

Stack-Übersicht

Womit wir bauen, warum — und was sonst geht · Stand 04.10.2026

Eine funktionierende Kombination — nicht die einzige

Dieser Stack läuft produktiv, auf einem Server, ohne Entwicklerteam. Er ist kein Standard und keine Empfehlung für jeden Fall. Lesen Sie die Spalte „Warum“ — die Begründung ist übertragbar, die Produktnamen sind es nicht. Ab der Oberfläche beginnt der Eigenbau: dort sitzen Ihre Abläufe, und die hat niemand sonst.

Die vier Bausteine — mehr ist es nicht

Baustein	Was er tut	Beispiele
1 · Laufzeit	führt das Modell auf Ihrer Maschine aus	Ollama · vLLM · llama.cpp
2 · Modell	denkt	Mistral · EuroLLM · Qwen · Llama
3 · Oberfläche	da arbeiten die Menschen	OpenWebUI · eigene App
4 · Wissen	macht, dass es Ihr Unternehmen kennt	Volltext · Vektor · Knowledge Graph

Der fünfte Baustein ist keine Technik: Freigaben, Protokoll, Verantwortung. Den bekommen Sie bei keinem Anbieter dazu — er entsteht durch Entscheidungen, nicht durch Installation.

Unser Stack, Zeile für Zeile

Reverse-Proxy	Caddy
WARUM	Holt und erneuert TLS-Zertifikate von selbst. Genau ein Dienst mit offenen Ports — alles andere liegt dahinter.
SONST	nginx (mehr Kontrolle, mehr Konfiguration) · Traefik (stärker bei vielen Containern)
Backend	Node + Express, Eigenbau
WARUM	Die Fakten kommen aus dem Code, das Modell formuliert nur. Wer das trennt, bekommt keine erfundenen Zahlen.
SONST	Python mit FastAPI (stärker im Daten- und KI-Umfeld) · Go (schneller, strenger)
Oberfläche	React, Eigenbau
WARUM	Hier sitzen Ihre Abläufe: Freigaben, Rollen, Kartenansichten. Genau das passt in keine Fertiglösung ganz hinein.
SONST	OpenWebUI (in Minuten startklar) · LibreChat · Flask mit serverseitigen Seiten
Lokales Modell	Ollama
WARUM	Ein Befehl, läuft auf Laptop und Server gleich. Zum Starten unschlagbar einfach.
SONST	vLLM (schneller bei vielen gleichzeitigen Nutzern) · llama.cpp (sparsam) · LM Studio (grafisch)
EU-Modell	Mistral, Paris
WARUM	Festes Register im Code: nur EU oder lokal. Kein Schalter in der Oberfläche kann das ändern.
SONST	EuroLLM · Teuken · Apertus (Schweiz) — Lizenz jeweils prüfen, nicht jede erlaubt kommerzielle Nutzung
Volltextsuche	Meilisearch
WARUM	Für das Exakte: Aktenzeichen, Produktnummern, Namen. Findet, was wörtlich dasteht — das kann Vektorsuche nicht.
SONST	Elasticsearch (mächtiger, schwerer) · Typesense · PostgreSQL-Volltext, wenn Postgres ohnehin läuft
Vektor & Graph	Vektoren als Datei, Graph im Code
WARUM	Bei einem Nutzer und überschaubarem Bestand braucht es keine eigene Datenbank. Das ist eine bewusste Grenze, kein Versäumnis.
SONST	pgvector (wenn Postgres da ist) · Qdrant · Chroma · Neo4j für echte Graphen

Betrieb	Docker Compose
WARUM	Fünf Container, eine Datei, ein Server. Nachvollziehbar für jemanden, der nicht dabei war.
SONST	Kubernetes — aber erst ab einem Team, das es betreibt. Vorher kostet es nur Komplexität.
Sicherung	age + rclone, Schlüssel im Haus
WARUM	Nächtlich, verschlüsselt, außer Haus abgelegt — aber der Schlüssel bleibt bei Ihnen. Sonst sichern Sie für jemand anderen mit.
SONST	restic · borg — beide gleichwertig, Geschmacksfrage
Werkstatt	Git + eigenes Repository
WARUM	Jede Änderung nachvollziehbar, jeder Stand wiederherstellbar. Ohne das ist ein Eigenbau nicht wartbar.
SONST	GitHub · GitLab · Gitea oder Forgejo, wenn es im eigenen Haus liegen soll — siehe Bauanleitung

Modell und Maschine — was worauf läuft

Maschine	Modellgröße	Dafür reicht es
Laptop / kleiner Server, CPU, 8–16 GB	3–4 Mrd. Parameter, 4-Bit	Zusammenfassen, sortieren, einfache Fragen. Langsam, aber ehrlich nutzbar.
Eine Grafikkarte, 24 GB VRAM	bis ca. 24 Mrd. Parameter, 4-Bit	Hier wird es alltagstauglich. Mistral Small, EuroLLM 9B.
Server, 48 GB+ oder mehrere GPUs	große Modelle, volle Genauigkeit	Mehrere Nutzer gleichzeitig, anspruchsvolle Fachtexte.

Faustregel für den Speicherbedarf

Bei 4-Bit-Quantisierung: Milliarden Parameter × 0,6 = benötigte Gigabyte VRAM. Ein 24-Mrd.-Modell braucht also rund 14 GB. Und das Gegenteil der verbreiteten Annahme gilt: Nehmen Sie das KLEINSTE Modell, das reicht — nicht das größte, das noch läuft.

Wissen — drei Methoden, die sich ergänzen

- Volltextsuche für das Exakte. Aktenzeichen, Produktnummern, Namen — findet, was wörtlich dasteht.
- Vektorsuche für das Gemeinte. Findet den Vorgang auch, wenn die Frage anders formuliert war.
- Knowledge Graph für Zusammenhänge. Wer gehört zu welcher Firma, welches Dokument zu welchem Vorgang.
- Alle drei zusammen sind in komplexem Unternehmenswissen robuster als jede einzelne Methode.
- Embeddings lokal erzeugen — sonst wandern genau die Dokumente nach außen, die Sie schützen wollten.

Fertiglösung oder Eigenbau?

Fertiglösung

OpenWebUI & Co.

DAFÜR

In Minuten startklar. Sieht aus wie ChatGPT, also holen Sie Menschen damit ab. Perfekt, um das Prinzip zu zeigen und Akzeptanz zu gewinnen.

ABER

Ihre Abläufe passen nie ganz hinein. Freigaben, Rollen und Anbindungen enden an der Grenze des Produkts — und die verschiebt ein anderer.

Eigenbau

Ihre Oberfläche, Ihre Regeln

DAFÜR

Freigaben, Rollen, CRM- und Postfach-Anbindung, Protokoll — alles so, wie Ihr Betrieb tatsächlich läuft, nicht wie ein Anbieter es vorsieht.

ABER

Kostet Zeit und braucht jemanden, der es betreibt und patcht. Dafür wächst es mit und ist Ihr Asset, nicht die Roadmap eines Anbieters.

Die Entscheidungsregel

Fertiglösung, wenn Standardprozesse reichen. Eigenbau, wenn Ihre Abläufe, Freigaben oder Integrationen selbst Teil des Wettbewerbsvorteils sind. Und ein dritter Weg, der meistens der richtige ist: mit einer Fertiglösung starten, um Akzeptanz zu gewinnen — und erst dann selbst bauen, wenn Sie belegen können, was nicht hineinpasst.

Die Klammer — fünf Regeln, die im Code stehen

Diese fünf Regeln sind kein Betriebshandbuch, sondern Code. Was nur im Handbuch steht, hält niemand ein.

1 Lesen ist frei. Schreiben hat einen Schalter.

Jeder Weg nach außen bekommt einen eigenen Schalter, Vorgabe: aus. Werkzeug ohne Mandat? Dann startet der Server gar nicht erst.

2 Der Mensch drückt den Knopf.

Die KI entwirft. Mails senden und Kundendaten löschen gibt es nur als Knopf für Menschen, nie als KI-Werkzeug.

3 Fremdtext ist Daten, nie Anweisung.

Wer Text von außen gelesen hat, darf im selben Schritt nichts mehr schreiben. Das entschärft Prompt Injection an der Wurzel.

4 Nur erlaubte Anbieter — erzwungen.

Nur EU oder lokal. Nicht-europäische Anbieter sind aus dem Code entfernt, nicht bloß abgeschaltet.

5 Protokoll ohne Inhalt.

Hash-Kette, jede Zeile signiert — aber ohne Mailtexte oder Kundendaten im Log. Nachvollziehbarkeit ohne zweite Datenhalde.

Und die Regel, die man am leichtesten vergisst

Jede Komponente patchen — auch die quelloffene. Offen heißt nicht sicher, es heißt nur nachprüfbar. Ein Update-Plan gehört ab Tag 1 dazu, nicht ab dem ersten Vorfall.