# DocGenerator

Sichere Suche & Berichtsentwürfe aus technischen Reports (Rapid Prototype)

# Ausgangslage

Problem

**Heute**

- Wissen steckt in PDFs und Ordnerstrukturen
- Suche kostet Zeit (Dateinamen, Stichworte, Lesen)
- Doppelarbeit bei ähnlichen Berichten
- Skalierung: mehr Reports ⇒ linear mehr Aufwand

## Pain

Zeit, Qualität, Wiederverwendung

# Zielbild

In Minuten statt Stunden

**Was wird besser?**

- Semantisch finden: auch ohne exakte Schlagworte
- Entwürfe aus Quellen: kurz, strukturiert, nachvollziehbar
- Immer mit Quellen/Seiten/Nachweis
- Angebotsprozess: Textbausteine/Checklisten (keine Kalkulation aus Reports)

**Leitsatz: KI liefert Vorschläge – Entscheidungen & Freigaben bleiben beim Menschen.**

# Was ist DocGenerator?

3 Kernfunktionen

### 1) Ingest

PDFs einlesen & strukturieren

Abschnitte + Metadaten

### 2) Search

**Semantische Suche**

**Treffer + Quellen**

### 3) Generate

**Berichtsentwurf**

**aus Top-Fundstellen**

# Architektur (management-tauglich)

lokal betreibbar • skalierbar

## Datenfluss


Erklärung: Wir machen Dokumenttext maschinen-suchbar und geben Treffer + Quellen zurück.

# Bedenken: Halluzinationen

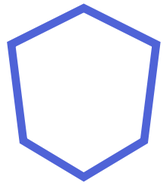
Wie wir Vertrauen schaffen

**Sorge**

- KI erfindet plausible Aussagen
- Unklare Nachvollziehbarkeit

**Design-Gegenmaßnahmen**

- Suche liefert Originaltext-Ausschnitte
- Generierung basiert auf Top-Fundstellen
- Quellenpflicht (Dokument/Seite)
- Output als Entwurf (Human-in-the-loop)



# Qualität & Nachvollziehbarkeit

messbar • prüfbar

**Qualitätssicherung**

- Nutzer prüft Treffer + Quellen vor Nutzung
- Entwurf ≠ Freigabe (Fachprüfung bleibt Pflicht)
- Erfolgsmessung im Feldversuch: Zeit, Trefferqualität, Fehlerklassen

# Betrieb & Datenschutz

lokal / im eigenen Rechenzentrum

**Kontrollierter Betrieb**

- Kein externer SaaS-Zwang: Betrieb im eigenen Netzwerk möglich
- Modelle lokal oder intern (DB) betreibbar
- Mit größeren internen Modellen steigt Qualität – Architektur bleibt gleich

# Rapid Prototype – aber produktionsnah

professioneller Unterbau

**Rapid Prototype (heute)**

- kleiner Scope
- schnelle Iteration
- messbarer Nutzen

**Produktionsnaher Unterbau**

- klare Module
- Docker-fähig
- erweiterbar (Rechte/Logging)

# Zeitersparnis

Business Case (messbar)

**Nutzen**

- Schneller finden: Sekunden statt Minuten
- Schneller starten: Entwurf aus Quellen statt Copy/Paste
- Messplan: Vorher/Nachher Zeit + Nutzerfeedback

# Roadmap

Phase 1: 4 Wochen • Testgruppe: 10 Nutzer

## Phasen + Gates

### Phase 1

Qualität prüfen
~400 Dateien
4 Wochen

**Gate: Freigabe**

### Phase 2

On-Prem Migration
(DB Hardware)

**Gate: KPI Review**

### Phase 3

Feldversuch
10 Nutzer

**Gate: Go/No-Go**

### Phase 4

Rollout
Nachbarabteilungen

# Gate 1: Entscheidung

Freigabe Phase 1

**Freigabe (risikoarm, messbar)**

- Umfang: ~2 Berichtsjahre (~400 Dateien)
- Ziel: Zeitersparnis + Qualitätskennzahlen + Fehlerklassen
- Ergebnis: Messbericht (Zeit/Qualität) + Empfehlung für Migration & Testgruppe

# Migration: Hostinger → DB Hardware

Einschätzung

**Durchführbarkeit: hoch**

- Container-/Service-Design ist On-Prem-tauglich
- Zeitaufwändig typischerweise: Security/Compliance, Netzwerk/TLS, Rechte, Modellbetrieb
- Empfehlung: IT/Security früh einbinden, klare Betriebsprozesse

# Feldversuch

kontrolliert • 10 Nutzer

**Pilot-Setup**

- 10 Nutzer, definierte Use-Cases (Suche, Entwurf, Quellenprüfung)
- Feedback-Schleife + KPI-Review
- Ergebnis: Entscheidung für Rollout & Prioritäten

# Funktionsausbau

nach Pilot priorisiert

**Beispiele**

- Konservativer Modus: nur zitieren + zusammenfassen
- Quellen klickbar, Export (PDF/Word)
- Rollen/Rechte: Suche vs. Generierung
- Feedbackbutton: hilfreich/falsch/Quelle fehlt

# Software & Lizenzen

Transparenz für den Einsatz im Unternehmen

**Stack (Rapid Prototype)**

- FastAPI/Uvicorn (API) – Open Source
- SQLite (DB) – Open Source / Public Domain
- Ollama (Model-Server) – Open Source (Betrieb intern)
- Modelle (Embeddings/LLM) – Lizenz pro Modell prüfen
- PDF-Tools (pdfplumber/pypdf/reportlab) – Open Source

**Hinweis: Vor Rollout werden Modell-Lizenzen und interne Standards formal geprüft.**