GANZ EINFACH

KI-SOUVERÄNITÄT

Das Handbuch für lokale LLMs: Schritt für Schritt zu RAG, autonomen Agenten und privater KI-Infrastruktur auf eigener Hardware



Danilo Sieren

Danilo Sieren

Ganz Einfach

KI Souveränität

Sprachmodelle und Bildmodelle lokal verstehen und sicher nutzen

Überarbeitete und erweiterte
Neuauflage 2026

Vorwort

Eine lokale KI kann Texte entwerfen, Dokumente durchsuchen und Bilder erzeugen, während du entscheidest, auf welchem Rechner die Verarbeitung stattfindet. Dieser Gestaltungsspielraum ist der Kern von KI-Souveränität. Er entsteht durch nachvollziehbare Entscheidungen über Modelle, Daten, Berechtigungen und Betrieb. Ein heruntergeladenes Modell allein genügt dafür noch nicht.

Dieses Buch begleitet dich von der ersten lokalen Antwort bis zu einem wartbaren System mit Dokumentensuche und begrenzten Werkzeugen. Du lernst, wie Sprachmodelle arbeiten, welche Hardware sie benötigen und wie du ihre Ergebnisse prüfst. Ein eigener Schwerpunkt behandelt lokale Bildgenerierung. Hinzu kommen Sprachverarbeitung, mehrere Benutzer, Fernzugriff, Sicherungen und ein Wiederanlauf ohne Internet.

Die Beispiele beginnen mit unverfänglichen Übungsdaten. Später baust du daraus ein kleines Wissensarchiv und eine Anwendung, die strukturierte Entwürfe erstellt. Du kannst den grafischen Einstieg wählen oder die Schnittstellen genauer untersuchen. Beide Wege führen zum selben Ziel: Du sollst verstehen, was dein System tut, und es auch dann beherrschen, wenn eine Antwort falsch ist oder ein Dienst ausfällt.

Lokaler Betrieb hat Grenzen. Er macht ein Modell weder allwissend noch unfehlbar und ersetzt keine fachliche Prüfung. Auch Datenschutz, Lizenzen und die Sicherheit anderer Menschen bleiben praktische Aufgaben. Dieses Buch zeigt deshalb neben den Arbeitsschritten jeweils die Kontrollen, an denen du einen brauchbaren Aufbau erkennst.

So arbeitest du mit diesem Buch

Beginne mit den Kapiteln 1 bis 5. Wähle anschließend einen Einstieg: LM Studio für das Arbeiten in einer grafischen Anwendung oder Ollama für einen lokalen Dienst mit Kommandozeile. Du musst nicht beide Programme installieren. Die späteren Kapitel erklären, wann eine zusätzliche Oberfläche, eine Dokumentensuche oder ein Server tatsächlich einen Nutzen bringt.

Die Beispiele mit Terminalbefehlen nennen ihre Voraussetzungen. Ein Befehl für eine Unix-Shell gehört in das Terminal von macOS oder Linux; ein PowerShell-Beispiel gehört in PowerShell. Führe Befehle einzeln aus und lies die Ausgabe. Bei einem Fehler gehst du zur beschriebenen Kontrolle zurück, statt mit den nächsten Schritten fortzufahren. Anhang A enthält eine kleine Übungsbibliothek; Anhang B führt durch die Abnahme eines privaten Arbeitsplatzes.

Modellnamen bezeichnen konkrete Beispiele. Eine neuere Modellfamilie muss für deine Aufgabe nicht geeigneter sein. Halte Anwendungsversion, Modellkennung, Quantisierung und Einstellungen fest. Benutze zunächst Kopien öffentlicher oder erfundener Dokumente. Vertrauliche Originale kommen erst hinzu, wenn Datenwege und Wiederherstellung geprüft sind.

Die Quellen und die technischen Angaben haben den Stand 2. November 2026. Bedienoberflächen, verfügbare Pakete und Modellkataloge verändern sich. Die angegebenen offiziellen Dokumentationen helfen dir, eine abweichende Bezeichnung einzuordnen. Entscheidend bleiben die erläuterte Funktion und der kontrollierte Datenweg.

Inhalt

Vorwort 2
So arbeitest du mit diesem Buch 3
1 Was KI Souveränität praktisch bedeutet 7
2 Sprachmodelle verständlich einordnen 11
3 Datenwege und Bedrohungen planen 16
4 Hardware ohne Fehlkauf auswählen 20
5 Modelle und Lizenzen sorgfältig auswählen 25
6 Mit LM Studio grafisch beginnen 30
7 Ollama als lokalen Dienst einrichten 34
8 Den ersten lokalen Arbeitsplatz abnehmen 39
9 Gute Aufträge formulieren 42
10 Die lokale API nutzen 47
11 Eine gemeinsame Oberfläche mit Containern betreiben 53
12 Dokumente für die Suche vorbereiten 58
13 RAG und semantische Suche verstehen 63
14 Ein lokales Archiv mit AnythingLLM aufbauen 69
15 Antworten und Quellen systematisch prüfen 73
16 Zahlen und Tabellen verlässlich verarbeiten 79
17 Lokale Bildgenerierung verstehen 83
18 Mit ComfyUI das erste Bild erzeugen 87
19 Bildbearbeitung und Workflows vertiefen 92
20 Sprache und lokale Assistenz einbinden 96
21 Prompt Injection und Erweiterungen beherrschen 101
22 Werkzeuge und Agenten kontrolliert einsetzen 105
23 MCP und Automatisierung einordnen 110
24 Ein geprüftes Entwurfsprojekt entwickeln 115
25 Fernzugriff und mehrere Benutzer absichern 121
26 Sicherungen und Wiederherstellung planen 126

27 Updates und Modellwechsel kontrollieren 131
28 Datenschutz und Recht im aktuellen Rahmen 136
29 Leistung und Qualität nachvollziehbar messen 140
30 Fehler nach Schichten eingrenzen 144
31 Kosten und Energie sachlich vergleichen 148
32 Offlinebetrieb und langfristige Unabhängigkeit sichern 152
A Eine vollständige Übungsbibliothek 157
B Einen privaten Arbeitsplatz vollständig abnehmen 165
C Laufzeiten und Anpassungen vertiefen 171
D Fallwerkstätten für schwierige Entscheidungen 179
E Begriffe verständlich nachschlagen 187
F Vorlagen für den eigenen Betrieb 193
G Ein Wissensarchiv vom Entwurf bis zum Betrieb 198
Ausführlicher Inhalt 205
Stichwortverzeichnis 214
Quellen und weiterführende Dokumentation 220
Impressum 224

1 Was KI Souveränität praktisch bedeutet

1.1 Eine Aufgabe und ihre Grenzen bestimmen

Eine KI-Anwendung ist dann nützlich, wenn sie eine konkrete Aufgabe unter vertretbaren Bedingungen erledigt. Für den Einstieg genügt eine Frage wie diese: Kann mein Rechner aus drei freigegebenen Handbüchern die passende Wartungsanweisung finden und die Fundstelle nennen? Daraus lassen sich Anforderungen ableiten. Du brauchst lesbare Dokumente, eine Suche, ein geeignetes Sprachmodell und eine Möglichkeit, das Ergebnis mit dem Original zu vergleichen.

Die abstrakte Absicht, eine eigene KI für alles zu betreiben, hilft bei der Auswahl kaum. Sie führt leicht zu großen Downloads, unnötigen Erweiterungen und teurer Hardware. Schreibe deshalb vor der Installation drei Dinge auf: Welche Eingaben sind vorgesehen? Welches Ergebnis soll entstehen? Welche Handlung bleibt ausdrücklich bei dir? Bei einer Dokumentensuche bleibt beispielsweise die Entscheidung über eine Reparatur beim Menschen.

Ergänze einen Rückweg. Wenn die KI nicht funktioniert, müssen die Handbücher weiterhin mit einer gewöhnlichen Suche erreichbar sein. Eine zusätzliche Anwendung sollte deinen Zugang zum eigenen Wissen verbessern und keinen neuen Zwang schaffen. Dass du sie vorübergehend abschalten kannst, ist ein gutes Merkmal eines beherrschbaren Systems.

1.2 Kontrolle hat mehrere Ebenen

Du kannst Modellgewichte besitzen und trotzdem von einem fremden Dienst abhängig sein. Eine lokale Oberfläche kann ihre Embeddings bei einem Cloudanbieter erzeugen. Ein Bildprogramm kann lokale Modelle und kostenpflichtige Onlinedienste im selben Arbeitsbereich anbieten. Ein selbst betriebener Server kann für Anmeldung, Updates oder Lizenzprüfung weiterhin eine Verbindung benötigen.

Unterscheide deshalb die Ausführung des Modells von der gesamten Verarbeitungskette. Zur Kette gehören Dateieinlesung, Texterkennung, Suche, Sprachverarbeitung, Speicherung und Export. Auch Sicherungen und Fehlermeldungen können vertrauliche Inhalte enthalten. Kontrolliere jede beteiligte Komponente und halte fest, welche Verbindungen notwendig sind.

Ein lokaler Aufbau bietet dir eigene Entscheidungen über Versionen und Speicherorte. Im Gegenzug übernimmst du Wartung, Schutz und Wiederherstellung. Diese Aufgaben verschwinden bei einem Cloudangebot nicht; dort verteilen sie sich anders zwischen dir und dem Anbieter. Beurteile beide Betriebsformen anhand deiner tatsächlichen Anforderungen, nicht anhand eines pauschalen Vertrauensurteils.

1.3 Lokal und Cloud sinnvoll vergleichen

Für einen privaten Entwurf ohne personenbezogene Daten kann ein externer Dienst eine angemessene Lösung sein. Ein vertrauliches Archiv mit strengen Vorgaben kann einen lokalen Aufbau erfordern. Manchmal ist eine Kombination zweckmäßig: unverfängliche Recherche extern, interne Unterlagen ausschließlich lokal. Die Grenze muss vor der Eingabe klar sein.

Behauptungen, jeder Cloudanbieter verkaufe sämtliche Prompts oder trainiere immer mit allen Eingaben, sind unzutreffende Verallgemeinerungen. Bedingungen unterscheiden sich nach Produkt, Vertrag und Einstellung. Umgekehrt ist eine beliebige lokale Installation kein Beweis für vollständigen Datenschutz. Entscheidend sind überprüfbare Datenwege, Berechtigungen und Vereinbarungen.

Vergleiche zudem die Qualität. Ein kleineres Modell kann eine eng begrenzte Aufgabe zuverlässig genug erledigen und bei offenen Fachfragen versagen. Ein großes externes Modell kann bessere Antworten liefern und trotzdem ungeeignet sein, wenn die Verarbeitung nicht zulässig ist. Du suchst eine tragfähige Gesamtlösung für deinen Anwendungsfall.

1.4 Kosten und Unabhängigkeit realistisch bewerten

Ein lokales Modell kann ohne laufende Modellgebühr arbeiten. Es benötigt dennoch Speicherplatz, Energie und deine Zeit. Eine neue Grafikkarte verursacht Kosten, bevor die erste sinnvolle Antwort entsteht. Rechne auch Sicherungsmedien und einen möglichen Hardwareausfall ein. Kapitel 31 zeigt eine nachvollziehbare Beispielrechnung.

Unabhängigkeit bedeutet außerdem, eine andere Laufzeit oder ein anderes Modell einsetzen zu können. Bewahre Texte und Dokumente in üblichen Formaten auf. Exportiere wichtige Chatresultate mit ihrem Kontext. Eine Datenbank, die nur mit einer nicht mehr verfügbaren Anwendung lesbar ist, kann zur neuen Abhängigkeit werden.

Eine konservative Entscheidung ist oft die beste erste Investition: vorhandenen Rechner prüfen, ein kleines Modell herunterladen, eine Aufgabe testen und erst danach einen Ausbau planen. Wenn das Ergebnis nicht überzeugt, hast du etwas über die Aufgabe gelernt, ohne ein Rechenzentrum im Arbeitszimmer errichten zu müssen.

1.5 Dein erstes Betriebsprofil schreiben

Formuliere dein Vorhaben in einem kurzen Absatz. Ein Beispiel lautet: „Ich verwende eine lokale Anwendung allein auf meinem Notebook. Sie verarbeitet Kopien öffentlicher Gerätehandbücher. Sie darf weder Nachrichten versenden noch Dateien verändern. Ergebnisse prüfe ich an der angegebenen Originalseite.“ Das Profil enthält Zweck, Benutzerkreis, Daten und Befugnisse.

Notiere zusätzlich eine überprüfbare Erfolgsbedingung. Beispielsweise sollen acht von zehn vorbereiteten Fragen mit einer richtigen Fundstelle beantwortet werden, während bei fehlenden Angaben eine erkennbare Enthaltung erfolgt. Eine bestimmte Trefferquote ist dabei dein Abnahmekriterium, keine allgemeine Zusage über lokale KI.

Bewahre dieses Profil zusammen mit den technischen Notizen auf. Wenn später ein Messenger oder ein Agent hinzukommt, ändert sich der Datenweg. Dann prüfst du das Profil erneut. So wächst dein System durch begründete Entscheidungen und nicht durch eine Sammlung eingeschalteter Funktionen.

2 Sprachmodelle verständlich einordnen

2.1 Training und Anwendung unterscheiden

Beim Training werden Parameter eines Modells anhand von Daten angepasst. Bei der späteren Anwendung, der Inferenz, werden diese Parameter für Berechnungen verwendet. Ein Chat mit einem gewöhnlichen lokalen Modell trainiert dessen Gewichte nicht automatisch weiter. Dass eine Oberfläche den Gesprächsverlauf speichert, ist ein anderer Vorgang.

Ein Sprachmodell erzeugt Text in kleinen Einheiten, den Tokens. Es nutzt den verfügbaren Kontext, um weitere Tokens auszuwählen. Daraus können verständliche Erklärungen, gute Entwürfe und auch überzeugend formulierte Fehler entstehen. Die sprachliche Sicherheit einer Antwort ist kein verlässlicher Maßstab für ihre Richtigkeit.

Das Modell enthält keine geordnete Sammlung vollständig abrufbarer Originalseiten. Eine Frage nach einer Quelle kann deshalb eine erfundene Literaturangabe hervorbringen. Wenn du Belege brauchst, musst du sie aus einer überprüfbaren Informationsquelle bereitstellen oder unabhängig beschaffen. Die Dokumentensuche in den späteren Kapiteln löst einen Teil dieser Aufgabe.

2.2 Tokens und Kontext verstehen

Ein Token kann ein Wortteil, ein Satzzeichen oder eine andere Zeichenfolge sein. Wie viele Tokens ein deutscher Absatz benötigt, hängt vom Tokenizer des Modells ab. Eine feste Gleichsetzung von einem Token mit einem Wort führt zu falschen Schätzungen. Benutze nach Möglichkeit die Anzeige deiner Laufzeit.

Das Kontextfenster begrenzt, wie viel Material für eine einzelne Berechnung verfügbar ist. Darin konkurrieren Anweisungen, Verlauf, Dokumentauszüge, Werkzeugantworten und die entstehende Ausgabe um Platz. Eine beworbene maximale Kontextlänge ist keine Zusage, dass dein Rechner sie bei guter Geschwindigkeit vollständig nutzen kann.

Auch in einem großen Kontext kann eine wichtige Stelle übersehen werden. Relevante Informationen können zwischen ähnlichen Absätzen untergehen oder durch eine fehlerhafte Zusammenfassung verloren gehen. Prüfe lange Aufgaben deshalb mit gezielten Fragen zu Randstellen und widersprüchlichen Angaben, statt nur die maximal einstellbare Zahl zu betrachten.

2.3 Parameter und Modellarchitektur lesen

Die Bezeichnung 4B steht gewöhnlich für ungefähr vier Milliarden Parameter. Sie ist eine Größenangabe und keine Qualitätsnote. Trainingsdaten, Architektur, Nachtraining und die konkrete Aufgabe beeinflussen das Ergebnis ebenfalls. Zwei Modelle gleicher Größenklasse können sich bei deutscher Sprache oder Tabellenanalyse deutlich unterscheiden.

Bei einem Mixture-of-Experts-Modell wird nur ein Teil der Experten pro Berechnung aktiviert. Eine Kennung wie „A3B“ kann auf eine aktive Teilmenge hinweisen. Daraus folgt nicht, dass nur diese Teilmenge als Gewichte gespeichert werden muss. Plane Speicher anhand der tatsächlichen Modelldateien und der dokumentierten Laufzeit ein.

Manche neueren Modelle verbinden verschiedene Mechanismen für Kontextverarbeitung. Eine vereinfachte Speicherformel für einen klassischen Transformer kann deshalb nur eine Orientierung liefern. Sie ersetzt keine Messung. Für eine Kaufentscheidung zählt der nachgewiesene Betrieb genau der gewünschten Variante.

2.4 Quantisierung ohne falsche Versprechen

Quantisierung stellt Modellwerte mit weniger Bits dar. Das spart typischerweise Speicher und kann den Betrieb auf kleinerer Hardware ermöglichen. Dabei ändern sich numerische Eigenschaften; die Qualität kann je nach Modell, Aufgabe und Verfahren unterschiedlich leiden. Eine gute Vier-Bit-Variante ist nicht grundsätzlich jeder Acht-Bit-Variante unterlegen oder gleichwertig.

Bezeichnungen wie Q4_K_M sind konkrete Format- und Verfahrenskennungen. Sie bedeuten nicht, dass die gesamte Anwendung exakt vier Bit je Parameter benötigt. Zusätzliche Datenstrukturen, unterschiedlich quantisierte Teile und Laufzeitpuffer kommen hinzu. Die Dateigröße ist daher die bessere erste Information als eine stark vereinfachte Rechnung.

Vergleiche Quantisierungen mit denselben Eingaben. Achte besonders auf Zahlen, seltene Begriffe und das Befolgen enger Ausgabevorgaben. Eine Variante, die beim freien Schreiben angenehm wirkt, kann bei strukturierter Extraktion häufiger scheitern. Halte diese Unterschiede in deiner Modellübersicht fest.

2.5 Gedächtnis und Persönlichkeit einordnen

Wenn ein Chatbot deinen Namen verwendet, kann dieser im aktuellen Verlauf, in einer gespeicherten Notiz oder in einem Anwendungsprofil stehen. Daraus folgt keine persönliche Beziehung und kein menschliches Erinnern. Untersuche die Speicherfunktion deiner Oberfläche, bevor du intime Angaben eingibst.

Ein Systemprompt ist eine vom Programm bereitgestellte Anweisung. Er kann Tonfall und Aufgabenrahmen beeinflussen, aber keine physische Sicherheitsgrenze ersetzen. Eine Anweisung „Lies niemals private Dateien“ schützt keine Datei, wenn das Programm dem Modell tatsächlich freien Dateizugriff gibt.

Behandle das Modell als eine leistungsfähige, fehlbare Verarbeitungskomponente. Diese Einordnung ist praktisch: Du kannst seine nützlichen Leistungen verwenden, ohne Aussagen über Absichten, Loyalität oder Bewusstsein als technische Eigenschaften vorauszusetzen. Die Kontrolle bleibt bei den Menschen und Programmen, die Eingaben auswählen und Handlungen ausführen.

2.6 Sampling und Kontextspeicher genauer betrachten

Bei autoregressiver Textausgabe entsteht für den nächsten Schritt eine Verteilung über mögliche Tokens. Das Auswahlverfahren entscheidet, welcher Kandidat verwendet wird. Temperatur verändert dabei die Verteilung; Top-k beschränkt die Auswahl auf eine Anzahl von Kandidaten, Top-p auf eine kumulierte Wahrscheinlichkeitsmasse. Nicht jede Laufzeit und jedes Modell verwendet dieselben sinnvollen Kombinationen.

Wenn du mehrere Auswahlparameter gleichzeitig stark veränderst, kann die Ausgabe unverständlich oder auffällig wiederholend werden. Für einen Vergleich notierst du deshalb alle Werte und verwendest zunächst die empfohlenen Einstellungen der konkreten Modellkarte. Ein Temperaturwert lässt sich nicht als allgemeiner Regler von „falsch“ nach „wahr“ verstehen. Er steuert Auswahlverhalten innerhalb dessen, was das Modell berechnet.

Viele Transformer speichern während der Verarbeitung Schlüssel- und Wertdarstellungen früherer Tokens im KV-Cache. Dadurch müssen bestimmte Zwischenberechnungen nicht bei jedem Ausgabeschritt vollständig neu erfolgen. Der Cache benötigt zusätzlichen Speicher. Wie stark er mit Kontext und Parallelität wächst, hängt von Architektur, Datentyp und Laufzeit ab. Neuere hybride Architekturen können andere Zusammenhänge aufweisen.

Für eine anschauliche Probe lädst du dasselbe Modell zunächst mit kleinem Kontext und stellst eine kurze Frage. Notiere die belegten Ressourcen. Wiederhole den Lauf mit einem längeren, aber tatsächlich benötigten Dokument und passender größerer Kontextzuteilung. Beobachte sowohl das Laden als auch die Verarbeitung. Die Differenz ist eine Beobachtung deines Systems, keine allgemeine Speicherformel.

Einige Laufzeiten reservieren Speicher bereits beim Laden, andere verändern die Belegung während des Betriebs. Vergleiche deshalb denselben Zeitpunkt im Ablauf. Wenn du nur nach dem Entladen misst, siehst du den entscheidenden Spitzenbedarf nicht. Diese Unterscheidung erklärt, warum dieselbe Gewichtsdatei auf zwei scheinbar gleich ausgestatteten Rechnern unterschiedlich gut laufen kann.

3 Datenwege und Bedrohungen planen

3.1 Schutzbedarf anhand konkreter Daten festlegen

Eine Bedienungsanleitung, eine Personalakte und ein privates Tagebuch verlangen unterschiedliche Entscheidungen. Beginne mit wenigen nachvollziehbaren Kategorien: öffentlich, intern und besonders schutzbedürftig. Diese Einteilung ist ein Arbeitshilfsmittel und ersetzt keine gesetzliche Klassifikation. Sie hilft dir, den erlaubten Verarbeitungsweg vor dem Import zu bestimmen.

Schreibe auf, wer die Originaldateien lesen darf und ob eine Kopie für die KI-Verarbeitung zulässig ist. Ein beruflich berechtigter Zugang erlaubt nicht automatisch jeden privaten Import. Auch Informationen über andere Menschen bleiben relevant, wenn du sie auf deinem eigenen Rechner verarbeitest.

Verwende für die ersten Versuche öffentliche oder erfundene Daten. So kannst du eine Fehlkonfiguration beobachten, ohne bereits vertrauliche Inhalte zu verteilen. Ein Test mit einer harmlosen Datei ist ein vollständiger technischer Test, wenn er dieselben Formate und Schritte durchläuft.

3.2 Die Verarbeitungskette zeichnen

Zeichne auf einem Blatt den Weg von der Datei zur Antwort. Typische Stationen sind Oberfläche, Textextraktion, Embedding-Modell, Index, Sprachmodell und Verlaufsspeicher. Ergänze alle optionalen Dienste, etwa Übersetzung, Websuche, Bilderkennung und Sprachausgabe. Jeder externe Endpunkt gehört in die Zeichnung.

Ein lokal laufendes Sprachmodell genügt nicht, wenn ein vorgeschalteter OCR-Dienst Seiten an einen Server überträgt. Ebenso kann eine lokale Suche ihre Treffer an einen externen Chatdienst senden. Prüfe die Auswahl des jeweiligen Anbieters für jede Funktion separat. Die Bezeichnung „lokal“ in einem Produktnamen beantwortet diese Frage nicht.

Füge Sicherung und Export hinzu. Ein verschlüsselter Systemdatenträger schützt keine unverschlüsselte Kopie auf einem gemeinsam verwendeten USB-Stick. Eine Diagnose-Datei kann Prompts oder Pfade enthalten. Der Datenweg endet erst dort, wo auch Kopien und Protokolle berücksichtigt sind.

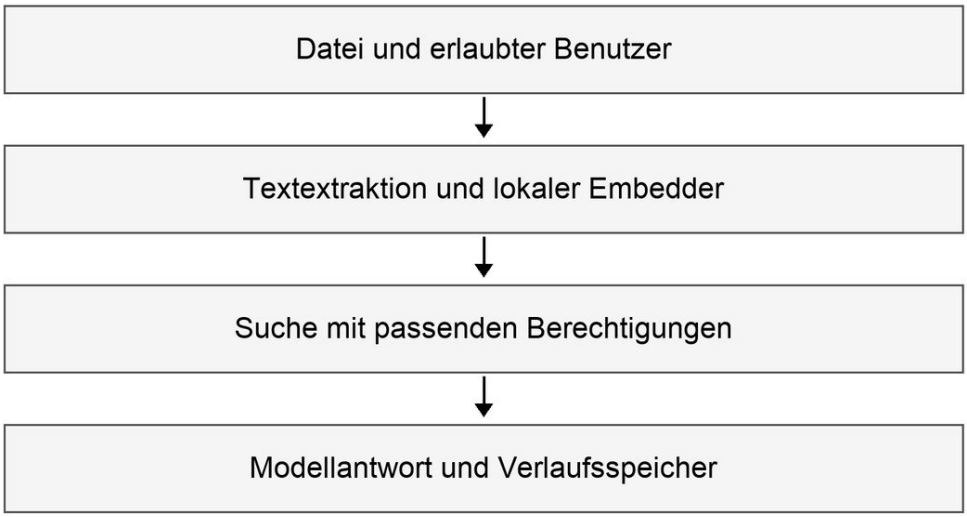


Abbildung 3.1 Der vollständige Dokumentenweg

3.3 Den wahrscheinlichen Gegner benennen

Für einen einzelnen Arbeitsplatz sind häufig versehentliche Freigaben, gestohlene Geräte und schadhafte Erweiterungen realistischer als ein hoch spezialisierter Angriff. Für einen öffentlich erreichbaren Server kommen fremde Anfragen, automatisierte Angriffe und erschöpfte Ressourcen hinzu. Deine Maßnahmen sollten zu diesem tatsächlichen Profil passen.

Betrachte auch erlaubte Benutzer. Ein Familienkonto soll beispielsweise gemeinsame Handbücher lesen, aber keine privaten Steuerunterlagen. Ein Administrator kann technisch weitreichende Einsicht besitzen. Versprich anderen Benutzern deshalb keine absolute Vertraulichkeit vor der Person, die Rechner, Sicherungen und Anwendung kontrolliert.

Eine Dokumentenquelle kann absichtlich Anweisungen enthalten, die das Modell von seiner Aufgabe abbringen sollen. Dieses Risiko heißt Prompt Injection. Es wird besonders wichtig, sobald die Anwendung Werkzeuge ausführt. Auch eine Datei auf deiner eigenen Festplatte kann zuvor aus einer unzuverlässigen Quelle gekommen sein. [D51]

3.4 Schutzmaßnahmen der richtigen Ebene zuordnen

Eine Firewall begrenzt Netzwerkzugriffe. Eine Anmeldung unterscheidet Benutzer. Eine Berechtigungsprüfung entscheidet, welche Daten ein Benutzer lesen darf. Verschlüsselung schützt bestimmte Übertragungs- oder Speicherwege. Ein Systemprompt beeinflusst dagegen das Verhalten des Modells. Diese Maßnahmen erfüllen verschiedene Aufgaben.

Prüfe die Grenze jeweils mit einem negativen Test. Ein fremdes Konto versucht, eine private Sammlung zu öffnen. Ein zweites Gerät versucht, einen nur lokal vorgesehenen Port zu erreichen. Ein Dokument enthält eine Aufforderung zum Datenexport. Ein positiver Chatversuch beweist keine dieser Schutzwirkungen.

Beginne mit wenigen Funktionen. Ohne Werkzeuge, Fernzugriff und automatische Importe ist die Verarbeitungskette leichter zu verstehen. Füge eine Funktion erst hinzu, wenn du ihren Zweck, ihre Berechtigung und ihren Rückbau erklären kannst. Diese Beschränkung erleichtert später auch die Fehlersuche.

3.5 Löschung und Aufbewahrung vorher festlegen

Eine importierte PDF kann mehrere Ableitungen erzeugen: extrahierten Text, Textabschnitte, Vektoren, Vorschaubilder und Chatantworten. Wenn du die PDF entfernst, bleiben diese Ableitungen möglicherweise erhalten. Prüfe die Löschfunktion der Anwendung mit einer eindeutig erkennbaren Übungsinformation.

Lege eine passende Aufbewahrungsdauer fest. Für einen einmaligen Entwurf reicht es oft, nur das geprüfte Ergebnis aufzubewahren. Ein Wissensarchiv benötigt zusätzlich Versionsdaten und Herkunft. Je mehr Verlauf du dauerhaft speicherst, desto mehr musst du bei Auskunft, Export und Wiederherstellung berücksichtigen.

Schreibe die Entscheidung kurz auf. „Ungeprüfte Übungsimporte werden nach dem Test entfernt; freigegebene Dokumente werden mit Versionsdatum geführt“ ist konkreter als „Datenschutz ist wichtig“. Im laufenden Betrieb kannst du daran prüfen, ob die tatsächliche Speicherung noch dem vorgesehenen Zweck entspricht.

4 Hardware ohne Fehlkauf auswählen

4.1 Vorhandene Ausstattung aufnehmen

Notiere Betriebssystem, Prozessor, Arbeitsspeicher, Grafikeinheit und freien SSD-Speicher. Unter macOS findest du die grundlegenden Angaben über das Apple-Menü bei „Über diesen Mac“. Unter Windows helfen die Systeminformationen und der Task-Manager. Unter Linux unterscheiden sich die grafischen Werkzeuge je nach Distribution; die Herstellerkennung der GPU solltest du ausdrücklich festhalten.

Verwechsle den Grafikspeicher nicht mit dem allgemeinen Arbeitsspeicher. Eine dedizierte GPU besitzt meist eigenen VRAM. Bei Apple Silicon nutzen CPU und GPU gemeinsamen Speicher. Davon benötigen Betriebssystem und Anwendungen ebenfalls einen Teil. Die gesamte beworbene Kapazität steht nicht ausschließlich dem Modell zur Verfügung.

Prüfe zuerst, ob ein kleines Modell mit einem überschaubaren Kontext auf der vorhandenen Hardware arbeitet. Eine CPU-Ausführung kann für erste Versuche genügen, auch wenn sie langsam ist. Du brauchst diesen Versuch, um deine tatsächliche Wartebereitschaft und die Qualität der Aufgabe einschätzen zu können.

4.2 Speicherbedarf in Bestandteile zerlegen

Der laufende Bedarf besteht aus mehr als den Modellgewichten. Hinzu kommen Kontextspeicher, Laufzeitpuffer und weitere Modelle für Embeddings oder Bildverarbeitung. Bei mehreren gleichzeitigen Anfragen kann zusätzlicher Bedarf entstehen. Welche Bestandteile gemeinsam genutzt werden, hängt von Laufzeit und Modell ab.