

# Wochenend-Challenge: AI-Powered Real Estate Assistant

## 1. Der Kontext

Wir geben dir eine zweite Chance, dein technisches Verständnis unter Beweis zu stellen. Da du dieses Mal mehr Zeit hast (Wochenende), haben wir das Anforderungsniveau deutlich angehoben.

Wir suchen niemanden, der nur Tutorials nachbaut, sondern einen Entwickler, der **moderne AI-Architektur** (Agentic Workflows) mit **klassischer Softwareentwicklung** verbinden kann.

Deine Aufgabe ist es, einen Prototypen für einen KI-gestützten Immobilien-Assistenten zu bauen, der Maklern die tägliche Verwaltungsarbeit abnimmt.

## 2. Der Tech-Stack (Verbindlich)

Du musst zwingend folgende Komponenten in einer Microservice-Architektur orchestrieren:

1. **Frontend & User Interface:** Django (Python)
  - Dient als Dashboard und Chat-Eingabe für den Makler.
2. **Orchestrierung & Intelligenz:** n8n
  - Hier läuft das **Multi-Agenten-System**.
  - Der Agent analysiert den Intent des Users und entscheidet, welche Tools ausgeführt werden müssen.
3. **Tooling & Backend Logic:** MCP Server (Model Context Protocol)
  - Implementiert mit fastmcp (Python).
  - Stellt die Verbindung zur Datenbank her und definiert die ausführbaren Funktionen (Tools).
4. **Datenbank:**
  - MySQL oder PostgreSQL.

## 3. Die Architektur-Vorgabe

Wir erwarten "Separation of Concerns". Baue keinen Monolithen.

- **Django** darf **nicht** direkt Business-Logik auf der Datenbank ausführen (außer Read-Views für das Dashboard). Schreiboperationen müssen über den **Agenten** laufen.
- **n8n** fungiert als "Gehirn". Es empfängt den Befehl von Django, plant die Schritte und nutzt den **MCP Server** als "Hände", um die Arbeit zu erledigen.
- **MCP Server** kapselt die Datenbank-Logik.

## 4. Funktionale Anforderungen

Das System muss folgende Entitäten verwalten und über natürliche Sprache (Chat) steuern können:

### A. Kundenmanagement (CRM)

- **Unterscheidung:** Eigentümer (Owner) vs. Suchkunden (Buyer).
- **Daten:** Name, Kontakt.
  - *Spezifisch für Suchkunden:* Budget & Suchregion.
- **Use Case:** *"Leg einen neuen Suchkunden Max Müller an, er sucht in Frankfurt bis 1 Mio. Euro."*

### B. Immobilienmanagement

- **Daten:** Adresse, Preis, Typ (Haus/Wohnung).
- **Use Case:** *"Das Objekt in der Hauptstraße 5 kostet jetzt 450.000 Euro."*

### C. Terminplanung

- Verknüpfung von Kunde + Makler + (optional) Objekt.
- **Use Case:** *"Erstelle einen Besichtigungstermin mit Herrn Müller für das Objekt Hauptstraße am Dienstag um 14 Uhr."*

### D. Interaktions-Historie & Notizen

- Makler müssen Notizen zu Kunden hinterlegen können.
- **Zwingend mit Typisierung:** Gespräch, Telefonat oder E-Mail.
- **Use Case:** *"Notiere, dass ich Herrn Müller angerufen habe (Typ Telefonat). Er hat Interesse signalisiert."*

## 5. Datenbankvorgabe (Schema-Definition)

Das Datenmodell muss mindestens folgende Entitäten und Attribute abbilden. Du hast die Freiheit, Datentypen sinnvoll zu wählen.

### A. Properties (Immobilien)

Feld	Beschreibung	Wertebereich / Beispiel
<b>Straße</b>	Straßenname	String
<b>Hausnummer</b>	Hausnummer (inkl. Zusatz)	String (z.B. "7a")
<b>PLZ</b>	Postleitzahl	String (5-stellig)
<b>Ort</b>	Stadtname	String
<b>Land</b>	Ländercode oder Name	String
<b>Baujahr</b>	Erstellungsjahr	Integer
<b>Zustand</b>	Bewertung des Objekts	simpel, normal, gehoben (Enum)
<b>Zimmer</b>	Anzahl der Zimmer	Integer / Float (z.B. 3.5)
<b>Wohnfläche</b>	Fläche in m <sup>2</sup>	Float
<b>Art</b>	Objekttyp	Wohnung, Haus, MFH (Enum)

### B. Buyer (Suchkunden)

Die Suchkriterien können als JSON-Feld oder als separate Relationstabelle modelliert werden – deine Wahl.

Feld	Beschreibung	Hinweise
<b>Vorname</b>	Vorname des Kunden	
<b>Nachname</b>	Nachname des Kunden	
<b>Budget</b>	Maximales Budget	Decimal / Integer

<b>Suchkriterien</b>	<b>Strukturierte Anforderungen</b>	Sollte folgende Parameter enthalten: • <b>Art:</b> (siehe Properties) • <b>Zimmer:</b> (Min / Max Range) • <b>Wohnfläche:</b> (Min / Max Range) • <b>Zustand:</b> (simpel/normal/gehoben)
----------------------	------------------------------------	---

## C. Owner (Eigentümer)

Feld	Beschreibung
<b>Vorname</b>	Vorname des Eigentümers
<b>Nachname</b>	Nachname des Eigentümers
<b>Geburtsdatum</b>	Zur Identifikation
<b>E-Mail</b>	Kontaktmail
<b>Telefonnummer</b>	Kontaktnummer

## 6. OpenAI API Key

Dir steht der folgende API Key zur Verfügung (innerhalb von 24 Stunden abrufen, der Link ist nur einmalig aufrufbar):

**[REDACTED]**

**[REDACTED]** Achte bitte auf das Budget (10\$), ist dieses Budget ausgereizt, so kannst du nicht mehr auf die API zugreifen.

## 7. Abgabe

Bitte sende uns nach der Präsentation:

1. Link zum **GitHub Repository**.
  - N8n Flow
  - Django Frontend
  - MCP Server
2. Dockerfiles und Readme als Anleitung zum Aufsetzen auf einem lokalen Rechner.

Viel Erfolg!