

# **AI Document Assistant**

# **Projektvorstellung für BITS**

Java Fullstack mit Spring Boot, Angular, OpenAI, [Render.com](http://render.com)

## **1. Einleitung: Warum dieses Projekt zur BITS passt**

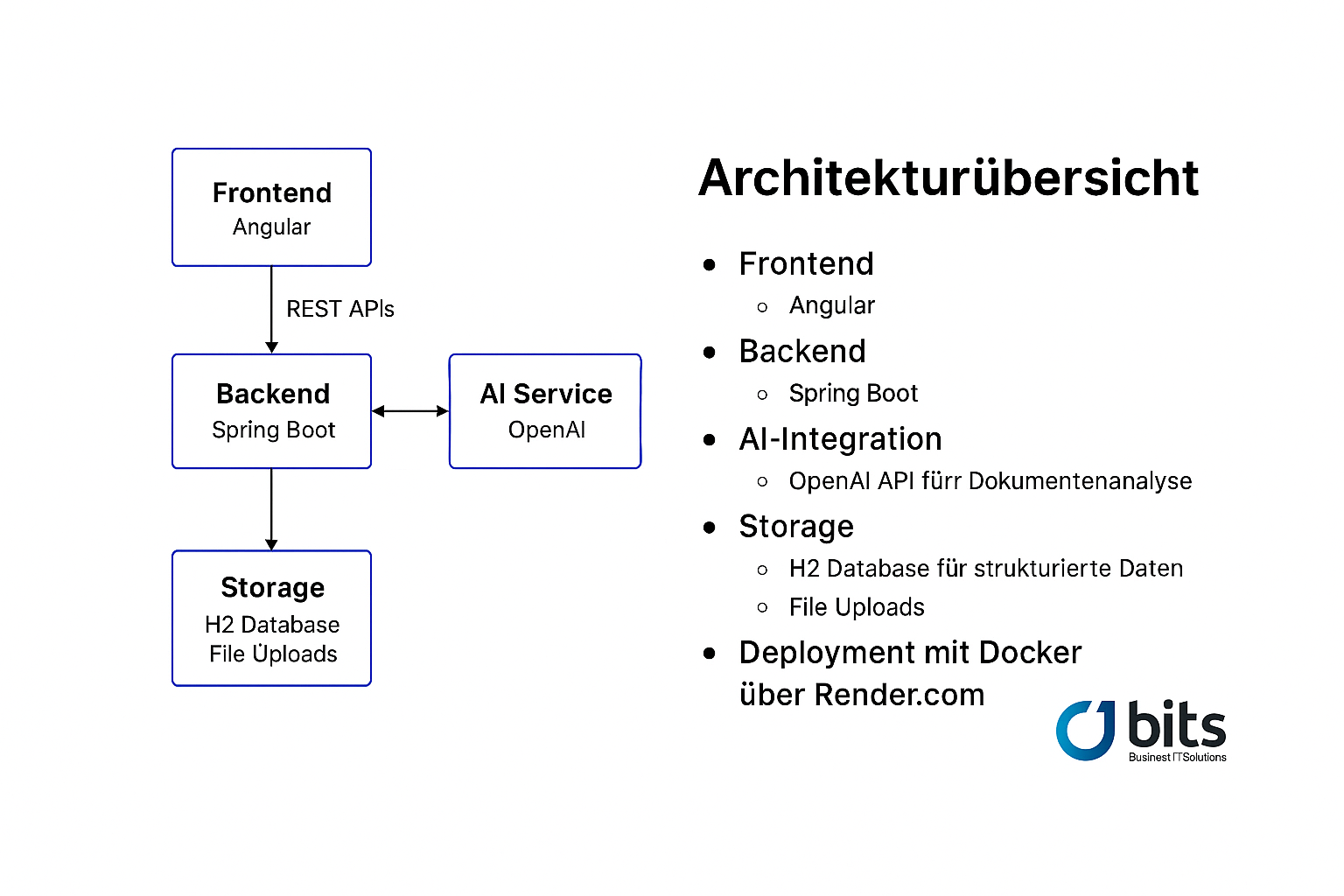
Deine Stelle: Full Stack Entwickler (Java/Angular), verantwortlich für Neu- und Weiterentwicklung digitaler Kundenlösungen  
Projektziel: Dokumente und Texte mit KI analysieren und strukturieren  
Relevanz für BITS:  
- Digital Solutions: KI für Datenmanagement, Webanwendungen  
- Consulting: KI-basierte Optimierung und Automatisierung  
- Kundenprojekte: Pharma, Automotive, Event, E-Commerce

## **2. Projektidee: "AI Document Assistant"**

Dokumente (PDF, DOCX, TXT) hochladen oder Text direkt eingeben  
KI analysiert Inhalt und erzeugt:  
- Zusammenfassung (GPT)  
- Schlüsselwörter  
- Technologie-Empfehlungen  
Benutzer gibt Feedback zur Analyse  
Prompt wird automatisch verbessert (Feedback Loop)

## **3. Architektur & Ablauf (Monolith)**

Beispiel: Nutzung von AI-Docs-Assist bei BITS  
  
Angenommen, ich arbeite im Team Digital Solutions bei BITS und bekomme von einem Kunden ein technisches Dokument – zum Beispiel eine Systemarchitektur-Beschreibung für ein neues Webportal.  
  
Das Dokument ist 25 Seiten lang und enthält Informationen zu Architektur, eingesetzten Frameworks, API-Endpunkten, Datenbankmodell und geplanten Integrationen.  
  
Normalerweise müsste ich viel Zeit investieren, um die wichtigsten Informationen herauszufiltern. Stattdessen lade ich das Dokument in AI-Docs-Assist hoch.  
  
Nach der Analyse bekomme ich:  
- Zusammenfassung: 3 prägnante Sätze mit den Kernpunkten der Architektur und Zielen  
- Schlüsselwörter: z. B. „Spring Boot, Angular, REST API, PostgreSQL, Docker, Keycloak“  
- Technologie-Empfehlungen: Ergänzende Vorschläge wie „CI/CD mit Jenkins, Monitoring mit Grafana“  
  
Mit diesen Ergebnissen habe ich in wenigen Minuten einen Überblick und kann gezielt mit den Kollegen oder dem Kunden in die Diskussion gehen.



**Frontend** (Angular):  
- Datei-Upload oder Texteingabe  
- Anzeige der Analyse-Ergebnisse  
- Feedback (Quick & Detail)  
  
**Backend** (Spring Boot):  
- DocumentController: Upload & Routing  
- AiService: Aufruf OpenAI + Fallback  
- PromptOptimizer: Optimierung bei negativem Feedback  
- FeedbackService: Speicherung & Auswertung  
- DocumentService: Speicherung & Abruf  
  
Deployment:  
- Frontend: <https://bits-ai-docs-assist-demo.onrender.com>  
- Backend API: <https://bits-ai-docs-assist.onrender.com>

## **4. Vision: Microservice-Erweiterung**

Document-Service (Upload + Storage)  
AI-Service (Analyse + Prompt-Tuning)  
Feedback-Service (Bewertung, Statistik)  
Prompt Optimizer (Qualitätslogik)  
API Gateway (Routing, Security)  
Service Discovery (Eureka / Kubernetes)

## **5. Markt & Potenzial**

Trends:  
- Microsoft Copilot, Google Duet AI  
- Wachsende Nachfrage nach Custom AI  
  
Stärken:  
- Schlanker Aufbau, erweiterbar  
- Feedback-Modul für Lerneffekt  
- Vollständiger Tech-Stack: Dev2Prod  
  
Verbesserungspotenzial:  
- Mehr AI-Provider  
- Authentifizierung  
- UI/UX Optimierung

## **6. Fazit**

Du bekommst ein fertiges, ausbaufähiges Projekt  
Ich bringe Fullstack + Architektur + AI-Erfahrung  
Direkt einsetzbar für Kundenprojekte oder interne Tools  
Genau die Schnittmenge aus Java, Angular, AI, Architektur und Cloud