer und einheitliche Evaluationsregeln sind in allen Varianten sinnvoll. Diese Maßnahmen verbessern nicht nur die Umweltbilanz, sondern auch Kosteneffizienz, Transparenz, Governance-Fähigkeit und Resilienz des Vorhabens.

Grenzen der Aussagekraft und nächster Datenerhebungsbedarf

Die größte methodische Einschränkung ist der fehlende Zugriff auf primäre Systemdaten der zukünftigen Zielsysteme. Da es sich bei der vorliegenden Studie um eine wissenschaftlich fundierte Auslotung der potentiellen Möglichkeiten, Rahmenbedingungen und deren Auswirkungen handelt und zum Zeitpunkt der Studiendurchführung weder finale Architekturentscheidungen noch verifizierte Standort-, Auslastungs- oder Lieferantendaten für produktionsreife Realisierungen existierten, musste diese Studie notwendigerweise mit Literaturproxies, veröffentlichten Referenzwerten und projektinternen Strukturannahmen arbeiten (Ligoziat et al., 2022; Delort et al., 2023; Faiz et al., 2024; Desroches et al., 2025).

Neuere comparative LCAs, die LLMs menschlicher Arbeitsleistung gegenüberstellen, zeigen zwar für bestimmte Schreibaufgaben teils niedrigere direkte Umweltwirkungen pro Outputeinheit, verwenden jedoch eine andere funktionale Einheit und einen anderen Vergleichsgegenstand als die hier vorgenommene Variantenanalyse. Sie sind daher für die breitere Debatte über Substitution, Produktivität und Rebound relevant, ändern aber die hier getroffene Rangfolge zwischen V1, V2 und V3 nicht (Ren, 2024).

Für eine nachgelagerte, deutlich belastbarere Ökobilanz sollten mindestens die folgenden Primärdaten erhoben werden: (i) tatsächliche Hardwarekonfiguration inklusive Nutzungsdauer und Beschaffungsweg; (ii) standortspezifische Rechenzentrums-KPIs nach EU-Schema (PUE, WUE, REF, ERF); (iii) Telemetriedaten aus Training, Adaptation und Inference (Token, Batchgrößen, Laufzeit, Auslastung, Cache-Treffer, Retrieval-Tiefe); (iv) Datenvolumina, Speicherhaltung und Replikationslogik; (v) Rücknahme-, Refurbishment- und Recyclingpfade. Siehe Tabelle 35.

Tabelle 35 Erforderliche Primärdaten

Datenbereich	Erforderliche Primärdaten in der nächsten Projektphase
Hardware	BOM-nahe Beschaffungsdaten, GPU/CPU-Typen, Stückzahlen, Nutzungsdauer, Refurbishment-Optionen
Rechenzentrum	PUE, WUE, REF, ERF, Strommix (market- und location-based), Standortwasserstress, Kühlkonzept
Modelle / Workflows	Parameterzahl, Quantisierung, Prompt-/Outputlängen, RAG-Tiefe, Agentenschritte, Hit Rates
Trainingsläufe	Experimentprotokolle, Fehlläufe, Hyperparameter-Suchen, Adapter-Versionen, Distillation-Runs
End-of-life	Take-back, Ersatzteilverfügbarkeit, Wiederverwendung, Recyclingquote CRM-haltiger Komponenten

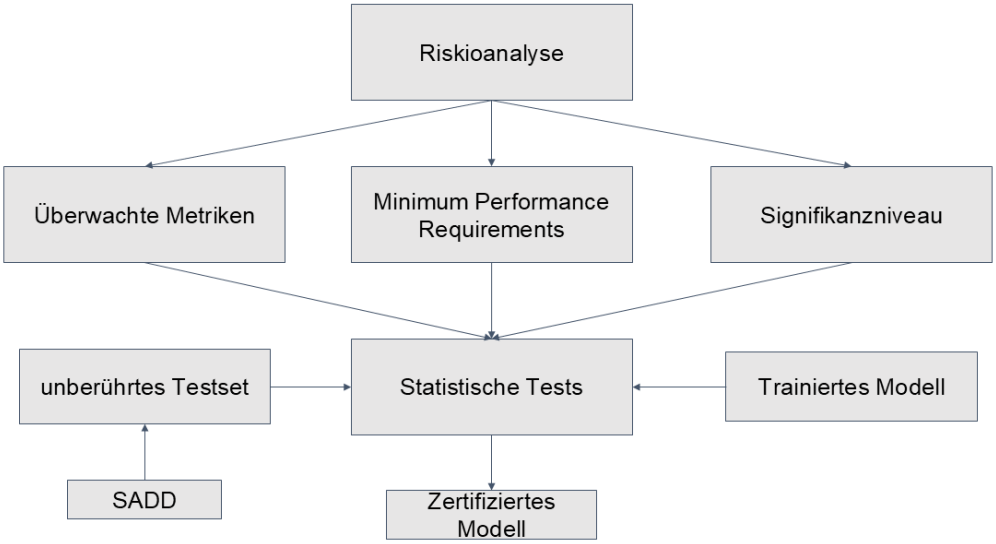
Kontinuierliche Qualitätssicherung

Kontinuierliche Qualitätssicherung bildet das Rückgrat vertrauenswürdiger KI-Systeme über ihren gesamten Lebenszyklus hinweg. Anders als bei klassischen Softwaresystemen ist Qualität hier kein statischer Zustand, der einmalig geprüft und anschließend vorausgesetzt werden kann. Vielmehr unterliegen KI-Modelle dynamischen Veränderungen – sei es durch neue Daten, Modellanpassungen, veränderte Einsatzbedingungen oder schleichende Verteilungsverschiebungen (Data Drift). Entsprechend muss Qualität fortlaufend überwacht, überprüft und bei Bedarf neu abgesichert werden.

Im Zentrum steht dabei das Konzept der **Functional Trustworthiness**, also der nachweisbaren funktionalen Qualität eines maschinellen Lernsystems (Nessler, Doms & Hochreiter, 2023). Vertrauen entsteht in diesem Kontext nicht durch reine Dokumentation oder deklarative Aussagen, sondern durch quantifizierte, statistisch belastbare Evidenz. Voraussetzung dafür ist ein klar definierter Einsatzbereich, messbare Anforderungen sowie valide Testverfahren, die belastbare Aussagen über das Systemverhalten erlauben.

Die grundlegende Struktur dieses Ansatzes lässt sich wie folgt beschreiben: Ausgehend von einer Risikoanalyse werden überwachte Metriken, Mindestanforderungen sowie ein Signifikanzniveau definiert. Diese Elemente fließen in statistische Tests ein, die auf einem unberührten Testset basieren und auf ein trainiertes Modell angewendet werden. Das Ergebnis ist ein zertifiziertes Modell, dessen Qualität nicht nur behauptet, sondern empirisch abgesichert ist. Die **Stochastic Application Domain Definition (SADD)** bildet dabei die Grundlage für die Auswahl geeigneter Testdaten und stellt sicher, dass die Tests den intendierten Einsatzbereich tatsächlich repräsentieren.

Abbildung 3 Functional Trustworthiness



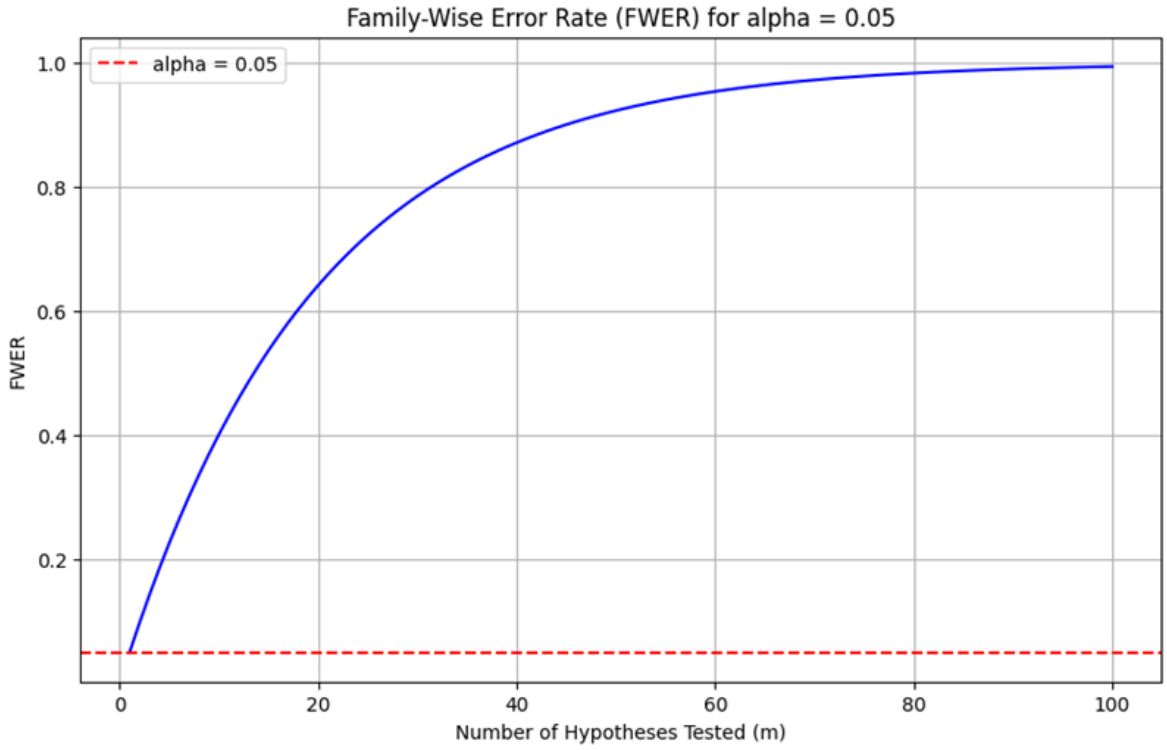
Die SADD beschreibt den Einsatzbereich als stochastischen Sampling-Prozess, wodurch Leistungsmetriken als Erwartungswerte über eine Verteilung interpretiert werden. Diese Formalisierung ist entscheidend: Ohne eine explizite Definition der Datenverteilung sind Testergebnisse nicht interpretierbar. Gleichzeitig ermöglicht sie reproduzierbare und vergleichbare Tests und schafft die Basis für unabhängige Prüfungen durch Dritte. Anstelle fixer Testsets tritt somit ein kontrollierter Sampling-Mechanismus, der Overfitting und Informationslecks systematisch verhindert.

Darauf aufbauend übersetzen Minimum Performance Requirements (MPR) die Ergebnisse einer Risikoanalyse in konkrete, messbare Anforderungen. Risiken werden dabei nicht nur qualitativ beschrieben, sondern quantitativ operationalisiert. Dies umfasst die Definition geeigneter Metriken sowie die Festlegung minimal akzeptabler Schwellenwerte, beispielsweise in Form von Fehlerraten unter vorgegebenen Konfidenzniveaus. Erst durch diese Formalisierung wird eine objektive Bewertung der Systemleistung möglich.

Ein wesentliches Element kontinuierlicher Qualitätssicherung ist die statistisch valide Überprüfung dieser Anforderungen – sowohl vor der Inbetriebnahme als auch während des laufenden Betriebs. Da KI-Systeme wiederholt getestet werden (etwa nach Updates, Re-Training oder bei erkannten Veränderungen im Einsatzkontext), reicht die Anwendung klassischer, einmaliger Hypothesentests nicht aus. Wiederholte Tests führen zu einer Inflation der Fehlerwahrscheinlichkeit, der sogenannten Family-Wise Error Rate (FWER). Dieses

Verhalten lässt sich anschaulich daran erkennen, dass bereits bei einer moderaten Anzahl wiederholter Tests die Wahrscheinlichkeit, mindestens einen Fehlentscheid zu treffen, schnell gegen 1 steigt – selbst wenn das Signifikanzniveau einzelner Tests konstant bleibt.

Abbildung 4 Family-Wise Rerror Rate



Um dennoch belastbare Garantien zu gewährleisten, sind sequentielle Testverfahren erforderlich. Diese verteilen ein fixes Gesamt-Signifikanzniveau über eine Folge von Tests und ermöglichen so eine statistisch korrekte Re-Zertifizierung auch im laufenden Betrieb. Entsprechende Verfahren stellen sicher, dass die Gesamtaussagekraft über den Lebenszyklus hinweg erhalten bleibt.

Methodisch lässt sich die kontinuierliche Qualitätssicherung entlang eines strukturierten Vorgehens verankern: Zunächst wird der Zweck des KI-Systems und sein Einsatzbereich klar definiert. Darauf folgt eine systematische Risikoanalyse, aus der überprüfbare Qualitätsziele abgeleitet werden. Die anschließende Formalisierung der Anwendungsdomäne (SADD) schafft die Grundlage für valide Tests. Datenmanagement, Training und Validierung erfolgen kontrolliert und zielgerichtet, um Verzerrungen und Informationslecks zu vermeiden. Schließlich werden die definierten Anforderungen statistisch geprüft und im Betrieb kontinuierlich überwacht. Bei den zu erwartenden dynamischen Veränderungen (neue Daten,

Modellanpassungen, veränderten Einsatzbedingungen, etc.), ist es ebenso erforderlich, regelmäßig die Aktualität der Risikoanalyse und die Formalisierung der Anwendungsdomäne und somit auch des Testsets zu überprüfen und gegebenenfalls anzupassen.

Neben der statistischen Bewertung bildet die prozessuale Absicherung entlang des KI-Lebenszyklus eine zweite zentrale Säule der Qualitätssicherung. Dazu zählen Maßnahmen wie strukturiertes Datenmanagement, kontrollierte Trainingsprozesse, Monitoring im Betrieb sowie menschliche Aufsicht. Erst das Zusammenspiel beider Dimensionen – statistische Evidenz und robuste Prozesse – ermöglicht eine nachhaltige Sicherstellung von Qualität und ist zugleich Voraussetzung für eine spätere Zertifizierung, insbesondere im Kontext regulierter Hochrisiko-Anwendungen.

Je nach Realisierungsvariante eines KI-Systems ergeben sich unterschiedliche Ansatzpunkte für die Qualitätssicherung. Beim Training von Basismodellen bestehen die größten Eingriffsmöglichkeiten, da sowohl Trainingsdaten als auch Trainingsprozess vollständig kontrolliert werden können – allerdings bei gleichzeitig hoher Komplexität und praktischen Grenzen hinsichtlich Datenqualität und -menge. Bei der Modelladaption (z. B. Finetuning) liegt der Fokus stärker auf der gezielten Steuerung zusätzlicher Trainingsdaten sowie auf nachgelagerten statistischen Tests, um verbleibende Unsicherheiten einzugrenzen. Bei der Integration von Retrieval-Augmented Generation (RAG) verschiebt sich die Qualitätssicherung zusätzlich in den Betrieb: Neben der statistischen Bewertung zentraler Zuverlässigkeitsparameter gewinnen kontinuierliche Überwachung, Bias-Korrektur sowie sicherheitsorientierte Verfahren wie Red Teaming oder adversariales Testen an Bedeutung.

Insgesamt zeigt sich, dass kontinuierliche Qualitätssicherung kein optionaler Zusatz, sondern eine grundlegende Voraussetzung für den verantwortungsvollen Einsatz von KI darstellt. Sie verbindet mathematisch fundierte Prüfverfahren mit organisatorischen und prozessualen Maßnahmen und schafft damit die Grundlage für belastbares Vertrauen in KI-Systeme – sowohl aus technischer als auch aus regulatorischer Perspektive.

Digitale Souveränität durch PPP-Strukturen

Zwei Kernaspekte digitaler Souveränität in KI-basierten Systemen sind Datensouveränität und Betriebssicherheit.

Datensouveränität

Bei der **Einbindung von LLMs in größere Systemkontexte** (z.B. RAG-Systeme) sind sowohl die Trainingsdaten für das/die eingesetzten LLMs als auch die Daten, die bei Anfragen an ein LLM als Kontext mitgegeben werden, zu betrachten (Tabelle 36). Da ein bereits vorhandenes LLM eingebunden wird, hat man in dieser Variante keinen Einfluss auf die Daten, mit denen das LLM trainiert wurde.

Volle Souveränität bzw. Kontrolle über die Trainingsdaten eines LLMs kann nur von denen erzielt werden, die das Modell trainieren. Der Wunsch nach Kontrolle über die Trainingsdaten ist ein wesentlicher Motivator für die Entwicklung nationaler Foundation Models wie z.B. Apertus (Schweiz) oder GPT-NL (Niederlande). Bei Open Source Modellen werden deutlich mehr Informationen über ihre Trainingsdaten als bei proprietären Modellen bekannt gegeben. Bei letzteren werden mittlerweile typischerweise keine Informationen über die verwendeten Trainingsdaten geben. Die Transparenz bei Open Source Modellen reicht in der Regel von einer allgemeinen Dokumentation der Datenquellen bis hin zur Veröffentlichung der kuratierten, gefilterten Datensätze selbst. Je nach Detailgrad der gegebenen Informationen lässt sich besser oder schlechter nachvollziehen wie die Trainingsdaten zusammengesetzt sind und nach welchen Kriterien sie kuratiert sind. Siehe Tabelle 38.

Volle Kontrolle herrscht hingegen hinsichtlich der Kontextdaten, die in einem RAG-System dem LLM mitgegeben werden. Dies erfordert eine sorgfältige Kuratierung und Wartung der Daten in den eigenen Vektordatenbanken. In agentischen KI-Systemen nimmt die Kontrolle über die Daten signifikant ab, bzw. es müssen gezielt Kontrollmechanismen eingebaut werden, da agentische Systeme selbständig übergeordnete Ziele verfolgen, diese in Teilaufgaben zerlegen, Werkzeuge (Tools) nutzen, eigenständig planen und Entscheidungen treffen,

um das Ziel zu erreichen. Mit dem Tool-Zugriff steigen auch die Sicherheitsanforderungen erheblich — insbesondere gegen indirekte Prompt-Injection — und es zeichnen sich Standardisierungsbemühungen wie das Model Context Protocol (MCP) ab, die für interoperable Tool-Anbindung in der Verwaltung relevant werden können.

Wird ein bestehendes Modell auf eigenen Daten angepasst (**Fine-Tuning**), gilt für das zugrunde liegende LLM, was oben zu LLMs gesagt wurde. Für die zum Fine-Tuning verwendeten Daten, hingegen, gilt volle Kontrolle. Wichtig ist, dass die Daten möglichst gut kuratiert und an die Aufgabe(n) des adaptierten LLMs angepasst sind. Siehe Tabelle 37.

Während die Datenmengen für RAG-Kontexte als auch für parametereffizientes Finetuning (z.B. mit LoRA) eher gering sind und sich der Datenkuratierungsaufwand in Grenzen hält, sind die für das Foundation-Model-Training erforderlichen Datenmengen enorm, was einen entsprechend hohen Datenbeschaffungs- und -kuratierungsaufwand erfordert.

Tabelle 36 Einbindung von LLMs in größere Systemkontexte (z.B. RAG)

Daten in LLM	Kontext-Daten	Datenmenge	Datenkuratierungsaufwand
Kein Einfluss auf Daten	Volle Kontrolle	Gering ⇔ Mittel	Mittel
Info nur bei Open Source Modellen	Müssen gut kuratiert sein!		

Tabelle 37 Finetuning mit LoRA

Daten in LLM	Trainingsdaten	Datenmenge	Datenkuratierungsaufwand
Kein Einfluss auf Daten	Volle Kontrolle	Gering ⇔ Mittel	Mittel
	Info nur bei Open Source Modellen	Müssen gut kuratiert sein!	

Tabelle 38 Foundation Model Training

Trainingsdaten	Datenmenge	Datenkuratierungsaufwand
Volle Kontrolle	Sehr hoch	Sehr hoch
		Die Datenqualität beeinflusst maßgeblich die Modellqualität!

Betriebssicherheit

Ein weiterer wesentlicher Aspekt von Souveränität ist Betriebssicherheit. Betriebssicherheit kann auf verschiedene Arten hergestellt werden. **Hohe Souveränität** ist über den On-Premise-Betrieb zu erreichen, wobei auch das/die verwendeten LLMs lokal, in abgeschotteten, sicheren Umgebungen laufen müssen. Auch beim Betrieb über die Cloud kann hohe Betriebssicherheit erreicht werden, vorausgesetzt es handelt sich um eine souveräne Cloud in der EU von einem EU-Betreiber.

Einschränkungen der Betriebssouveränität: Vorsicht ist geboten bei Clouds, die von US-Anbietern betrieben werden, selbst wenn die Clouds in Europa betrieben werden. Denn US-Betreiber unterliegen dem US Cloud Act: dieser erlaubt US-Behörden den Zugriff auf Daten, selbst wenn diese in der EU gespeichert sind (cf. IONOS White Paper, 2021). Bei der Verwendung chinesischer cloudbasierter Applikationen ist zu bedenken, dass der Datenfluss nach China nicht kontrollierbar ist und mit staatlichen Online-Filtern und eventueller Zensur von Inhalten zu rechnen ist. Des Weiteren ist bei der Verwendung von kostenlosen Online-Tools sowie bei Anbietern aus autoritären Drittstaaten von einer sehr geringen Betriebssicherheit auszugehen.

Neben der Vorherrschaft von amerikanischen (hauptsächlich kommerziellen) Anbietern von LLMs, wie z.B. OpenAI, Google, Amazon, Anthropic, Meta, xAI haben sich chinesische Anbieter besonders bei frei zugänglichen Open-Weight-Modellen hervorgetan, siehe z.B. DeepSeek-V4, Qwen3.6, GLM-4, Kimi K2.6. Entsprechend werden chinesische Anbieter für on-premise eingesetzte LLMs immer mehr ein Thema.

Um die **Sicherheit von LLMs** zu bewerten gibt es eine Reihe von Benchmarks und zugehörige Leaderboards wie z.B. Cisco LLM Security, Hydro X LLM Safety, F5 Labs AI Security

Leaderboards, Huggingface LLM Safety Leaderboard.9F9 Die Benchmarks unterscheiden sich in dem, was sie abtesten und sind daher nicht direkt vergleichbar, können aber einen ersten groben Eindruck zu Sicherheitsaspekten in einzelnen LLMs geben. Das erspart aber nicht die Notwendigkeit eigene, applikationsspezifische Tests zu entwickeln und laufend durchzuführen und sich gegen Risiken bei der Verwendung von LLMs abzusichern (Greshake et al., 2023; NIST, 2024; Xue et al., 2024; OWASP 2024, 2026).

PPP-Strukturen

Aufgrund der Komplexität KI-basierter Systeme ist für eine effiziente und zielführende Umsetzung das gut abgestimmte Zusammenwirken verschiedener Stakeholder unerlässlich.

Dazu gehören:

- Domänenexpert:innen, User:innen und Datenverantwortliche aus den einzelnen Wirkungsbereichen in der öffentlichen Verwaltung,
- Softwareingenieur:innen und Forscher:innen mit Schwerpunkt KI-basierte Systeme,
- Softwareingenieur:innen mit Schwerpunkt Entwicklung und 24/7-Betrieb von Applikationen für die öffentliche Verwaltung,
- Expert:innen aus der KI-Forschung, wenn es um Wissensdestillation, Parameter Efficient Finetuning bzw. das Training von Basismodellen geht, wobei es deutliche Unterschiede hinsichtlich der Anforderungen an Personal- und Rechenressourcen zwischen dem Training großer Multipurpose-Modelle und kleinerer, spezialisierter Modelle gibt.
- Während sich Einrichtungen wie die AI-Factories (AI:AT in Österreich) für die zeitlich begrenzte Entwicklung von forschungs- und rechenintensiven (HPC-Infrastruktur) Prototypen anbieten, braucht die Umsetzung in ein Produktionssystem und der tägliche Betrieb Einrichtungen wie das BRZ. Auch hier gilt die enge Verzahnung zwischen den Stakeholdern, da schon in der Prototypenentwicklung der Betrieb des Produktionssystems mitgedacht werden muss und domänen- sowie KI-spezifische Expertisen von Anfang an eingebracht werden müssen.

[9] leaderboard.aidefense.cisco.com/rankings, hydrox.ai/leaderboard, f5.com/labs/casi, huggingface.co/spaces/AI-Secure/llm-trustworthy-leaderboard

- In wie weit On-Premise-Lösungen, der Einsatz einer souveränen nationalen oder Europäischen Cloud zu bevorzugen ist, hängt von verschiedenen Faktoren ab, dazu gehören u.a. der erforderliche Grad an Daten- und Betriebssouveränität, die Anzahl der User:innen und die Anzahl der Anfragen da das oder die LLMs, die Anforderungen an die Leistungsfähigkeit des bzw. der eingesetzten LLMs wie z.B. Reasoningkapazitäten, extra große Kontexte, Bedarf an Multimodalität.
- Sowohl die Entwicklung eines gemeinsamen KI-Stacks für die öffentliche Verwaltung als auch die Entwicklung einzelner Applikationen bedarf der Begleitung von Rechtsexpert:innen mit Schwerpunkt auf KI-spezifische Fragen wie Datenschutz und AI Act.
- Des Weiteren muss bei der Entwicklung der Systeme die kontinuierliche Qualitätssicherung und die Anforderungen an eine spätere Zertifizierung im Sinne des AI Acts inklusive dem Monitoring KI-ethischer Aspekte mitgedacht werden.
- Ebenso flankierend zu Systementwicklung und -betrieb sind KI-spezifische Aspekte von Safety und Security zu beachten.

Synthese und Strategische Empfehlungen

Roadmap

Im Folgenden werden Entwicklungsphasen für eine national abgestimmte KI-Referenzarchitektur skizziert, wobei auf verschiedene technische Umsetzungsvarianten – Einbindung bestehender LLMs in größere Systeme, Modelloptimierung und Basismodelltraining – eingegangen wird. Bei jeder Phase sind Fragen der Umsetzbarkeit, Souveränität, Anforderungen an Humanressourcen und Rechenkapazitäten, Qualitätssicherung, ethische und rechtliche Themen sowie ökologische Auswirkungen mitzuberücksichtigen (Details siehe bei den Beschreibungen der jeweiligen Realisierungsvarianten im vorderen Teil dieses Dokuments.)

Phase 1 (KI-Stack -- Basis)

Im Sinne einer effizienten Umsetzung KI-basierter Applikationen in der öffentlichen Verwaltung, bei der frühest möglich für die öffentliche Verwaltung nutzbringende Ergebnisse erzielt werden können, empfehlen wir ein phasenorientiertes Vorgehen bei dem zuerst die technischen und prozeduralen Grundlagen geschaffen werden, um LLMs in flexibler, sicherer und rechtskonformer Weise in größere Systemkontexte und Anwendungszusammenhänge einzubauen und einen gesicherten 24/7 Betrieb zu gewährleisten. Anmerkung: Dieser Prozess wurde bereits mit dem vom BKA initiierten GovGPT gestartet, sowie dem Umsetzungsplan für ausgewählte Applikationen in der öffentlichen Verwaltung im laufenden Jahr 2026, inkl. der Entwicklung des „LLM as a service“ durch das BRZ.

Phase 2 (KI-Stack -- Flexible Erweiterbarkeit)

Parallel dazu und basierend auf den bereits gemachten Erfahrungen dieser ersten Umsetzung eines gemeinsam nutzbaren KI-Stacks und den technischen Hintergrunddokumenten (techn. Realisierungsvarianten und Ausbaustufen, ethische, rechtliche und ökologische Aspekte) aus der „Bundes-LLM“ Studie schlagen wir eine detaillierte Ausarbeitung eines technischen und prozeduralen Konzepts für die systematische Erweiterung des KI-Stacks für die öffentliche Verwaltung vor. Dazu ist der Austausch bzw. die Zusammenarbeit von Stakeholdern essentiell, wie z.B. den Domänenexpert:innen in den Wirkungsbereichen, dem BRZ, KI-

Forschung, Systementwicklung und AI:AT. Diese Arbeit führt direkt in die Umsetzung des erweiterbaren Stacks, auf Basis dessen, was bereits technisch umgesetzt wurde. Benötigte Personalressourcen und -kompetenzen, Hardware- und Softwareanforderungen und damit verbundene Kosten lassen sich aus den Ausführungen zu den Realisierungsvarianten im vorherigen Abschnitt dieses Dokuments ableiten.

Phase 3 (Modelloptimierung)

Dazu gehören die Anpassung bestehender Modelle (i) auf spezifische Anforderungen aus den Wirkungsbereichen, die über die Einbindung bestehender Modelle nicht erreicht werden kann; (ii) lokale, spezialisierte Agenten, die effizient und kostengünstig lokal betrieben werden können. Dabei ist zwischen Finetuning-Ansätzen und Wissensdestillation (Knowledge Distillation) zu unterscheiden. Beim Finetuning geht es um die Anpassung eines vortrainierten LLMs durch Training mit domänenspezifischen Daten (z. B. Verwaltungsanweisungen). Bei der Wissensdestillation werden kleinere, lokal einsetzbare Modelle erstellt, die Eigenschaften bzw. Kapazitäten von größeren 'Teacher'-Modellen erben.

Für die Modelloptimierung kann je nach Anforderung eine Kombination aus Finetuning und Wissensdestillation sinnvoll sein, z.B.: beginnend mit dem Finetuning eines (kleineren) Teacher Modells mittels Methoden der Low Rank Adaptation (LoRA) auf domänenspezifischen Daten, dann die Verwendung des LoRA-adaptierten Teacher für Wissensdestillation auf ein kleineres bzw. effizienteres Student-Modell und eventuell weitere LoRA auf dem Student-Modell für zusätzliches Finetuning. Für die Auswahl geeigneter Methoden und deren optimaler Konfiguration ist das Verständnis der theoretischen Grundlagen unerlässlich.

Phase 4 (Basismodelltraining)

Die Entwicklung eines nationalen Foundation Modells wie z.B. dem Schweizer Modell Apertus oder dem niederländischen Modell GPT-NL ist zeit- (minimum 2 Jahre Entwicklungszeit) und ressourcenaufwändig (Daten, Speicherplatz, Rechenleistung, Energiekosten, Personal) und entsprechend kostspielig. Des Weiteren müssen die Modelle gewartet und regelmäßig retrainiert werden, um ihr Wissen auf dem aktuellen Stand zu halten. Soll ein Basismodell kommerzialisiert oder auch nur zentral gehostet werden, muss eine Architektur aufgebaut werden, die Training, Hosting, Skalierung und gegebenenfalls Abrechnung zuverlässig vereint und eine entsprechende Rechenzentrumsinfrastruktur muss aufgebaut, betrieben und gewartet werden.

Im Gegensatz zum Training eines sehr großen Modells (LLM) kann das Training eines Small Language Models (SLM) von Grund auf für sehr spezialisierte Anforderungen wirtschaftlich und technisch sinnvoll sein, besonders dann, wenn (i) bestehende Modelle bestimmte spezifischen Anforderungen prinzipiell nicht erfüllen können, wie z.B. in hochspezialisierten Domänen oder wenn die Daten in einer Fachsprache vorliegen, die in allgemeinen Trainingsdaten kaum bzw. nicht vorkommt; (ii) wenn mit streng vertraulichen Daten gearbeitet werden soll, die niemals mit fremden Basismodellen (auch nicht per Finetuning) in Berührung kommen dürfen; (iii) wenn das Modell direkt auf Kleinstgeräten (Smartphones, Sensoren, Wearables) laufen soll, muss die Architektur von Beginn an auf minimale Parameterzahl optimiert werden. Ebenso wie für Phase 3, erfordert Phase 4 tiefgreifendes Verständnis der theoretischen Grundlagen. Beide Phasen sind eng mit entsprechender Grundlagenforschung verknüpft.[10]

Die Phasen 1 und 2 sind der Entwicklung und Implementierung eines gemeinsamen, rechts- und betriebssicheren, flexibel erweiterbaren KI-Stacks und der Integration bestehender LLMs gewidmet und haben einen starken Engineering- und Prozessmodellierungs-Fokus.

Die weiteren Phasen befassen sich mit der Optimierung von bestehenden Modellen (Phase 3) und der Entwicklung kleinerer, spezialisierter von Grund auf trainierter Basismodelle (Phase 4). Die Phasen 3 und 4 sind in ihrer Entwicklung stark forschungslastig, wobei Expertise in Wissensdestillation, Parameter Efficient Finetuning und Modelltraining sowie Expertise hinsichtlich der Prozesse und Anforderungen aus der Anwendungsdomäne zusammenspielen müssen. Dabei ist auch die Definition und Einrichtung der Rolle von Datenverantwortlichen von besonderer Bedeutung. Diese Personen sind für die Bereitstellung und Wartung der für bzw. in der Anwendung benötigten Daten zuständig.

Um den maximalen Nutzen der Anwendungen zu gewährleisten, ist die Einbindung der Systemnutzer:innen von Anfang an und in allen Phasen der Entwicklung erforderlich (userzentriertes Design). Dies kommt auch einem

[10] In diesem Zusammenhang sei auch das österr. Cluster of Excellence Bilateral Artificial Intelligence (BILAI, fwf.ac.at/en/research-radar/10.55776/COE12) erwähnt, in dem Grundlagenforschung betrieben wird, um ein LLM zu entwickeln, das symbolische Tools wie z.B. Solvers versteht und somit die Möglichkeit eröffnet symbolische und subsymbolische Methoden miteinander zu vereinen.

angemessenen Usererwartungsmanagement zugute. Mit anderen Worten, für die Umsetzung aller Phasen multi-disziplinäre Teams unerlässlich sind.

Beispielarchitekturen

Referenzarchitektur gemeinsamer KI-Stack

Im Folgenden präsentieren wir einen Überblick zu einer Referenzarchitektur eines gemeinsamen KI-Stacks. Referenzarchitekturen für LLM-Stacks werden in verschiedenen Veröffentlichungen beschrieben (z.B. Radovanovic & Bornstein, 2023; Hitzel, 2025; Stone et al., 2025; Copei et al., 2025; Hou et al., 2025). Diese Beschreibung versucht, die wesentlichen Aspekte auf einer hohen Abstraktionsebene zu beschreiben. Die präsentierte Referenzarchitektur trennt Anwendung, Orchestrierung, Wissenszugriff, Modellbetrieb, Governance und Infrastruktur, weil moderne LLM-Anwendungen nicht nur aus einem Modell bestehen, sondern aus einem Zusammenspiel von Modellen, Daten, Tools, Workflows, APIs, Sicherheitsmechanismen und Monitoring. Als Leitprinzip gilt: Das Modell ist austauschbar beziehungsweise dem jeweiligen Anwendungsfall entsprechend, die Plattform bleibt stabil. Neue Modelle, RAG-Varianten, Agenten, Fine-Tuning-Verfahren oder Wissensdestillation sollen ergänzt werden können, ohne dass Fachanwendungen komplett neu gebaut werden müssen. In weiterer Folge werden die einzelnen Komponenten der vorgeschlagenen Referenzarchitektur besprochen:

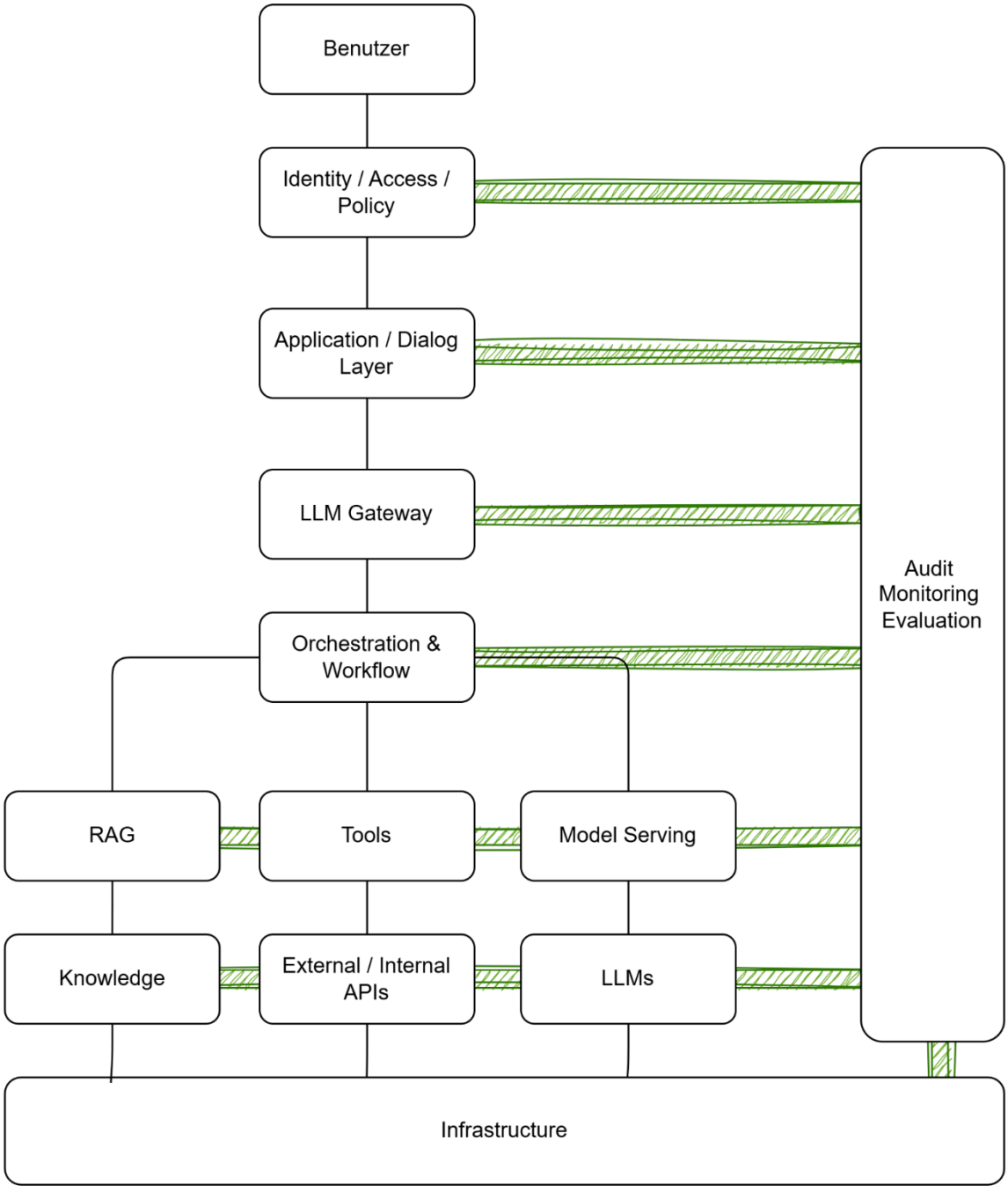
Application und Dialog Layer: Diese Schicht stellt die nutzbaren Anwendungen bereit: Chat, Fachassistenten, Recherchewerkzeuge, Dokumentenprüfung, API-Zugriff (REST etc.) und eingebettete Funktionen in Fachsystemen.

Identity, Access und User Context Layer: Diese Schicht regelt, wer welche Modelle, Daten, Werkzeuge und Funktionen nutzen darf. Dazu gehören Identity Provider wie OAuth2, ein Rollen- und Rechtemodell, mögliche Mandantenfähigkeit sowie Policy Enforcement.

LLM Gateway: Das Gateway ist die zentrale Kontrollstelle zwischen Anwendungen und Modellen. Es verhindert, dass jede Anwendung direkt mit einem Modell verbunden ist. Dazu gehören unter anderem: Modell-Routing, Tokenbasierte Abrechnung sowie Sicherheitsfilter (Erkennung von Prompt Injection oder Information Leakage).

Orchestration und Workflow Layer: Diese Schicht entscheidet, wie eine Anfrage bearbeitet wird: direktes LLM, RAG, Tool-Aufruf, Agentenworkflow, Klassifikation oder menschliche Freigabe. Dazu gehören Prompt- und Template-Management, Workflow-Orchestrierung agentische Erweiterungen sowie ein Human-in-the-loop Ansatz bei kritischen Aktionen.

Abbildung 5 Referenzarchitektur: gemeinsamer KI-Stack



RAG: Da RAG-Systeme ein weit verbreiteter Use Case sind, sind sie im Modell extra angeführt. Im Wesentlichen ist dieser Teil dafür verantwortlich, Informationen aus der Knowledge-Schicht zu holen und den Kontext mit Prompt und Informationen anzureichern für die Generierung mit dem LLM. Ein weiterer Aspekt ist die Vorbereitung von neuem Wissen für das System. Dazu gehören die Aufnahme und Verarbeitung von Dokumenten und die Übergabe der dabei entstehenden Informationen an die Knowledge-Schicht zur Speicherung.

Knowledge (und Retrieval Layer): Diese Schicht liefert dem System kontrolliertes, aktuelles und berechtigungsgeprüftes Wissen. Dazu gehören z.B. Abfragen von Vektor-, Graph- oder andere Datenbanken, ein Retrieval-Ansatz, der relevante Informationen liefert (z.B. über einen hybriden Ansatz mit semantischer und Volltextsuche, Reranking und Berücksichtigungen von Metadaten und Aktualität).

Model Serving und Model Management Layer: Diese Schicht betreibt die Modelle robust, skalierbar und austauschbar. Diese Schicht verwaltet die verschiedenen Modelltypen (instruct, reasoning, multimodal, etc.) sowie Informationen über die Modelle an sich oder auch den KV-Cache. Reasoning-Modelle erzeugen vor der Antwort interne Chain-of-Thought-Token und erhöhen Tokenverbrauch und Latenz je Anfrage um ein Vielfaches; ihr Einsatz ist daher auf reasoning-intensive Tasks (z.B. juristische Subsumtion) zu beschränken.

Tool, Action und Protocol Layer: Diese Schicht erlaubt kontrollierte Aktionen außerhalb der Modelle, z.B. Datenbankabfragen, Berechnungen, Fachsystemzugriffe oder Agent-zu-Agent-Kommunikation.

Audit/Monitoring/Evaluation: Diese Schicht bildet Querschnittsthemen im Kontext von Governance, Security, Auditing, Monitoring und Operations ab. Diese Querschnittsschicht stellt sicher, dass der gesamte AI-Stack kontrolliert, regelkonform und nachvollziehbar betrieben wird. Verschiedene Aspekte werden in verschiedenen Schichten des Systems behandelt. Zu den Aspekten, die zu dieser Schicht gehören, sind z.B. die folgenden zu zählen: Modellfreigabe, Promptmanagement, auditierbares Logging der verschiedenen Vorgänge, Monitoring von verschiedenen Aspekten des Systems, Überwachung der Leistungsfähigkeit des Systems (Durchsatz; Latenz -- getrennt nach Time-to-First-Token und Inter-Token Latency; Ressourcenverbrauch etc.), KI-spezifisches Monitoring wie z.B. Halluzinationsindikatoren oder Antwortqualität, Red-Teaming und Safety-Test.

Infrastructure Layer: Diese Schicht stellt die technische Basis bereit. Dazu gehören GPUs, CPUS, RAM, persistenter Speicher (Filesystem, CEPH, etc.), Containerisierung per z.B.

Kubernetes, Datenbanken, Netzwerk, Backup- und Ausfallsicherheit sowie weitere Themen wie SLAs. Darunter ist auch die Anbindung externer Cloud-Systeme zu sehen, wenn Modelle oder andere Services dort betrieben werden.

Referenzarchitektur hybrider Arbeitsprozess

Im Folgenden wird eine Architektur für einen hybriden Arbeitsprozess beispielhaft beschrieben, bei dem je nach Anfrage bzw. Anforderung unterschiedliche LLMs aus einem Pool spezialisierter LLMs angesteuert werden können. Über ein API-Gateway inklusive Identitätscheck und rollenbasierter Zugriffskontrolle (RBAC) wird die Anfrage an einen Taskrouter weitergegeben, wobei auch abgeschätzt wird, in wie weit persönliche Information in der Anfrage bzw. dem Prozess der Anfragebeantwortung involviert ist (PII Classifier). Der Task Router verteilt die Anfrage dann auf verschiedene Experten-Module (General Assistant, …, Legal Assistant) und diese greifen gezielt auf spezialisierte Modelle zurück (Specialist Model Pool).

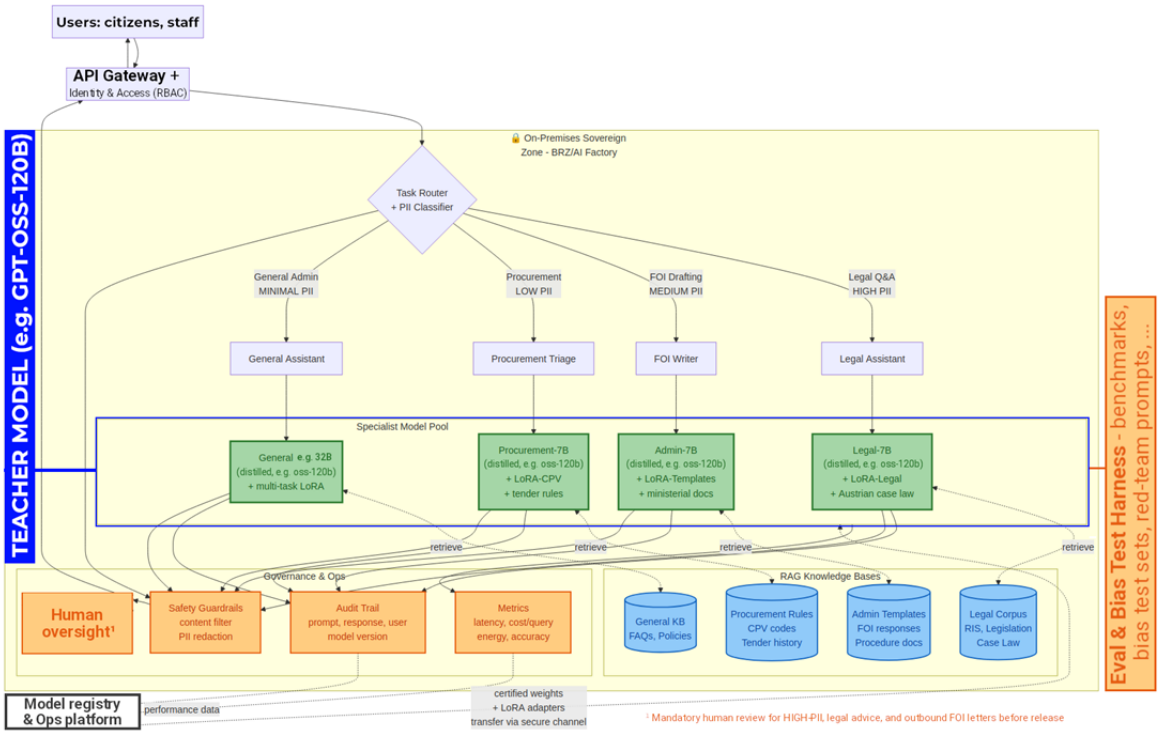
Im Specialist Model Pool befinden sich Modelle, die für den On-Premise-Betrieb optimiert wurden. Im konkreten Beispiel sind alle Modelle von einem größeren Teacher Modell (GPT-OSS 120B) auf ein kleineres Student Modell destilliert und für einen bestimmten Aufgabenbereich mittels LoRA angepasst worden. GPT-OSS 120B wurde im Beispiel als Teacher Modell ausgewählt, weil es eine vergleichsweise hohe Leistungsfähigkeit aufweist und es ein Open-Weights-Modell ist, d.h. die Modellgewichte frei zugänglich sind, was Modellanpassungen und lokale Nutzung bzw. Nutzung in einer eigenen Cloud ermöglicht. Somit kann das Modell DSGVO-konform betrieben werden. Die Modellgewichte sind unter der Apache 2.0 Lizenz veröffentlicht. Des Weiteren ist GPT-OSS 120B aufgrund großer Kontextlänge für Anwendungen in der Verwaltung, wo es um sehr lange Dokumente geht, gut geeignet. Jedoch ist zu bedenken, dass das Modell überwiegend auf englischsprachigen Daten trainiert ist. Für deutschsprachige Verwaltungsanwendungen sind europäische Alternativen wie Apertus-70B, EuroLLM-22B oder Mistral Large 2 als Teacher-Modelle ebenfalls zu prüfen und passen besser zum Souveränitätsanspruch der vorgeschlagenen Architektur. Die konkrete Wahl ist anwendungsfallabhängig und sollte vor Festlegung der Architektur empirisch validiert werden. Eine effiziente Modellarchitektur und die Auslegung für die Nutzung mit Quantisierung ermöglichen einen effizienten lokalen Betrieb. Durch Wissensdestillation kann das Modell noch weiter effizient gemacht werden. Die adaptierten Modelle müssen in ihrer Entwicklungsphase und auch während des Betriebs, insbesondere wenn sich die externen Daten, auf die zugegriffen wird, ändern, verschiedenen Evaluierungen und Tests unterzogen werden. Entsprechend ist in der Architektur ein Bereich für Testkomponenten

vorgesehen (Eval & Bias Test Harness). Ebenso sind Testkomponenten und -prozeduren vorzusehen, die das gesamte System testen und mittels derer überwacht werden kann, ob sich die Systemleistung in Laufe des Betriebs verändert, sei es z.B. aufgrund neuer Daten in den Wissensdatenbanken (RAG Knowledge Bases in der Beispielgraphik), veränderten Nutzer:innenverhaltens, des Austauschs von einzelnen Systemkomponenten.

Die Architektur ist so ausgelegt, dass das System souverän und DSGVO-konform on-premise betrieben werden kann, für den laufenden, stabilen und sicheren Betrieb im BRZ oder in einer souveränen Cloud. Für die Entwicklung des Systems, insbesondere für die Erstellung der Specialist Models bieten sich die Kapazitäten der AI Factory (AI:AT) an. Eine Komponente zur Regelung der Human Oversight ist im Governance-und-Operations-Bereich der Architektur vorgesehen.

Die vorgeschlagene Architektur gibt einen Überblick über die einzelnen benötigten Teilkomponenten. Deren konkrete Auswahl und die detaillierte Ausspezifizierung bzw. Umsetzung der einzelnen Komponenten hängt von den konkreten Anwendungen ab.

Abbildung 6 Referenzarchitektur: hybrider Arbeitsprozess



Referenzarchitektur Foundation Model Training

Im Folgenden wird eine Referenzarchitektur für das Training eines LLMs von Grund auf skizziert.

Die **Designphase** (Phase 0) legt das Fundament für alle nachfolgenden Phasen. Hier werden kritische Entscheidungen über Architektur, Tokenizer und Datensammlung getroffen, die den gesamten Trainingsprozess bestimmen.

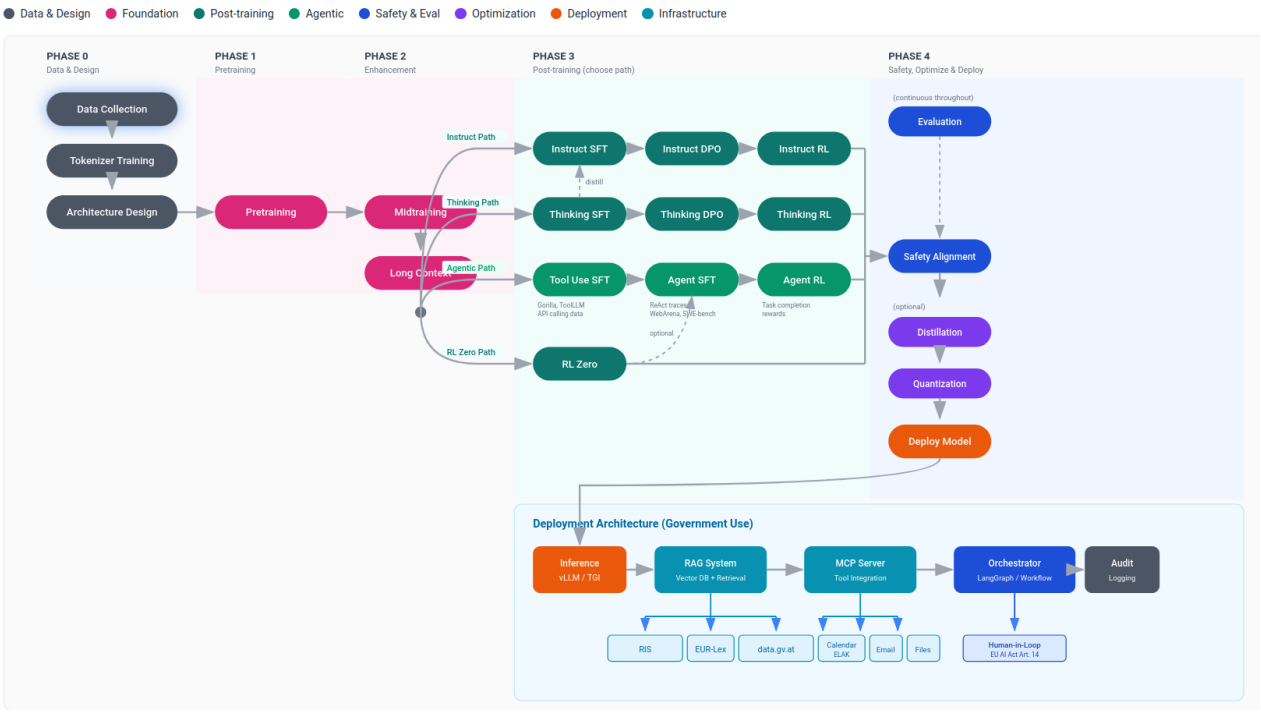
Das **Pretraining** (Phase 1) ist die rechenintensivste Phase und erfordert sorgfältige Planung bezüglich Scaling Laws, Trainingsstabilität und verteilter Infrastruktur.

Die **Enhancement-Phase** (Phase 2) umfasst Midtraining für Domain-Adaptation und Long-Context Extension, um das Basismodell für spezifische Anwendungsfälle zu optimieren.

Post-Training (Phase 3) transformiert das Basismodell in einen hilfreichen, harmlosen und ehrlichen Assistenten. Es gibt mehrere Pfade: den Instruct Path für Standard-Assistenten, den Thinking Path für Reasoning-Modelle, den Agentic Path für Tool-Use und den RL Zero Path für pure RL-basierte Reasoning-Entwicklung.

Die **finale Phase** (Phase 4) umfasst kontinuierliche Evaluation, Safety Alignment, optionale Distillation und Quantization sowie das eigentliche Deployment.

Abbildung 7 Referenzarchitektur: Foundation Model Training



Fazit aus der Studie

Die österreichische Bundes-LLM-Studie unterstreicht die **Notwendigkeit einer gemeinsamen, plattformbasierten Infrastruktur**, da **isolierte RAG-Systeme und LLM-as-a-Service-Lösungen** in den Ministerien **langfristig zu hohen Folgekosten, Ineffizienzen und technologischen Sackgassen führen**. Ein einheitlicher Software-Stack bündelt die Ressourcen, senkt die Gesamtbetriebskosten und sichert gleichzeitig souveräne, qualitativ hochwertige Anwendungen. Das begleitende Projektmanagement erfordert daher eine kostenbewusste, pragmatische Governance, die den Fokus strikt auf Wirtschaftlichkeit, Skaleneffekte und eine agile Umsetzung legt.

Das **Training eines nationalen österreichischen Foundation Modells** ist **unter ökonomischen Gesichtspunkten nicht empfehlenswert**, da es bereits eine Reihe von Open-Source- und Open-Weight-Modellen gibt, die für spezifische, nationale Zwecke adaptiert werden können, das Training von LLMs von Grund auf sehr kosten- und ressourcenintensiv ist, und die Qualität bestehender, aktiv gepflegter und weiterentwickelter Modelle aufgrund des Entwicklungsvorsprungs nur schwer (bei kleineren Modellen) bis nicht (bei den großen kommerziellen Modellen) und mit enormen Kosten erreichbar ist.

Das Ziel, um KI-basierte Systeme möglichst kosten effizient und zeitnah für die öffentliche Verwaltung zur Verfügung zu stellen, ist daher der **schrittweise Übergang von isolierten Einzellösungen zu einem gemeinsamen souveränen KI Stack**.

Phase 1: Prozessuale Voraussetzungen & Governance

Etablierung der Projekt-Governance: Einrichtung des unabhängigen, budgetverantwortlichen Projektmanagement-Gremiums zur Steuerung aller Ministerien.

Harmonisierung der Compliance: Definition einheitlicher Standards für Datenschutz (DSGVO-Konformität im Behördenkontext), Datensicherheit und Informationsklassifizierung (Geheimhaltungsstufen).

Kriterien für Qualitätssicherung (QS): Festlegung verbindlicher Metriken zur Messung von Halluzinationen, Antwortqualität und rechtlicher Korrektheit.

Bedarfs- und Ressourcenmatrix: Erfassung der konkreten Use Cases aller Ministerien zur Vermeidung von Doppelgleisigkeiten.

Phase 2: Technische Konzeption & Basis-Stack

Der Fokus liegt auf Modularität, um Herstellerunabhängigkeit (Vendor Lock-in vermeiden) und Erweiterbarkeit zu garantieren.

Zentrales API-Gateway: Aufbau einer Schnittstelle, über die Ministerien standardisiert Modelle anfragen können. Das bestehende BKA-System dient hier als erster Nukleus.

Modularer RAG-Standard-Stack: Entwicklung einer gemeinsamen Architektur für "Retrieval-Augmented Generation" (wiederverwendbare Vektordatenbanken, Standard-Parser für Ministeriumsdokumente).

Integrierte QS-Schicht: Implementierung einer vorgeschalteten Prüfinstanz, die Prompts und Modellantworten automatisch auf DSGVO-Verstöße, Falschinformationen oder unerlaubte Inhalte filtert.

Infrastruktur-Entscheidung: Bereitstellung einer hybriden Hosting-Umgebung (sichere Behörden-Cloud in Österreich für sensible Daten + optionale Schnittstellen zu kommerziellen Modellen für unkritische Aufgaben).

Phase 3: Phasen-Entwicklungsplan & Ausbaustufen

Stufe 1 (Konsolidierung): Migration bestehender RAG-Systeme der Ministerien auf den gemeinsamen Software-Stack zur unmittelbaren Senkung der Lizenz- und Betriebskosten.

Stufe 2 (Domänenspezifisches Fine-Tuning): Gezieltes Post-Training und Fine-Tuning von Open-Source-Basismodellen mit spezifisch österreichischen Rechtstexten, Verwaltungsdaten und dem Amtsdeutsch (Erhöhung der Präzision für Behördenprozesse).

Stufe 3 (Agentische Workflows): Übergang von reinen Frage-Antwort-Systemen zu autonomen Multi-Agenten-Systemen, die komplexe, ministerienübergreifende Prozesse (z. B. automatisierte Vorprüfung von Förderanträgen) selbstständig vorbereiten.

Glossar

Adversariales Testen (Adversarial Testing): Ein KI-Modell wird gezielt mit bösartigen, manipulierten oder extremen Eingaben gefüttert, um es zu Fehlern zu zwingen und somit Sicherheitslücken und Schwachstellen aufzudecken.

AGH (Auftraggeberhaftung für Sozialversicherungsbeträge)

AGI (Artificial General Intelligence): Allgemeine künstliche Intelligenz, die in der Lage ist, jede intellektuelle Aufgabe ähnlich oder besser als ein Mensch zu lösen. Im Gegensatz zu spezialisierter KI (Narrow AI) ist AGI derzeit noch nicht realisiert.

AI (Artificial Intelligence) bzw. **KI** (Künstliche Intelligenz)

Alignment: Bezeichnet die Ausrichtung eines KI-Systems auf menschliche Werte, Ziele und Sicherheitsanforderungen. Ziel ist es, unerwünschtes Verhalten oder schädliche Outputs zu vermeiden.

Apache 2.0 Lizenz: ist eine Software-Lizenz mit weitgehender Nutzungsfreiheit, für Details siehe apache.org/licenses/LICENSE-2.0

Attention Mechanism (Selbstaufmerksamkeit): Mechanismus, der es einem Modell erlaubt, für jedes Token die Relevanz anderer Tokens im Kontext zu gewichten und damit Kontext effizient zu berücksichtigen. Formal basiert Attention auf gewichteten Summen von Value-Vektoren, wobei die Gewichte durch Query-Key-Ähnlichkeiten bestimmt werden. (Vaswani et al., 2017)

Benchmark: Standardisierte Tests oder Datensätze zur Bewertung der Leistungsfähigkeit von KI-Modellen. Beispiele sind GLUE oder MMLU für Sprachmodelle.

Bias (Verzerrung): Systematische Verzerrung in Daten oder Modellen, die zu unfairen oder diskriminierenden Ergebnissen führen kann. Bias entsteht oft durch unausgewogene Trainingsdaten.

BOM (Bill of Materials, DE: Stückliste)

Byte Pair Encoding (auch BPE): Zu Deutsch Byte-Paar-Kodierung) ist ein Algorithmus zur Kodierung von Textsequenzen in kürzere Segmente durch die Erstellung und Nutzung einer Übersetzungstabelle. Eine leicht angepasste Version des Algorithmus wird in der Tokenisierung von LLMs verwendet. (Witten 1994)

CAPEX (capital expenditures): Investitionsausgaben für längerfristige Anlagegüter.

Chain-of-Thought (CoT) Prompting: Prompting-Technik, bei der das Modell explizit Zwischenschritte generiert, um komplexe Aufgaben zu lösen. CoT nutzt emergente Fähigkeiten großer Modelle und verbessert besonders reasoning-intensive Tasks. (Wei et al., 2023)

Context Window: Die maximale Menge an Text (Token), die ein Modell gleichzeitig berücksichtigen kann. Ein größeres Kontextfenster ermöglicht komplexere und längere Interaktionen.

CPU (Central Processing Unit): Hauptprozessor im Computer

CRA (Cyber Resilience Act): Cyberresilienz-Verordnung des Europäischen Parlaments und des Rates.

DevOps (Development Operations): DevOps ist eine Softwareentwicklungs-methodik, die die Bereitstellung von Hochleistungsanwendungen und -diensten beschleunigt, indem sie die Arbeit von Softwareentwicklungs- (Dev) und IT-Betriebsteams (Ops) kombiniert und automatisiert. (zitiert von ibm.com/de-de/think/topics/devops; letzter Lookup 12.6.2026)

Data Governance Act (DGA): Ziel des DGA ist die Förderung des Teilens von Daten öffentlicher Stellen.

Diffusion Model: Generatives Modell, das Daten schrittweise aus Rauschen rekonstruiert. Häufig für Bildgenerierung verwendet (z.B. Stable Diffusion).

DSG (Datenschutzgesetz)

DSGVO (Datenschutzgrundverordnung)

Embedding (Vektorrepräsentation): Abbildung von diskreten Daten (z.B. Tokens) in kontinuierliche Vektorräume, sodass semantische Ähnlichkeiten als geometrische Nähe modelliert werden. Embeddings sind Grundlage für Retrieval, Clustering und Ähnlichkeitssuche. (Mikolov et al., 2013)

Epoch: Ein vollständiger Durchlauf des Trainingsdatensatzes während des Trainings eines Modells. Mehrere Epochen verbessern typischerweise die Modellleistung.

ERF (Energy Reuse Factor)

EuGH (Europäischer Gerichtshof)

Fine-Tuning (inkl. PEFT): Nachtrainieren eines vortrainierten Modells an spezifische Aufgaben/Daten durch weiteres Training. Kann vollständig oder parametereffizient erfolgen. Parameter-efficient Fine-Tuning (PEFT) reduziert Ressourcenbedarf durch gezielte Anpassung kleiner Parameteranteile.

Freedom of Information (FOI) Anfragen: Informationsanfragen, in Österreich sind sie seit dem 1. September 2025 durch das neue Informationsfreiheitsgesetz (IFG) geregelt, cf. bundeskanzleramt.gv.at/themen/informationsfreiheitsgesetz.html.

Generative AI / generative KI: KI-Systeme, die neue Inhalte wie Text, Bilder oder Code erzeugen können. Basieren häufig auf großen neuronalen Netzen wie Transformern oder Diffusionsmodellen.

GPU (Graphics Processing Unit): Grafikverarbeitungseinheit

Hallucination (Modellhalluzination): Systematisches Generieren plausibler, aber falscher Informationen. Technisch entsteht dies durch die probabilistische Natur autoregressiver Modelle und fehlende Grounding-Mechanismen. (Lin et al., 2022)

IFG (Informationsfreiheitsgesetz): regelt den Zugang zu Information.

Inference: Der Prozess der Anwendung eines trainierten Modells auf neue Daten, um Vorhersagen oder Antworten zu generieren. Erfolgt typischerweise deutlich schneller als das Training.

IWG (Informationsweiterverwendungsgesetz): Deutsches Gesetz über die Weiterverwendung von Informationen öffentlicher Stellen.

KI (Künstliche Intelligenz) bzw. **AI** (Artificial Intelligence).

KV-Cache (Key-Value Cache): Zwischenspeicherung von Key- und Value-Vektoren aus vorherigen Tokens während der Inferenz. Reduziert die Komplexität von $O(n^2)$ auf $O(n)$ pro Token-Schritt und ist zentral für effizientes Serving großer Modelle. (Tay et al., 2022)

LLM (Large Language Model): Großes Sprachmodell, das auf riesigen Textmengen trainiert wurde und Sprache verstehen sowie generieren kann. Beispiele sind GPT-Modelle oder LLaMA.

LLMops (Large Language Model Operations): speziellen Praktiken und Workflows, die die Entwicklung, Bereitstellung und Verwaltung von KI-Modellen während ihres gesamten Lebenszyklus beschleunigen. (zitiert von ibm.com/de-de/think/topics/llmops, letzter Lookup 12.6.2026)

LoRA (Low-Rank Adaptation): PEFT-Methode (siehe Finetuning), bei der Gewichtsmatrizen durch Low-Rank-Zerlegungen approximiert werden. Dadurch lassen sich große Modelle effizient anpassen, ohne alle Parameter zu trainieren. (Hu et al., 2022)

Mixture of Experts (MoE): Architektur, bei der nur ein Teil des Modells (Experten) pro Input aktiviert wird. Ermöglicht hohe Modellkapazität bei vergleichsweiser geringer Rechenlast pro Inferenz. (Fedus et al., 2022)

ML (Machine Learning / Maschinelles Lernen)

Multimodal Model: Modell, das mehrere Datentypen gleichzeitig verarbeiten kann, z.B. Text, Bilder oder Audio. Ermöglicht komplexere Anwendungen wie Bildbeschreibung oder visuelle QA.

Neural Network / neuronales Netzwerk: Ein Netzwerk aus künstlichen Neuronen, das Muster in Daten lernt. Grundlage nahezu aller modernen KI-Systeme.

NIS 2-RL (NIS-2-Richtlinie): Richtlinie des Europäischen Parlaments und des Rates für ein hohes gemeinsames Cybersicherheitsniveau in der Europäischen Union.

NLP (Natural Language Processing): Teilgebiet der KI, das sich mit der Verarbeitung und Analyse natürlicher Sprache beschäftigt. LLMs sind eine zentrale Technologie in diesem Bereich.

NVMe SSD: SSD mit Non-Volatile Memory Express Protokoll; extrem schneller Datenspeicher.

Open-Weight-Modell: ist ein Modell bei dem die Gewichte frei verfügbar sind (z.B. unter der Apache 2.0 Lizenz), der Trainingscode und die Trainingsdaten aber nicht.

Open-Source-Modell: ist ein Modell bei dem der Trainingscode öffentlich ist,eingesehen, geändert und genutzt werden kann, jedoch unter Einhaltung der in der verwendeten Lizenz festgeschriebenen Bedingungen. Bei Open-Source-Modellen sind auch die Trainingsdaten offengelegt. Womit kontrolliert werden kann, ob die verwendeten Trainingsdaten DSGVO-konform sind und das Modell entsprechend verwendet werden kann.

OPEX (operational expenditure): laufende Betriebsausgaben.

Parameter: Im LLM-Kontext: Trainierbare Gewichte eines Modells, die während des Trainings angepasst werden. Große LLMs besitzen Milliarden bis Billionen Parameter.

PEFT (Parameter-Efficient Fine-Tuning / parametereffiziente Feinabstimmung): Methode, um gigantische KI-Modelle mit extrem wenig Rechenaufwand an neue, spezifische Aufgaben anzupassen. Nur ein sehr kleiner Anteil der Parameter eines Modells wird neu trainiert. Die mit Abstand populärste Technik innerhalb der PEFT-Methoden ist LoRA (Low-Rank Adaptation).

Perplexity: Maß für die Unsicherheit eines Sprachmodells bei der Vorhersage eines Tokens. Formal ist Perplexity die exponentielle Form der Kreuzentropie; niedrigere Werte bedeuten bessere Modellqualität.

Personally Identifiable Information (PII): personenbezogene Daten, die eine Person direkt oder indirekt identifizieren können, dazu gehören Namen, Adressen, E-Mail-Adressen, Sozialversicherungsnummern biometrische Daten udgl.

Prompt: Eingabetext, der einem Modell gegeben wird, um eine Antwort zu erzeugen. Die Gestaltung (Prompt Engineering) beeinflusst stark die Qualität der Ausgabe.

Prompt Engineering: Systematische Gestaltung von Eingaben zur Steuerung von Modellverhalten. Umfasst Techniken wie Few-Shot, Chain of Thought (CoT) oder strukturierte Prompts und ist entscheidend für Performance ohne Modelländerung.

Prompt Injection: Sicherheitsangriff, bei dem manipulierte Eingaben das Verhalten eines Modells verändern oder Sicherheitsrichtlinien umgehen. Besonders kritisch bei agentischen Systemen mit Tool-Zugriff. (Perez & Ribeiro, 2022)

Quantisierung: Methode, die in LLMs verwendet wird, um Gewichtungen und Aktivierungswerte von Daten mit hoher Genauigkeit in Daten mit niedrigerer Genauigkeit, wie 8-Bit-Ganzzahl (INT8), umzurechnen (cf. ibm.com/de-de/think/topics/quantization, letzter Lookup 12.6.2026).

RAG (Retrieval-Augmented Generation): Architektur, die ein Sprachmodell mit externem Wissen kombiniert, indem relevante Dokumente vor der Generierung abgerufen werden. Reduziert Halluzinationen und verbessert Aktualität sowie Nachvollziehbarkeit. (Lewis et al., 2020)

RAM (Random Access Memory): der Arbeitsspeicher eines Computers.

Read Team: Als Read Team oder als Rotes Team wird eine unabhängige Gruppe bezeichnet, welche bei einer Organisation oder einem Unternehmen eine Verbesserung der Effektivität im Sicherheitsmanagement bewirken soll, indem sie als Gegner auftritt. Ziel ist es dabei immer, Sicherheitslücken aufzuspüren, bevor ein externer Dritter diese ausnutzen kann. (de.wikipedia.org/wiki/Red_Team, letzter Lookup 12.6.2026)

REF (Renewable Energy Factor)

Reinforcement Learning (RL): Lernparadigma, bei dem ein Agent durch Belohnungen und Strafen optimiert wird. Wird u. a. für Entscheidungsprozesse eingesetzt und im LLM-Kontext primär zur Feinabstimmung verwendet.

RLHF (Reinforcement Learning from Human Feedback): Kombination aus supervised learning und RL, bei der menschliches Feedback genutzt wird, um ein Reward-Modell zu trainieren und das Verhalten eines LLMs zu steuern. Damit wird ein LLM so trainiert, dass es erwünschte Antworten gibt. (Ouyang et al., 2022)

Role-Based Access Control (RBAC, DE: rollenbasierte Zugriffskontrolle): Methode zur Verwaltung von Zugriffsrechten, wobei die Berechtigungen definierten Rollen, wie z. B. Administrator, HR, User) zugewiesen werden.

Scaling Laws: Empirische Gesetze, die zeigen, dass Modellleistung systematisch mit Modellgröße, Datenmenge und Rechenleistung skaliert. Grundlage für die Entwicklung großer Modelle. (Kaplan et al., 2020)

SCHUFA (Schutzgemeinschaft für allgemeine Kreditsicherung)

Self-Attention: Spezielle Form der Attention, bei der ein Input sich selbst referenziert. Ermöglicht globale Kontextmodellierung unabhängig von Sequenzlänge. (Vaswani et al., 2017)

Screening-LCA (vereinfachte Lebenszyklusanalyse / Screening-Ökobilanz): schnelle, kostengünstige und datenreduzierte Form der klassischen Lebenszyklusanalyse (LCA).

SLA (Service Level Agreement): Vertrag zwischen einem Dienstleister und einem Kunden, der den zu erbringenden Service und das zu erwartende Leistungsniveau festlegt. Ein SLA beschreibt auch, wie die Leistung gemessen und genehmigt wird und was passiert, wenn die Leistungsniveaus nicht erreicht werden. (zitiert von ibm.com/de-de/think/topics/service-level-agreement; letzter Lookup 12.6.2026)

SSD (Solid State Drive oder Solid State Disk): sehr schneller elektronischer Datenspeicher.

Token: Kleinste Verarbeitungseinheit eines Sprachmodells (z. B. Wort oder Wortteil). Tokens bestimmen u. a. Kontextgröße und Kosten.

Tokenisierung (Tokenization): Prozess der Zerlegung von Text in Tokens (Subwords, Wörter oder Zeichen). Moderne Verfahren wie BPE oder SentencePiece optimieren Kompromisse zwischen Vokabulargröße und Generalisierung.

Transformer: Architektur basierend auf Self-Attention, die parallele Verarbeitung von Sequenzen ermöglicht. Besteht aus Encoder- und/oder Decoder-Blöcken und bildet die Grundlage aller modernen LLMs. (Vaswani et al., 2017)

Unberührtes Testset (Holdout-Datensatz): Das sind Daten, die das KI -Modell weder beim Training noch bei der Feinabstimmung gesehen hat. Diese Daten dienen zur Prüfung der tatsächlichen Leistung des Modells.

UrhG (Urheberrechtsgesetz)

Vector Database (Vektordatenbank): Spezialisierte Datenbank für effiziente Ähnlichkeitssuche in hochdimensionalen Embedding-Räumen (z. B. mittels Approximate Nearest Neighbor (ANN)-Algorithmen wie Hierarchical Navigable Small Worlds (HNSW). Zentral für skalierbare RAG-Systeme.

VRAM (Video Random Access Memory): lokaler Speicher auf der Graphikkarte.

Zero-Shot / Few-Shot Learning: Fähigkeit eines Modells, Aufgaben ohne (Zero-Shot) oder mit wenigen (Few-Shot) Beispielen zu lösen. Entsteht durch starke Generalisierungseigenschaften großer Modelle. (Brown et al., 2020)

Tabellenverzeichnis

Abbildungsverzeichnis

Literaturverzeichnis

Brown, Tom et al.: Language models are few-shot learners. Advances in neural information processing systems, 2020, 33. Jg., S. 1877-1901.

Copei, Sebastian/Hohlfeld, Oliver/Kosiol, Jens/Ristoski, Aleksandar: LLMs Choose the Right Stack: From Patterns to Tools, IEEE Computer Society, 2025, pp. 276–283.

Delort, Etienne/Riou, Laura/Srivastava, Anukriti: Environmental Impact of Artificial Intelligence. INRIA / CEA Leti bibliographic report, 2023. HAL Id: hal-04283245.

Desroches, Clément et al.: Exploring the sustainable scaling of AI dilemma: A projective study of corporations' AI environmental impacts. Preprint, 2025, arXiv:2501.14334.

Dettmers, Tim, et al.: "Qlora: Efficient finetuning of quantized llms." Advances in neural information processing systems 36 (2023): 10088-10115.

EU: European Commission. Commission Delegated Regulation (EU) 2024/1364 of 14 March 2024 on the first phase of the establishment of a common Union rating scheme for data centres. Official Journal of the European Union, 2024/1364, 17.05.2024a.

EU: European Parliament and Council of the European Union. Regulation (EU) 2024/1252 of 11 April 2024 establishing a framework for ensuring a secure and sustainable supply of critical raw materials. Official Journal of the European Union, 2024/1252, 03.05.2024b.

EU: European Commission. Critical raw materials - 2023 list and Critical Raw Materials Act overview. Internal Market, Industry, Entrepreneurship and SMEs portal, accessed April 2026a.

EU: European Commission, Joint Research Centre. Raw Materials Information System (RMIS): advanced materials, electronics and CRM dependencies. Accessed April 2026b.

Faiz, Ahmad et al.: LLMCarbon: Modeling the End-to-End Carbon Footprint of Large Language Models. ICLR 2024 (OpenReview id: aIok3ZD9to).

Fedus, William/Barret, Zoph/Shazeer, Noam: "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity." Journal of Machine Learning Research 23.120 (2022): 1-39.

Greshake, Kai et al.: Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection, arXiv (2023). — arXiv:2302.12173.

Fernandez, Jared et al.: Energy Considerations of Large Language Model Inference and Efficiency Optimizations. Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025), 32556-32569.

Hauzenberger, Lukas et al.: Effective Distillation to Hybrid xLSTM Architectures. arXiv preprint arXiv:2603.15590, 2026.

Hinton, Geoffrey/Vinyals, Oriol/ Dean, Jeff: "Distilling the knowledge in a neural network." arXiv preprint arXiv:1503.02531 (2015).

Hitzel, Uli: The LLM Stack – A Practical Guide to Understanding AI, 2025.

Hoffmann, Jordan, et al.: "Training compute-optimal large language models." 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Hou, Xinyi/Zhao, Yanjie/Wang, Haoyu: LLM Applications: Current Paradigms and the Next Frontier, arXiv, 2025.

Hu, Edward J., et al.: "Lora: Low-rank adaptation of large language models." Iclr 1.2 (2022): 3.

IONOS White Paper: Aktuelle Rechtslage zum US-Datentransfer unter besonderer Berücksichtigung des US CLOUD Act. 2021. business-services.heise.de/it-management/cloud-computing/beitrag/dsgvo-und-cloud-act-rechtliche-risiken-beim-datentransfer-in-die-usa-4220

ISO: Environmental management – Life cycle assessment – Principles and framework (ISO 14040:2006). Geneva: International Organization for Standardization, 2006a.

ISO: Environmental management – Life cycle assessment – Requirements and guidelines (ISO 14044:2006). Geneva: International Organization for Standardization, 2006b.

Jiang, Peng et al.: Preventing the Immense Increase in the Life-Cycle Energy and Carbon Footprints of LLM-Powered Intelligent Chatbots. Engineering 40 (2024): 202-210. doi.org/10.1016/j.eng.2024.04.002.

Kaplan, Jared et al.: „Scaling Laws for Neural Language Models“, 23. Jänner 2020, arXiv: arXiv:2001.08361. doi: 10.48550/arXiv.2001.08361.

Kwon, Woosuk, et al.: "Efficient memory management for large language model serving with pagedattention." Proceedings of the 29th symposium on operating systems principles. 2023.

Lewis, Patrick, et al.: "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in neural information processing systems 33 (2020): 9459-9474. (neueste Version auf arxiv von 2021 arxiv.org/pdf/2005.11401)

Li, Pengfei et al.: Making AI Less 'Thirsty': Uncovering and Addressing the Secret Water Footprint of AI Models. arXiv:2304.03271, 2023.

Ligozat, Anne-Laure et al.: Unraveling the Hidden Environmental Impacts of AI Solutions for Environment: Life Cycle Assessment of AI Solutions. Sustainability 14(9) (2022): 5172. doi.org/10.3390/su14095172.

Lin, Stephanie/Hilton, Jacob/Evans, Owain: "Truthfulqa: Measuring how models mimic human falsehoods." Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers). 2022.

Luccioni, Alexandra Sasha/Viguier, Sylvain/Ligozat, Anne-Laure: Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. Journal of Machine Learning Research 24(253) (2023): 1-15.

Mikolov, Tomas, et al.: "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

Nessler, Bernhard/ Doms, Thomas/Hochreiter, Sepp: "Functional trustworthiness of AI systems by statistically valid testing." arXiv preprint arXiv:2310.02727 (2023).

NIST: National Institute of Standards and Technology (US): Artificial intelligence risk management framework: generative artificial intelligence profile (Nr. NIST AI 600-1). Gaithersburg, MD: National Institute of Standards and Technology (U.S.), 2024.

Ouyang, Long, et al.: "Training language models to follow instructions with human feedback." Advances in neural information processing systems 35 (2022): 27730-27744.

OWASP Foundation: OWASP Top 10 for LLM Applications 2025 (Nr. Version 2025): OWASP Foundation, 2024

OWASP Foundation: OWASP GenAI Data-Security Risks and Mitigations 2026 (Nr. v1.0): OWASP Foundation, 2026

Patterson, David et al.: Carbon Emissions and Large Neural Network Training. arXiv:2104.10350, 2021.

Perez, Fábio/Ribeiro, Ian: "Ignore previous prompt: Attack techniques for language models." arXiv preprint arXiv:2211.09527 (2022).

Radovanovic, Matt/Bornstein, Rajko: Emerging Architectures for LLM Applications, 2023.

Ren, Shaolei et al.: Reconciling the contrasting narratives on the environmental impact of large language models. Scientific Reports 14 (2024): 26310. doi.org/10.1038/s41598-024-76682-6.

Tay, Yi, et al.: "Efficient transformers: A survey." ACM Computing Surveys 55.6 (2022): 1-28.

Stone, Simon/Crossett, Jonathan/Luker, Tivon/Leligdon, Lora/Cowen, William/Darabos, Christian: Dartmouth Chat - Deploying an Open-Source LLM Stack at Scale, in Practice and Experience in Advanced Research Computing 2025: The Power of Collaboration, Association for Computing Machinery, New York, NY, USA, pp. 1–5.

Vaswani, Ashish, et al.: "Attention is all you need." Advances in neural information processing systems 30 (2017).

Wang, Xiaorong et al.: Energy and Carbon Considerations of Fine-Tuning BERT. Findings of the Association for Computational Linguistics: EMNLP 2023, 9058-9069. doi.org/10.18653/v1/2023.findings-emnlp.607.

Wei, Jason, et al.: "Chain-of-thought prompting elicits reasoning in large language models." Advances in neural information processing systems 35 (2022): 24824-24837.

Witten, Ian H.: Managing gigabytes: compressing and indexing documents and images. New York: Van Nostrand Reinhold, 1994. Zugegriffen: 31. März 2026. [Online]. Verfügbar unter: archive.org/details/managinggigabyte0000witt

Xue, Jiaqi et al.: BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models, arXiv (2024). — arXiv:2406.00083