PRAXIS-SPICKZETTEL · DIN A4 ZUM AUSDRUCKEN

Das KI-Cockpit – Kompakt-Referenz

Prozesse, Zahlen und Abhängigkeiten im Blick: Souveräne Dashboards bauen mit lokaler KI — ohne Vorkenntnisse und Lizenzkosten

Michael Estel · Begleit-Monorepo mit ZIP-Download und Live-Demos: https://on-premise-ai-cockpit.github.io

Live-Demos & ZIP

DIE 5 AUSBAUSTUFEN

Stufe 2–5 erweitern dieselbe cockpit.html

Stufe 1 — **Diagramme aus Text (Kap. 2)** keine Installation
Mermaid, BPMN 2.0 und C4 aus Text statt veralteter Word-Bilder („Doc-Rot").

Stufe 2 — **Das erste Cockpit (Kap. 3)** Doppelklick
Eine lokale `cockpit.html` mit KPI-Kacheln, Chart.js und Prozess-Wegweiser.

Stufe 3 — **+ Dropzone & Prozess-Editor (Kap. 4)** Doppelklick
DATEV-/Excel-Exporte (`.csv/.xlsx`) lokal laden; dazu `prozess_editor.html` .

Stufe 4 — **+ Datenbank & Wissensgraph (Kap. 5)** Mini-Server (Python über IT)
`sql.js` + `Vis.js` : Einzelbuchungen, 500 Kontakte, Blast Radius je Funktion.

Stufe 5 — **+ KI-Funktionen (Kap. 6)** + Ollama (IT-Freigabe)
PDF-Extraktor, Posteingangs-Vorsortierung und Gremien-Orakel mit lokaler KI.

DIE 4 KERNPRINZIPIEN

Meta-Prompting
Erst den Bauplan-Prompt von der KI entwerfen lassen, dann Code erzeugen.

Ping-Pong-Prinzip
Klein starten, testen (`F12`), Fehlermeldung zurück an die KI.

Tankdeckel-Prinzip
Die Cockpit-Datei enthält keine produktiven Echtdaten; echte Daten laden Sie zur Laufzeit.

Kellner-Prinzip
Das Cockpit fragt die lokale KI per `fetch` an; der Mensch prüft jedes Ergebnis.

HARDWARE-KOMPASS (ABSCHNITT 1.5)

Dienstrechner	Lokale Modelle (Beispiele)
8 GB RAM	`qwen3.5:4b` · `llama3.2:3b` (kurze Aufgaben)
16 GB RAM	`qwen3.5:9b` · `gemma4:12b` (Allrounder, JSON)
32 GB RAM	`qwen3.5:27b` · `gemma4:26b` (lange Dokumente)

GOVERNANCE STATT SCHATTEN-IT: DIE 6 LEITPLANKEN

Übergabe an IT, ISB und Personal- bzw. Betriebsrat (Abschnitt 8.5)

1 Datenhaltung (Tankdeckel)
Die Cockpit-Datei enthält keine produktiven Echtdaten. Mitgelieferte Beispieldaten sind fiktiv; echte Exporte laden Sie zur Laufzeit.

2 Offline über /libs
Alle Bibliotheken liegen lokal in `/libs/` . Prüftest: `F12 → Netzwerk` zeigt keine externen Adressen.

3 Lokale KI abgesichert
`OLLAMA_HOST` bleibt `127.0.0.1` ; nie `OLLAMA_ORIGINS="*"` (erweitert nur); Cloud-Modelle aus; Zündschlüssel-Regel.

4 Rollen & Versionierung
ISO-Datum im Namen (`JJJJ-MM-TT_cockpit_v1.2.html`), `/archiv/` für Rollback, zwei Pflege-Verantwortliche.

5 Lizenzen & Nachweise
Stückliste (SBOM) in `/libs/README_LIBS.md` (MIT, Apache-2.0, `bpmn.io` -Logo sichtbar) plus einseitige Architekturskizze.

6 Mitbestimmung & Protokolle
Cockpit schreibt keine Nutzungsprotokolle; Logs von Ollama/Mini-Server regelt die IT. Nicht für Leistungskontrolle ausgelegt, Kritikalität je Funktion statt Person.

VIER MASTER-PROMPTS UND DIE SICHEREN STARTBEFEHLE

Kopiervorlagen auf on-premise-ai-cockpit.github.io

1. Meta-Prompting — Einleitung

```
Ich möchte [Ziel beschreiben, z. B. ein lokales HTML-Dashboard für vier Kostenstellen]. Entwerfen Sie mir dafür einen strukturierten Prompt für eine Programmier-KI, ohne externe Server-Abhängigkeiten.
```

2. Lokale Pseudonymisierung — Abschnitt 1.4

```
Ersetzen Sie alle Personennamen ([NAME_1]), Firmen ([FIRMA_A]), Orte und Aktenzeichen konsistent. Dieselbe Entität erhält stets denselben Platzhalter. Erhalten Sie Satzbau und Fachkontext.
```

3. JSON-Strukturzwang — Abschnitt 6.1

```
Antworten Sie AUSSCHLIESSLICH mit gültigem JSON ohne Einleitung oder Markdown, mit diesen Schlüsseln:
{"rechnungsnr": "", "lieferant": "",
  "kostenstelle": "", "netto_eur": 0}
```

4. Ausbau-Regel — Stufe 3 bis 5

```
Fügen Sie Ihre bisherige cockpit.html vollständig ein und lassen Sie alle vorhandenen Reiter unverändert. Ergänzen Sie nur den neuen Reiter [Name].
```

Sichere Startbefehle — ab Stufe 4 / Stufe 5 (Abschnitt 6.2)

```
# Mini-Server im Cockpit-Ordner (Python stellt meist die IT bereit; macOS: python3):
python -m http.server 8000 --bind 127.0.0.1
# Ollama: localhost ist ab Werk erlaubt – KEIN OLLAMA_ORIGINS nötig. Cloud-Modelle abschalten:
$env:OLLAMA_NO_CLOUD="1"; ollama serve
# llama-server statt Ollama: nur lokal und mit Schlüssel
llama-server -m modell.gguf --host 127.0.0.1 --api-key-file schluessel.txt
```

Zeitungstest (Abschnitt 1.1) & Hybrid-Strategie (Abschnitt 1.6): Wäre es problematisch, wenn die Information morgen in der Zeitung stünde? Dann Zone Rot: nur lokal arbeiten. Zone Grün (leere Spaltenköpfe, fiktive Beispieldaten, Code): nur eine von Ihrer Organisation freigegebene Cloud-KI mit abgeschaltetem Training – oder lokale KI. Pseudonymisierte Echtdaten nur mit Freigabe des Datenschutzes in eine Cloud.

Frei zum Ausdrucken und für interne Workshops · **on-premise-ai-cockpit.github.io**