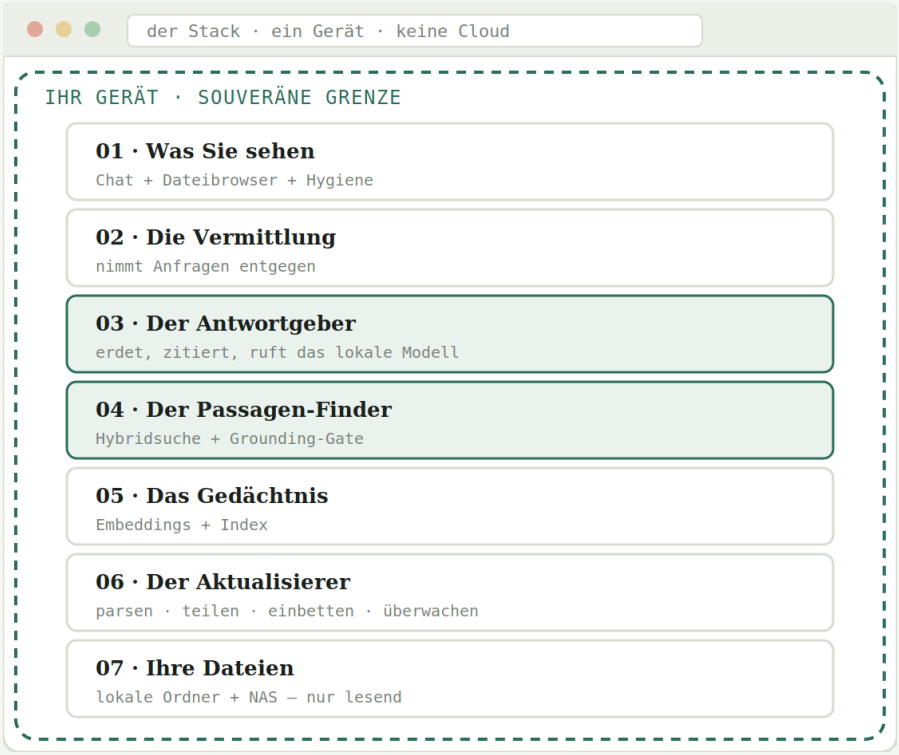
SIM RESEARCH INSTITUTE AG

EIN WHITEPAPER · 2026

The Tidy Machine

Aufbau eines souveränen, selbst betriebenen KI-Assistenten für Ihre eigenen Dokumente — ein lokales Wissensportal für Privatpersonen und KMU.



SOUVERÄN · LOKAL · CHAT-FIRST · GEERDET

INHALT

Was Sie erwartet

01 Das Problem: Sie finden nicht, was Sie haben
02 Die Grundidee: ein Bibliothekar für Ihre Dateien
03 Für wen es ist und wo es läuft
04 Gestaltungsprinzipien
05 Wie es funktioniert
06 Architektur: ein Gerät, in Schichten
07 Wie eine Frage fliesst
08 Technologie-Stack
09 Betrieb & Hardware
10 Datenklassifizierung: die vier Stufen
11 Datenschutz und Datensouveränität
12 Sicherheit & Governance für KMU
13 Umsetzung: der Prototyp
14 Evaluation und Qualität
15 Grenzen und ehrliche Hinweise
16 Roadmap / Ausbaustufen
17 Erste Schritte
– Glossar

Zu finden, was man bereits besitzt, gehört zu den am meisten unterschätzten Problemen des Alltags am Computer. Dieses Papier beschreibt ein lokales Wissensportal: ein kleines, selbst gebautes KI-System, das Ihre eigenen Dokumente liest und es erlaubt, ihnen Fragen in natürlicher Sprache zu stellen — mit Antworten, die in den Quelldateien verankert sind und auf sie zurückverweisen. Entscheidend: Modell, Suche und Antwort laufen vollständig auf Ihrer eigenen Hardware, kein Dateiinhalt verlässt je Ihre Kontrolle.

ZUSAMMENFASSUNG

Fragen Sie Ihre Dateien — auf Ihrem eigenen Gerät

Dateien häufen sich über Ordner, Laufwerke und Geräte schneller an, als sie irgendjemand ordnen kann, und herkömmliche Suche hilft nur, wenn man die genauen Wörter kennt. Ein **lokales Wissensportal** liest Ihre eigenen Dokumente und lässt Sie ihnen Fragen in natürlicher Sprache stellen — mit Antworten, die in den Quelldateien verankert sind und auf sie zurückverweisen.

Drei Eigenschaften prägen den Ansatz. Erstens ist er **retrieval-augmentiert**: das Sprachmodell antwortet ausschliesslich aus Passagen, die aus Ihren Dateien abgerufen wurden, nie aus seinem allgemeinen Training. Zweitens ist er **geerdet**: ist die Beleglage zu schwach, verweigert es eine Antwort, statt zu raten. Drittens, und das ist das Wichtigste, ist er **vollständig lokal**: Modell, Suche und Antwort laufen auf Ihrer eigenen Hardware, kein Dateiinhalt verlässt je Ihre Kontrolle. Die Architektur ist klein genug, dass eine einzelne Person sie betreiben kann, und skaliert dennoch sauber von einer Festplatte über ein NAS bis zu einem privaten Server — gleichermassen geeignet für eine Privatperson wie für ein KMU.

TEIL 1

Das Problem: Sie finden nicht, was Sie haben

Jede lange gewachsene Sammlung von Dateien strebt zur Unordnung. Dokumente kommen ununterbrochen herein — heruntergeladen, erhalten, erzeugt, «sicherheitshalber» kopiert — und jedes braucht Sekunden, um sauber abgelegt zu werden, aber nur den Bruchteil davon, um irgendwo hingeworfen zu werden. Über Jahre und über mehrere Personen hinweg ist das Ergebnis vorhersehbar: doppelte Versionen, denen niemand traut, eingescannte Dokumente, die sich nicht durchsuchen lassen, Ordner, an deren Anlage sich niemand erinnert, und Wissen, das faktisch verschwindet, weil es niemand finden kann.

Der tiefere Schaden ist nicht der Speicherplatz, sondern **verlorenes Wissen**. Ein Betrieb, der nicht rasch beantworten kann «Was haben wir mit diesem Lieferanten vereinbart?» oder «Wo ist die aktuelle Fassung dieser Richtlinie?», zahlt täglich für diese Reibung. Einer Privatperson geht es genauso, nur in kleinerem Massstab.

Eine Aufräumarbeit hat in Wahrheit zwei Kosten, und die meisten rechnen nur mit der zweiten. Die erste ist, **zu wissen, was man hat** — bevor man entscheiden kann, was bleibt, zusammengeführt oder verworfen wird, muss man den Inhalt Tausender Dateien verstehen. Das ist der eigentlich unmögliche Teil, und deshalb beginnt das Projekt nie. Die zweite ist, **die Entscheidungen zu treffen** — dies behalten, das verwerfen —, echte, aber begrenzte Arbeit, die nur Sie leisten können. Ein lokales Wissensportal reduziert die erste Kost auf fast nichts: Das Schwierige am Aufräumen war nie das Löschen, sondern zu wissen, was man hatte.

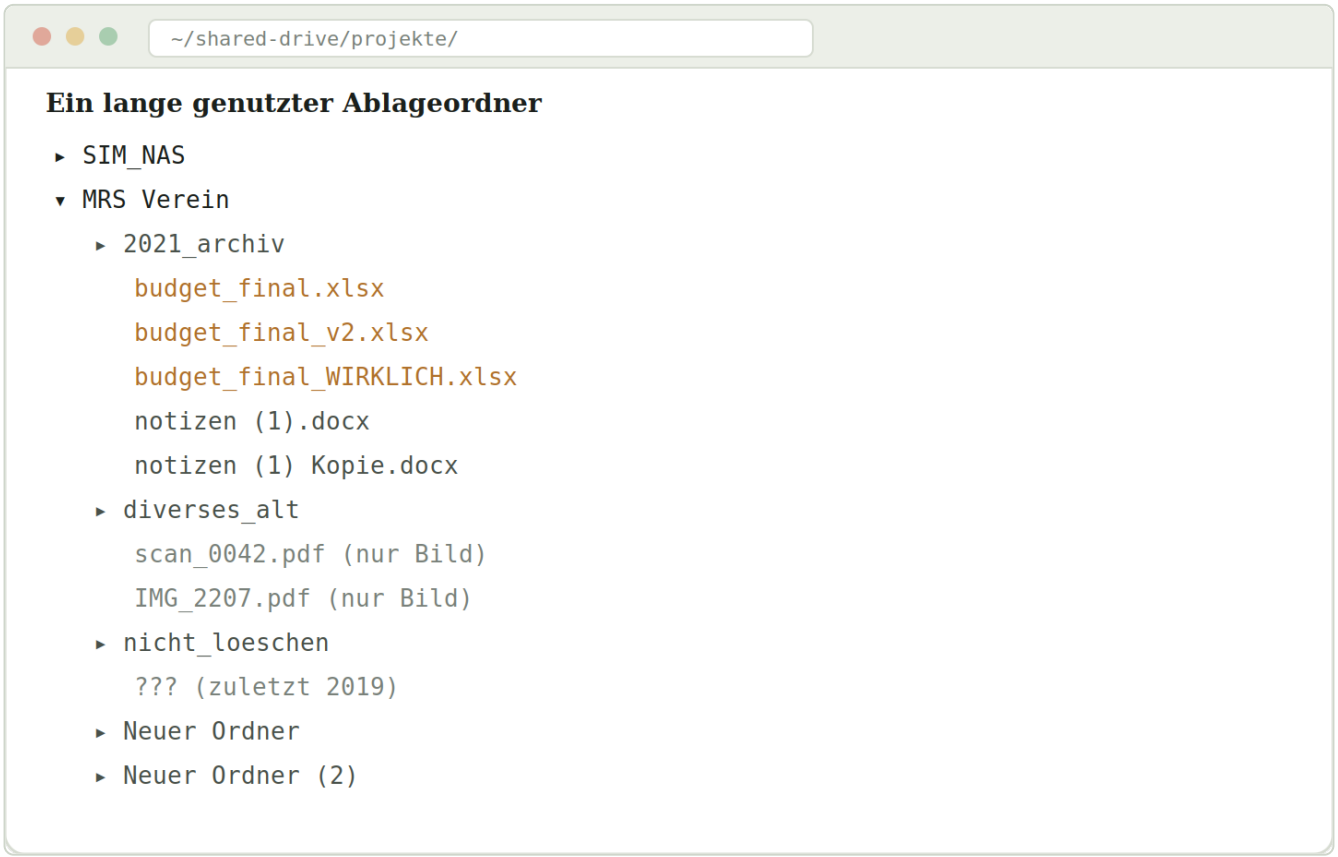


Abb. 1 — Der natürliche Zustand eines lange genutzten Ablageordners. Kein von jemandem verursachtes Unglück; einfach die Entropie bei der Arbeit. (Illustrativ.)

Herkömmliche Werkzeuge greifen auf bestimmte, behebbare Weisen zu kurz: Dateisuche braucht exakte Begriffe (sie trifft die getippten Wörter, nicht die gemeinte Bedeutung); sie kann Bilder von Text nicht lesen (eine eingescannte Seite ist unsichtbar); und sie kann nicht in natürlicher Sprache antworten (sie liefert eine Dateiliste, keine Antwort). Die Lücke, die dieses Papier schliesst, ist der Schritt von der *Suche nach Dateien* zum *Befragen Ihrer Dateien* — ohne irgendetwas in die Cloud zu schicken.

TEIL 2

Die Grundidee: ein Bibliothekar für Ihre Dateien

Stellen Sie sich einen Bibliothekar vor, dessen einzige Aufgabe es ist, jedes Dokument zu lesen, das Sie besitzen, und sich zu merken, was darin steht. Sie bitten ihn nicht, etwas wegzuwerfen oder Ihre Regale umzuräumen. Sie stellen Fragen — und der Bibliothekar findet die einschlägigen Passagen, antwortet in natürlicher Sprache und verweist auf die genauen Dokumente und Seiten.

Genau das tut ein lokales Wissensportal. Sie richten es auf Ihre Ordner; es liest und indexiert einmal alles und hält sich danach im Hintergrund aktuell. Von da an haben Sie eine einzige Oberfläche, auf der Sie:

- **Fragen** — eine Frage stellen und eine in Ihren Dateien verankerte Antwort erhalten, mit Zitaten, die direkt zur Quelle verlinken.
- **Suchen** — semantische *und* Stichwortsuche über alles Indexierte, wenn Sie die Treffer lieber selbst durchsehen.
- **Durchsuchen** — den Dateibaum, als zweite Ansicht neben dem Chat.
- **Aufräumen** — eine *Hygiene*-Ansicht, die eine Quelle scannt und exakte Duplikate, Versions-Geschwister, Müll- und Scratch-Dateien, Backup-Reste, übergrosse und veraltete Formate sowie Namens- oder Ordnerprobleme markiert — je mit Anzahl, freigebbarer Grösse und sicheren Aktionen (in einen Prüfordner verschieben oder löschen).

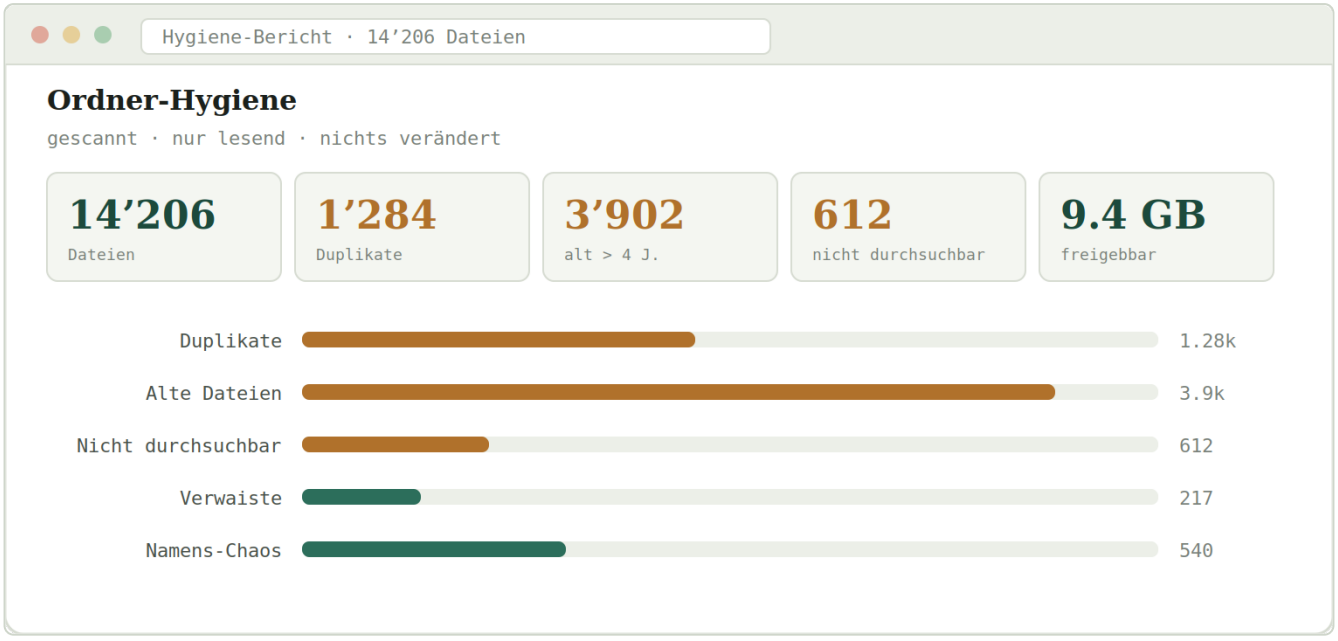


Abb. 2 — Ein illustrativer Hygiene-Bericht. Die Zahlen sind erfunden, doch die Kategorien sind genau jene, die ein Dateisystem verrotten lassen. Der Zweck des Berichts: ein vages Unbehagen in eine endliche To-do-Liste zu verwandeln.

Warum das heute möglich ist. Zwei Dinge sind zugleich reif geworden: die Fähigkeit, Text in einen Fingerabdruck seiner *Bedeutung* zu verwandeln (sodass eine Frage die richtige Passage auch bei anderer Formulierung findet), und leistungsfähige Sprachmodelle, klein genug, um auf einer gewöhnlichen Workstation zu laufen — kein Rechenzentrum, kein Konto, keine Gebühr. Zusammen lassen sie den ganzen Bibliothekar auf einem einzigen Gerät wohnen.

> IN EINEM SATZ

> Es ist ein neuer Vordereingang zu Dateien, die Sie bereits haben — kein neuer Ablageort. Ihre Dateien bleiben genau dort, wo sie sind; das Portal liest und indexiert sie nur.

TEIL 3

Für wen es ist und wo es läuft

Dasselbe System bedient zwei Zielgruppen auf einem Kontinuum: **Privatpersonen** (eine persönliche Wissensbasis über Dokumente, Korrespondenz, Scans und Notizen, auf einem Laptop oder Heimgerät) und **KMU** (eine gemeinsame Wissensoberfläche über Verträge, Richtlinien, Projektdateien und Akten, wo das Gedächtnis der Organisation heute in verstreuten Ordnern und in den Köpfen der Leute steckt).

Dass ein Design für beide passt, liegt daran, dass es **mit dem Speicher skaliert, nicht mit der Komplexität**. Die Referenzumsetzung nutzt ein kleines NAS als Dateiquelle, doch die Architektur ist über drei Betriebsstufen identisch:

Stufe	Wo die Dateien liegen	Typische Nutzung	Hinweise
Lokale Festplatte	Festplatte eines Geräts	Privatperson	Am einfachsten; alles auf einem Gerät.
NAS (Referenz)	Netzlaufwerk (SMB / NFS)	Person / kleines Team	Gemeinsame Quelle, nur lesend eingebunden.
Privater Server	Server im Haus	KMU	Mehr Rechenleistung (z. B. GPU), grösserer Bestand.

In jeder Stufe läuft das Portal **lokal** und die Dateiquelle ist **nur lesend**. Eine Stufe höher zu gehen ändert die Menge an Speicher und Rechenleistung, nicht das Design.

TEIL 4

Gestaltungsprinzipien (nicht verhandelbar)

1. **Souverän / vollständig lokal.** Alle Einbettung und alle Inferenz laufen auf Ihrer eigenen Hardware über Ollama. Keine Cloud-Abhängigkeit, keine Exposition gegenüber ausländischen Datengesetzen; kein Dateiinhalt verlässt je das Gerät.
2. **Massgeschneiderter Zugriff.** Für eine Person: Einzelnutzer, keine Konten, jeder Dienst an `localhost` gebunden. Für ein KMU: Zugriff auf Datei- und Netzwerkebene geregelt, statt nachträglich aufgesetzt.
3. **Chat-first.** Das Gespräch ist die primäre Oberfläche; der Dateibrowser ist sekundär.
4. **Schrittweise und überprüfbar.** Phase für Phase gebaut und eingeführt, mit einem Kontrollpunkt vor jedem Schritt — kein Urknall-Zusammenbau, und jede Antwort auf eine Quelle zurückführbar.

TEIL 5

Wie es funktioniert

Zwei Reisen beschreiben das ganze System: Dokumente, die in einen Index wandern, und eine Frage, die durch diesen Index zu einer geerdeten Antwort reist.

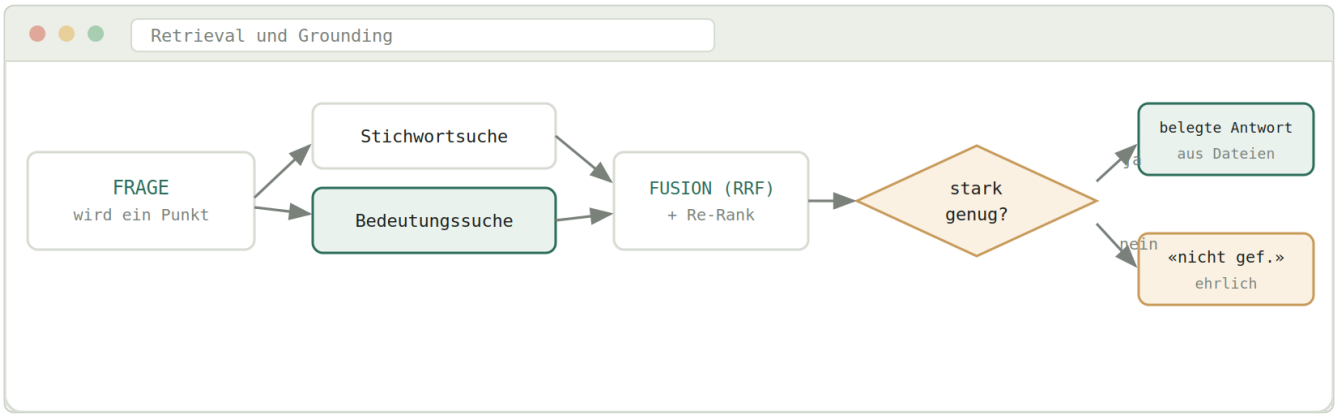


Abb. 5 — Eine Frage läuft durch Stichwort- und Bedeutungssuche; die Treffer werden fusioniert und neu gereiht; dann fragt ein Grounding-Gate, ob das Gefundene wirklich stark genug ist. Wenn ja, wird die Antwort aus diesen Passagen geschrieben und belegt. Wenn nein, sagt der Assistent es.

5.1 Einlesen — Ihre Dateien in einen Index überführen

Jede Datei macht eine fünfstufige Reise. **Entdecken:** Das Portal bemerkt, was seit dem letzten Mal neu oder verändert ist, indem es Änderungszeit und einen Inhalts-Hash (SHA-256) vergleicht, sodass Unverändertes übersprungen wird. **Parsen:** Text wird aus PDFs, Textdokumenten, Tabellen und Präsentationen extrahiert; ist eine Datei ein eingescanntes Bild ohne Textebene, liest OCR das Bild, damit es durchsuchbar wird. **Teilen:** Jedes Dokument wird in kohärente Passagen zerschnitten, an natürlichen Nahtstellen, nie mitten im Wort, jede mit einer Herkunfts-Markierung versehen. **Einbetten:** Jede Passage wird durch ein lokales Einbettungsmodell in einen Zahlen-Fingerabdruck ihrer *Bedeutung* verwandelt. **Indexieren:** Passagen und Vektoren werden in einer Vektordatenbank gespeichert, bereit für schnelle Nachbarschaftssuche.

> WAS EINE EINBETTUNG IST
>
> Eine Einbettung verwandelt eine Textpassage in eine Zahlenliste, die so angeordnet ist, dass Passagen über Ähnliches nahe beieinander liegen — sodass «Ferienanspruch» eine Passage über «Urlaubsguthaben» findet, auch ohne gemeinsame Wörter. Frage und Dokumente müssen mit *demselben* Modell gemessen werden; wechselt das Modell, muss neu indexiert werden.

5.2 Abruf — die richtigen Passagen finden, ehrlich

Eine Frage wird eingebettet und durch **Hybridsuche** geschickt: eine dichte (Bedeutung) und eine spärliche / Stichwort-Suche zugleich. Die beiden Trefferlisten werden mit Reciprocal Rank Fusion verschmolzen, was sowohl die Trefferquote der Bedeutung als auch die Präzision exakter Begriffe einfängt — Referenzen, Codes, Eigennamen, die rein semantisch untergehen. Ein optionaler Re-Ranker liest die besten Kandidaten erneut und ordnet sie nach echter Relevanz.

Dann der entscheidende Schritt — das **Grounding-Gate**. Jede abgerufene Passage trägt einen Ähnlichkeitswert; fällt der beste unter eine Schwelle (Cosinus ≥ 0.55 im Referenzbau), wertet das System dies als «kein belastbarer Beleg» und lässt *leeren* Kontext in den Prompt — was zuverlässig eine Verweigerung erzwingt statt einer Vermutung.

5.3 Erzeugung — eine geerdete, belegte Antwort

Überstehen gute Passagen das Gate, schreibt das lokale Sprachmodell eine Antwort **ausschliesslich aus diesen Passagen**, unter strenger Vorgabe: nur die gelieferten Auszüge nutzen, jede Aussage belegen und klar verweigern, wenn es keine Grundlage gibt. Übersteht nichts das Gate, gibt das Portal ein ehrliches «nicht gefunden» zurück. Die Verweigerung ist ein Merkmal, kein Versagen: Eine Antwort über Ihre Dateien soll aus Ihren Dateien kommen — oder gar nicht.

Abb. 6 — Eine geerdete Antwort. Sie antwortet in natürlicher Sprache, doch jede Aussage ist durch ein Zitat belegt, das auf die echte Datei verlinkt — prüfbar mit einem Klick. (Illustrativ.)

Abb. 7 — Die Verweigerung in Aktion. Sie deutet sogar an, warum die Antwort fehlen könnte — ein nicht durchsuchbarer Scan oder ein nicht indexierter Ordner —, was oft der nächste nützliche Hinweis ist.

5.4 Den Index aktuell halten

Lokale Ordner senden verlässliche Änderungsereignisse, sodass ein Beobachter den Index im Moment der Änderung aktualisiert. Netzfreigaben sind dafür unzuverlässig, daher wird das NAS nach Zeitplan (und auf Abruf) neu gescannt, wobei Änderungen über Änderungszeit und Inhalts-Hash erkannt werden, sodass nur Neues oder Verändertes erneut verarbeitet wird.

TEIL 6

Architektur: ein Gerät, in Schichten

Das Portal ist ein kleiner Stapel zusammenarbeitender Schichten, jede mit einer Aufgabe, alle in einem Gerät. Eine Frage reist *hinunter*; eine belegte Antwort wieder *hinauf*; Dateien wandern *hinauf* von der Quelle in den Index. Nichts überschreitet die äussere Grenze nach aussen.

Schicht	Aufgabe	Komponenten
01 · Präsentation	Geteilte Ansicht — Chat + Dateibrowser; Zitate öffnen die echte Datei.	Vite + React + Tailwind
02 · Anwendung	Die Vermittlung: nimmt Anfragen entgegen, koordiniert die Schichten, an localhost gebunden.	FastAPI + uvicorn
03 · Erzeugung	Baut den Prompt, erzwingt Erdung, fügt Zitate zusammen, ruft das lokale Modell.	Ollama (Qwen2.5 / Llama 3.x)
04 · Abruf	Hybridsuche, RRF-Fusion, Re-Rank, Grounding-Gate.	bge-reranker-v2-m3 (optional)

05 · Einbettung & Index	Verwandelt Passagen in Vektoren und speichert sie für schnelle Suche.	Ollama bge-m3, Qdrant
06 · Einlesen & Sync	Parsen · teilen · einbetten · überwachen / neu scannen.	PyMuPDF, python-docx, Tesseract, watchdog
07 · Quelle / Daten	Ihre Ordner und Laufwerke — nur lesend.	Lokale Ordner, NAS (SMB / NFS)

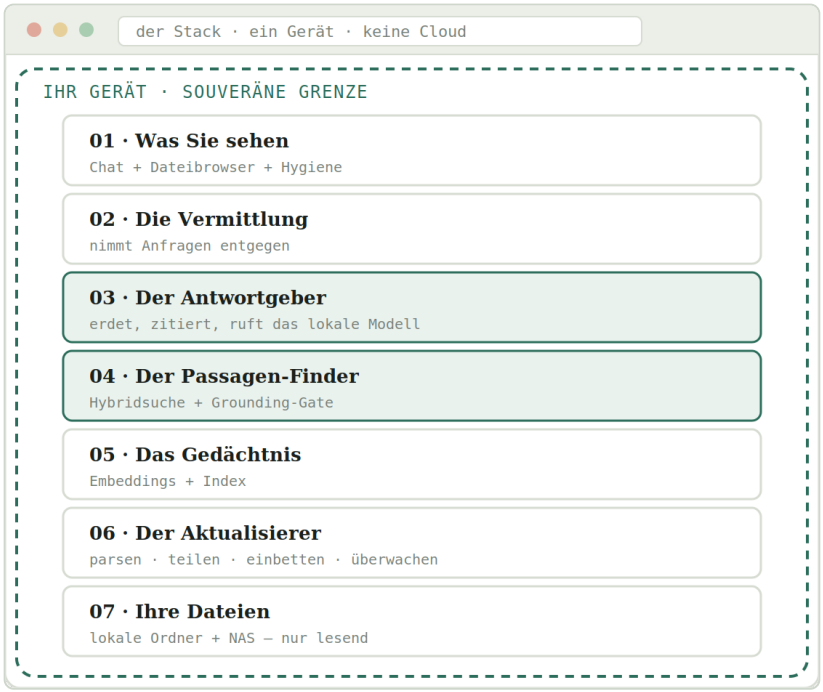


Abb. 8 — Die sieben Schichten. Von oben: was Sie sehen, die Vermittlung, der Antwortgeber, der Passagen-Finder, das Gedächtnis, der Aktualisierer und Ihre Dateien zuunterst. Die gestrichelte Linie ist der ganze Punkt — alles bleibt darin.

Bemerkenswert ist, wie klein das ist: eine Handvoll Programme auf einer Workstation — bescheiden genug, dass eine Person sie betreiben, verstehen und abschalten kann. Für diesen Zweck ist diese Bescheidenheit der Vorteil.

TEIL 7

Wie eine Frage fliesst

1. Sie tippen eine Frage in den **Chat** (Schicht 01).
2. Die **API** (02) empfängt sie und startet die Abruf-und-Antwort-Routine.
3. Die **Abrufschicht** (04) bettet die Frage ein, führt die Hybridsuche gegen den Index (05) aus, fusioniert und reiht neu und wendet das Grounding-Gate an.
4. Überstehen gute Passagen das Gate, baut die **Erzeugungsschicht** (03) daraus einen Prompt, ruft das lokale Modell und fügt die Antwort mit Zitaten zusammen. Übersteht nichts, kommt «keine geerdete Antwort gefunden».
5. Die Antwort reist wieder **hinauf** in den Chat; jedes Zitat verlinkt auf das echte Dokument.

Im Hintergrund beobachtet die **Einleseschicht** (06) lokale Ordner und scannt das NAS (07) neu — ein Aufwärtsfluss, der den Abfragepfad nie berührt.

TEIL 8

Technologie-Stack

Bereich	Wahl	Warum
Laufzeit / Inferenz	**Ollama** auf der Host-GPU	Führt Einbettungs- und Chat-Modelle lokal aus; kein API-Key, keine Cloud.
Einbettungen	**bge-m3** (alt. nomic-embed-text)	Mehrsprachig (DE/EN); dicht und spärlich — passt zu Qdrant-Hybrid.
Vektorspeicher	**Qdrant** (Docker)	Native Hybridsuche + Payload-Filter; leicht genug für eine Person.
Re-Ranker	**bge-reranker-v2-m3** (optional)	Lokal; hebt die Präzision, sobald der Basisabruf steht.
Chat-Modell	**Qwen2.5 (7B/14B)** oder Llama 3.x	An die verfügbare GPU angepasst.
Backend	**FastAPI + uvicorn** (Python 3.11+)	An 127.0.0.1 gebunden; Endpunkte /chat, /search, /files, /reindex, /health.
Parsing	**PyMuPDF, python-docx, openpyxl, python-pptx; Tesseract OCR**	OCR als Rückfallebene für Scans und Bilder.
Dateiüberwachung	**watchdog** + **APScheduler**	Verlässliche lokale Ereignisse; geplanter Rescan für die Freigabe.
Frontend	**Vite + React + Tailwind**	Geteilte Ansicht; Marken-Tokens als CSS-Variablen.
Paketierung	**Docker Compose**	Qdrant + Backend + UI als ein Stack; Ollama läuft auf dem Host.

> HINWEIS ZUM INDEXIERUNGS-DURCHSATZ
>
> Einen grossen Bestand in Einbettungen zu verwandeln ist rechenintensiv. Auf einer reinen CPU-Maschine ist das bei grossen Sammlungen langsam; ein GPU-Host macht die Voll-Indexierung praktikabel. Das ist der Hauptgrund, warum ein KMU mit grossem Archiv auf die Server-Stufe wechselt.

TEIL 9

Betrieb & Hardware

Docker Compose bündelt Qdrant, Backend und UI zu einem lokalen Stack; Ollama läuft auf dem Host. Das NAS wird über SMB/NFS eingebunden und neben lokalen Ordnern indexiert; es ist *nur Dateiquelle* und führt nie Inferenz aus. Da Netzüberwachung unzuverlässig ist, nutzt das NAS einen geplanten Rescan plus eine manuelle Neuindexierungs-Taste, wobei Änderungen über Änderungszeit und Inhalts-Hash erkannt werden. Alle Pfade, Modellnamen und Schwellen liegen in einer `.env`-Datei, und ein `/health`-Endpunkt meldet den Zustand des Stacks.

9.1 Hardware — was Sie tatsächlich brauchen

Die ehrliche Schlagzeile: **das läuft auf weit weniger Hardware, als man denkt.** Weil alles lokal und bescheiden ist, passt es auf gewöhnliche, tragbare Geräte — kein Serverraum nötig.

Stufe	Gerät	Eckdaten	Rolle
Einstieg	Lenovo ThinkPad X1 Nano	~0.9 kg · 13″ 2K · Intel Core i · integrierte Iris Xe (keine dedizierte GPU) · 16 GB RAM	Beleg, dass es auf minimaler Hardware läuft.
Power-User	Lenovo ThinkPad X1 Extreme Gen 5	Core i9-12900H (14C/20T) · 32 GB DDR5 · RTX 3080 Ti 16 GB · 1 TB SSD · 1.88 kg	Schneller, höhere Antwortqualität, tragbar.
KMU	NVIDIA DGX Spark (Founders Edition)	Blackwell-GPU · Arm-CPU (X925 + A725) · 128 GB unified · 4 TB NVMe · bis 100 Gb/s · ~1.2 kg · ≈ CHF 4’100	On-Premise-Appliance für ein Archiv.

Der Fortschritt ist real, nicht theoretisch. Das erste lauffähige Portal lief auf einem Sub-1-kg-Ultrabook mit nur integrierter Grafik — langsamer, mit bescheidenem Bestand, aber es beantwortete geerdete Fragen über seine eigenen Dateien. Ein GPU-Laptop führt ein 7–14B-Modell aus und indexiert einen persönlichen Bestand zügig. Für eine Organisation lässt der DGX Spark mit 128 GB *vereinheitlichtem* Speicher eine einzige kleine Appliance deutlich grössere Modelle halten und ausführen — und einen echten Bestand indexieren — vollständig vor Ort.

> HARDWARE, SOFTWARE ODER DAS GANZE – BETRIEBSBEREIT
>
> Die SIM Research Institute AG betreibt genau diesen Stack auf diesen Geräten. Je nach Bedarf helfen wir beim Beschaffen der Hardware, liefern einen einsatzbereiten DGX Spark, stellen die Software bereit — oder betreiben den ganzen Prozess als Managed Service. Vom einzelnen Laptop bis zur KMU-Appliance, auf jeder Stufe.

TEIL 10

Datenklassifizierung: die vier Stufen

Schutzbedarf ist kein Alles-oder-nichts. Ein einfaches, verbreitetes Modell gibt Personen und KMU ein gemeinsames Vokabular dafür, *wie sorgfältig* mit einem Dokument umzugehen ist — und es ist der Grund, warum eine lokale Architektur zählt.

Stufe	Name	Schutzbedarf	In der Praxis
1	**Öffentlich**	keiner	Frei zugänglich; keine Einschränkungen.
2	**Intern**	gering	Nur für Sie / die Organisation.
3	**Vertraulich**	erhöht	Besondere Personendaten sind mindestens vertraulich; Zugriffskonzept Pflicht.
4	**Geheim**	höchster	Nur in freigegebenen Umgebungen; strengste Zugriffs- **und Orts**-Kontrolle.

Die Stufen sind kumulativ. Der entscheidende Sprung ist der letzte. **Stufen 1–3 betreffen vor allem, *wer* die Daten sehen darf. Stufe 4 fügt eine Bedingung hinzu, *wo* sie physisch verarbeitet werden dürfen** — und diese eine Anforderung formt alles Technische um: kein Cloud-Modell, keine Cloud-Einbettungen, gar kein Weg nach aussen.

> *Ein Portal, das die strengste Stufe erfüllt, erfüllt alle übrigen gratis mit.*

Eine Person, die Ferienfotos indexiert, und ein KMU, das vertrauliche Akten hält, betreiben *dieselbe* Architektur; nur die Betriebsstufe und die Zugriffs-Governance unterscheiden sich. Die Metadaten jeder Passage können ihre Klassifizierung tragen, sodass die Herkunft jeder Antwort bis auf die Stufe nachvollziehbar ist.

TEIL 11

Datenschutz und Datensouveränität

Das ist das Herz des Designs. Bei einem Cloud-Werkzeug ist Datenschutz eine *Richtlinie* — ein Versprechen, das Sie nicht gelesen haben. Bei einem lokalen System ist Datenschutz *strukturell*: Es gibt keinen Weg nach aussen, also keinen Dienst, der Ihre Dateien *empfangen könnte*. Drei Eigenschaften machen es vertrauenswürdig: **kein Weg nach aussen** (nichts kann lecken, weil nichts gesendet wird); **nur lesende Quellen** (das Portal indexiert, verändert Ihre Dateien aber nie); und **Erdung als Vertraulichkeitskontrolle** (das Modell spricht über Ihre Inhalte nur aus Ihren Inhalten, nie aus seinem Training).

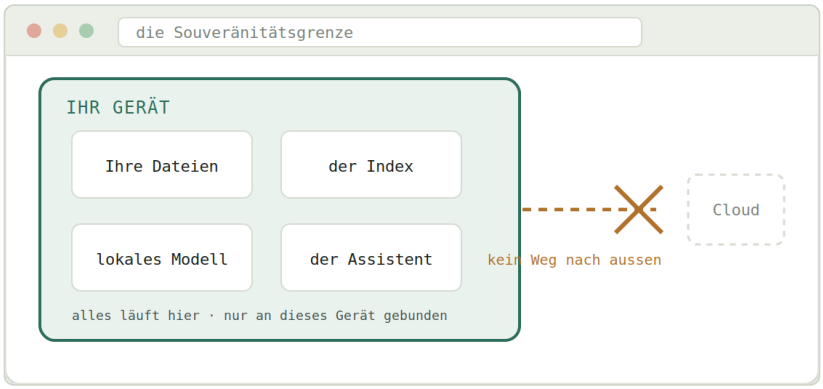


Abb. 10 — Die Souveränitätsgrenze. Jede Komponente lebt in einem Gerät; es gibt schlicht keinen Dienst, an den Ihre Dateiinhalte gesendet werden könnten. Datenschutz ist hier strukturell, kein Versprechen auf einer Einstellungsseite.

Ein praktischer Vorbehalt: Verschlüsselte Dokumente lassen sich nicht indexieren, daher muss die Quellschicht lesbare Dateien halten, deren Zugriff auf Dateisystemebene geregelt ist.

TEIL 12

Sicherheit & Governance für KMU

Das Basis-Design ist für Privatpersonen so wie es ist geeignet. Für ein KMU mit reguliertem oder hochsensiblem Material härten einige Ergänzungen es weiter, in der Reihenfolge ihres Werts:

1. **Audit-Log.** Ein anfügbares Protokoll, wer wann was gefragt und was zitiert wurde — optional hash-verkettet für Manipulationssicherheit. Günstig und konkret nützlich.
2. **Klassifizierungs-Metadaten je Passage.** Jede Passage mit ihrer Stufe markieren, damit Antworten filterbar und nachvollziehbar sind.
3. **Gehärteter Netzausgang.** Ein eigenes Docker-Netz mit expliziter Erlaubnisliste; auf einer Air-Gap-Maschine ohnehin gegenstandslos, aber günstig hinzuzufügen.
4. **Air-Gap-Installation.** Ein Offline-Paket für eine Workstation ganz ohne Internet.
5. **Governance freigegebener Anwendungen.** Die Dokumentation — Architektur, reproduzierbarer Build, Bedrohungsmodell und Testnachweis —, die eine formelle Freigabe stützt, wo verlangt.

TEIL 13

Umsetzung: der Prototyp

Ein vollständiges Portal läuft auf einer einzigen Workstation: alle sieben Schichten, Hybridabruf mit Grounding-Gate, ein strenger Zitat-Prompt mit fester Verweigerung, lokale Ordner und ein NAS als nur lesende, an `127.0.0.1` gebundene Quellen ohne Ausgangs-Inferenz, ein per nginx ausgeliefertes Frontend über einem FastAPI-Backend mit Qdrant und Ollama, alles per Docker Compose hochgefahren. Zum Zeitpunkt dieser Aufnahmen hielt der Index rund tausend Chunks über drei eingebundene Quellen.

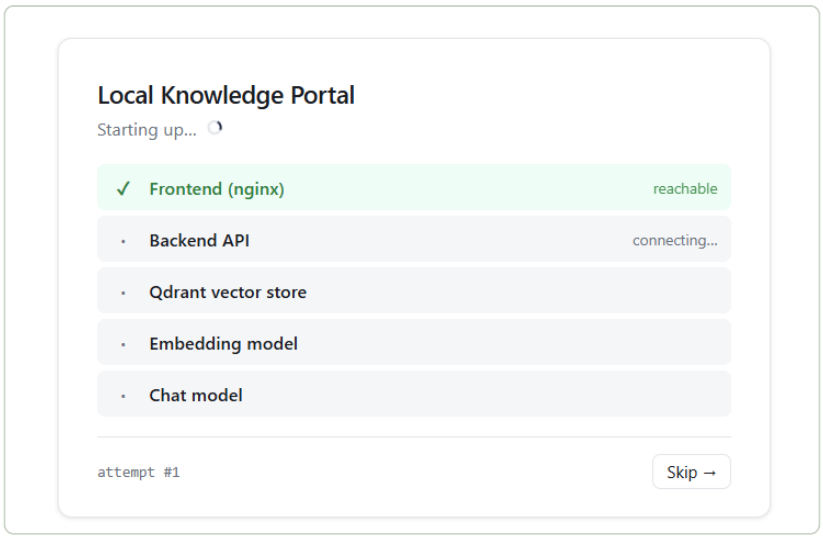


Abb. 13.1 — Das Portal startet. Der Stack kommt Schicht für Schicht hoch (Frontend → Backend → Qdrant → Einbettungsmodell → Chat-Modell); es meldet «ok» erst, wenn die Kette erreichbar ist.

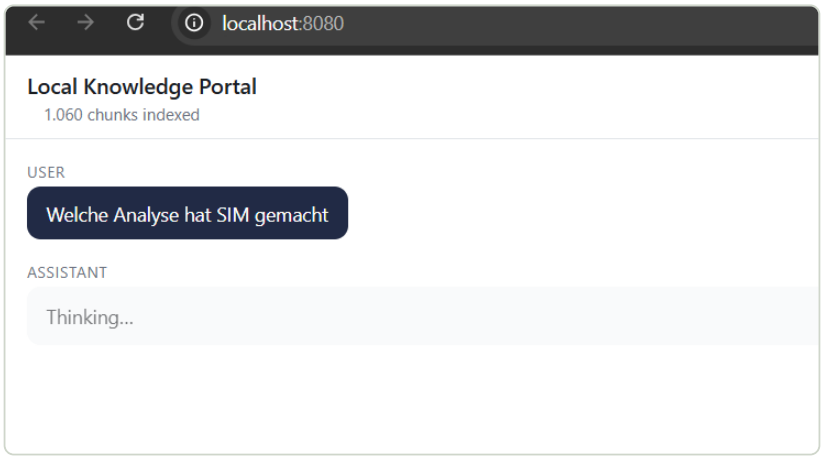


Abb. 13.2 — Eine geerdete Frage stellen. Eine Frage wird an das Portal gestellt; der Assistent verfasst eine belegte Antwort aus den indexierten Dateien. Dieselbe Form funktioniert auf Deutsch wie auf Englisch.

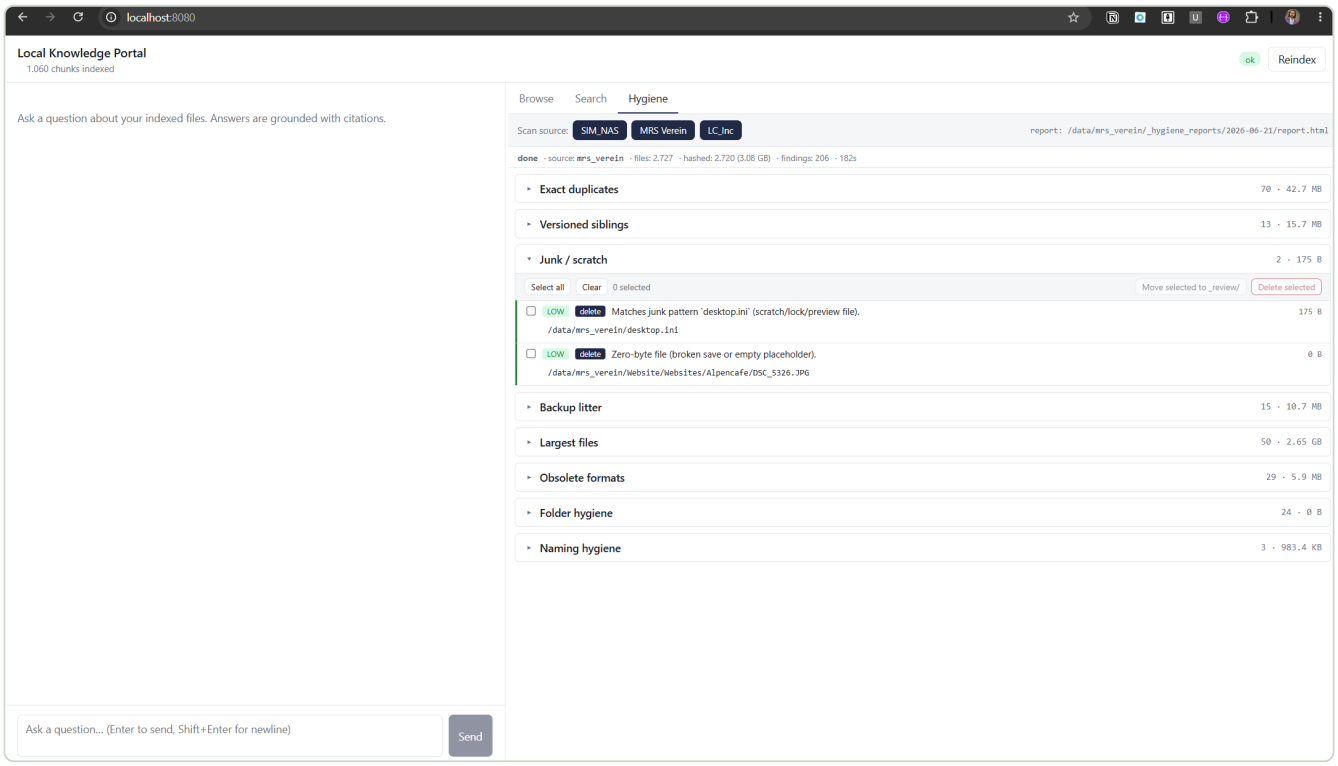


Abb. 13.3 — Der Hygiene-Bericht. Ein Scan über eine Quelle liefert Funde in handlungsfähigen Kategorien (Duplikate, Versions-Geschwister, Müll, Backup-Reste, grösste Dateien, veraltete Formate, Ordner- und Namens-Hygiene), je mit Anzahl, freigebbarer Grösse und sicheren Verschieben/Löschen-Aktionen.

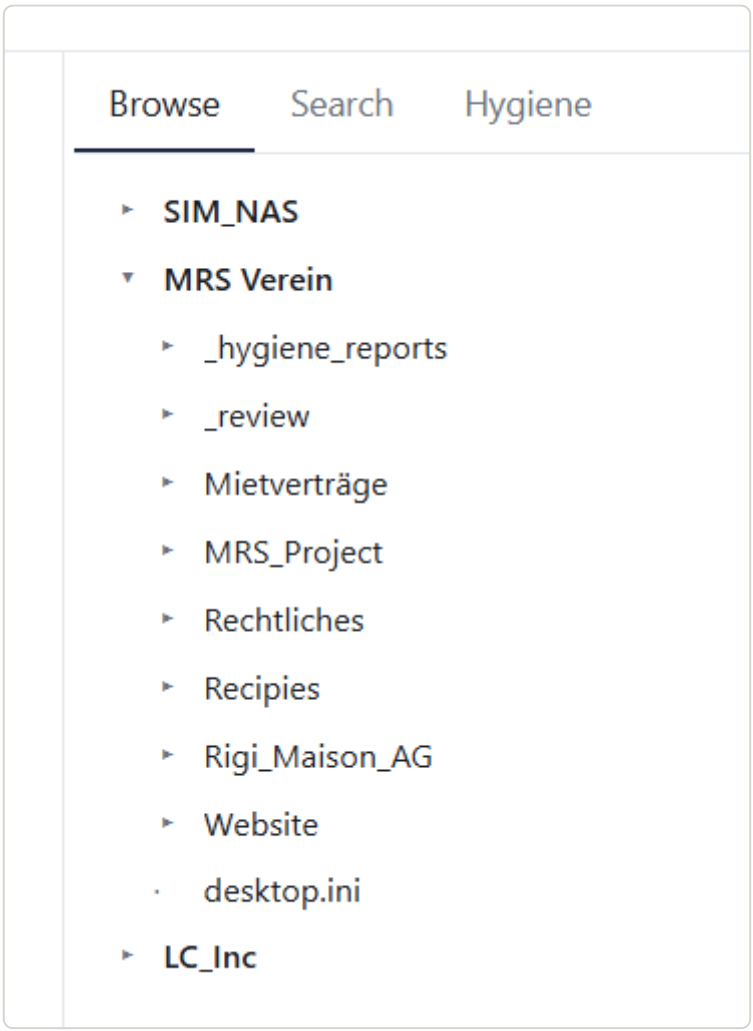


Abb. 13.4 — Die indexierten Quellen durchsuchen. Der Ordnerbaum über die eingebundenen Quellen, als zweite Ansicht neben dem Chat.

13.1 Ein Aufräumen mit dem Hygiene-Bericht

Der Bericht verwandelt eine uferlose Pflicht in eine endliche, überprüfbare Schleife. Arbeiten Sie eine Kategorie nach der anderen ab, statt alles auf einmal:

1. **Mit Duplikaten beginnen.** Pro Gruppe die kanonische Kopie behalten und den Rest in einen Prüfordner schieben — meist der grösste Gewinn auf einmal.
2. **Unsichtbares sichtbar machen.** Nicht durchsuchbare Scans per OCR mit einer Textebene versehen, damit sie nach der Neuindexierung beantwortbar werden.
3. **Alte und übergrosse Dateien sichten.** Lange unberührte und grösste Dateien als Stapel prüfen; das meiste gehört ins Archiv, nicht aufs Live-Laufwerk.
4. **Verwaiste und veraltete Formate neu zuordnen.** Das Portal fragen, was eine Datei ist, wenn ihr Name nichts sagt; Behaltenswertes ablegen, den Rest einreihen.
5. **Namen zuletzt bändigen.** Ist das Volumen erst gesunken, sind die restlichen Namens- und Ordnerprobleme klein genug, um sie von Hand zu lösen.
6. **Neu scannen und bestätigen.** Den Bericht erneut laufen lassen, die Zahlen sinken sehen, dann die nächste Kategorie wählen.

> EINE SANFTE REGEL

> **Verschieben vor Löschen.** Ein «zu prüfen»-Ordner, den Sie nach einem Monat leeren, kostet fast nichts und nimmt das einzige unumkehrbare Risiko. Das Portal hilft Ihnen, Dinge zu *finden*; es soll nie das sein, was sie verliert.

TEIL 14

Evaluation und Qualität

Ein Abrufsystem, das nie gemessen wird, wird blind optimiert. Der Referenzbau pflegt ein kleines **Evaluationsset** — rund 20–30 echte Fragen mit erwarteter Quelle — und prüft nach jeder Pipeline-Änderung, ob die richtige Quelle weiterhin in den Top-Treffern landet. Dasselbe Set kalibriert die Erdungsschwelle und entscheidet, ob optionale Verfeinerungen wirklich helfen, bevor sie aktiviert werden. Für ein KMU dient dieses Set zugleich als Testnachweis in einem Freigabeprozess.

TEIL 15

Grenzen und ehrliche Hinweise

- **Es trifft die Entscheidungen nicht.** Das Urteil darüber, was *zählt*, hängt von Kontext ab, den die Dateien nicht enthalten — das bleibt menschlich.
- **Es ist nur so gut wie das, was es lesen kann.** Verschlüsselte Dateien, exotische Formate und Ordner, auf die Sie es nie gerichtet haben, sind unsichtbar. Eine falsche oder fehlende Antwort heisst oft «diese Datei ist nicht im Index».
- **Es kann überzeugt falsch liegen.** Genau deshalb tragen Antworten Zitate — bei allem Wichtigen: durchklicken und prüfen.
- **Pflege ist eine Gewohnheit.** Dateien kommen weiter herein; ein regelmässiger Rescan hält den Index ehrlich. Die einmalige Einrichtung ist das Schwierige; der Unterhalt ist günstig.

TEIL 16

Roadmap / Ausbaustufen

Grob nach Wirkung pro Aufwand: ein Audit-Log; Klassifizierungs-Metadaten je Passage; gehärteter Netzausgang; ein Air-Gap-Installationspfad; GPU-Host-Migration für grosse Bestände; und, wo die Regulierung es verlangt, die Governance-Dokumentation für freigegebene Anwendungen. Auf der Abrufseite: das Evaluationsset pflegen, die Erdungsschwelle daran kalibrieren und Re-Ranking oder Anfrage-Verfeinerungen nur ergänzen, wo sie die Trefferquote nachweislich heben.

TEIL 17

Erste Schritte

1. **Einen Ordner wählen, den Sie gut kennen** — ein kleiner, vertrauter Rahmen, um Antworten an Ihrem Wissen zu prüfen.
2. **Zuerst die Bestandsaufnahme, nichts verändern** — die Form der Sammlung sichtbar werden lassen, bevor Sie handeln.
3. **Drei echte Fragen stellen** — zu sehen, wie es korrekt antwortet und ehrlich verweigert, schafft Vertrauen.
4. **Den Rahmen erst erweitern, wenn er verdient ist** — dann auf die Ordner richten, die Sie vergessen haben.

> *Der Bibliothekar hat jedes Dokument in Ihren Ordnern gelesen. Es bleibt nur, zu fragen.*

ANHANG

Glossar

LLM — die Komponente, die die Antwort in natürlicher Sprache schreibt; hier lokal. **RAG** (retrieval-augmented generation) — Antworten aus Passagen, die aus Ihren Dateien abgerufen wurden. **Einbettung** — ein Zahlen-Fingerabdruck der Bedeutung einer Passage. **Hybridsuche / RRF** — Bedeutungs- und Stichwortsuche, verschmolzen per Reciprocal Rank Fusion. **Re-Ranker** — ein zweiter Durchgang, der die besten Kandidaten nach echter Relevanz ordnet. **Grounding-Gate** — eine Qualitätsschwelle; darunter verweigert das System, statt zu raten. **OCR** — Text aus einem eingescannten Bild lesen. **Vektorspeicher** — eine Datenbank, gebaut, um die nächsten Vektoren zu einer Anfrage schnell zu liefern. **NAS** — ein Netzlaufwerk als nur lesende Dateiquelle. **Air-Gap** — ein Gerät ohne Netzweg nach aussen.

© 2026 SIM Research Institute AG · The Tidy Machine — ein Whitepaper. Architektur-Diagramme sind illustrativ; Prototyp-Screenshots stammen aus einem lauffähigen Aufbau auf einer Workstation. Souverän, lokal, geerdet — durch Konstruktion.