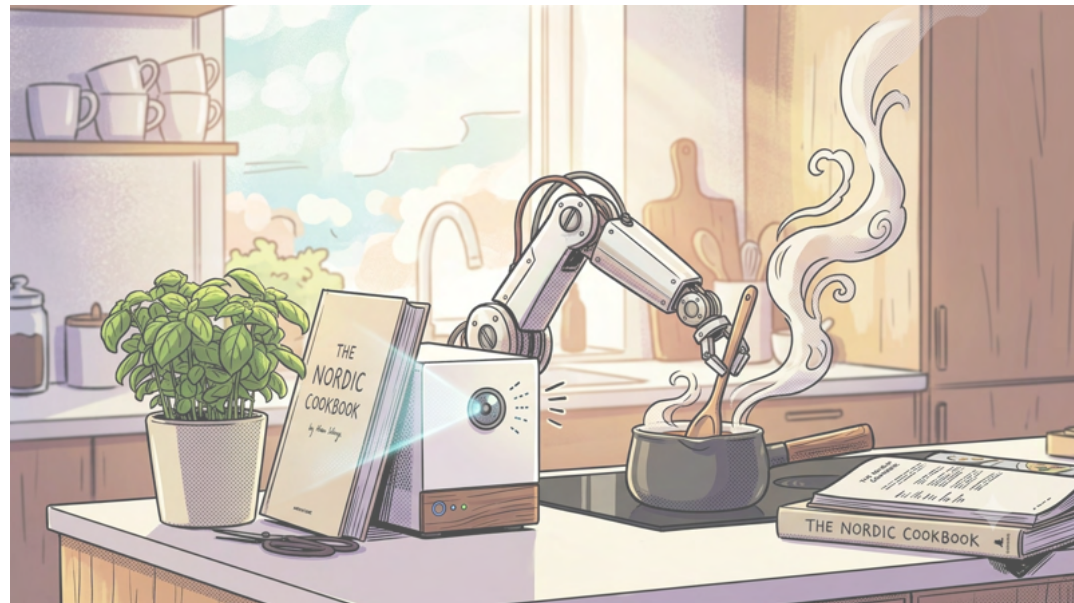


Smarte Rezept-Automatation

mit Mealie, Docker & n8n

... oder warum die KI im Zweifelsfall Zwiebelsuppe kocht.



MakerTalk @ FabLab Winti
Tobias Kammacher, 10.7.2026

Agenda für heute

1. Kontext

Rezept-Chaos & der "Tool-Belt"

2. Die KI

Parsing, Prompts & Probleme

3. Workflow

Scan per Handy (OCR-Pipeline), Automation mit n8n

4. Demo

Live-Action & Ausblick



Das Rezept-Chaos im Alltag

- **Die Realität:**

- Lose Zettel im Küchenschrank
- Screenshots auf dem Smartphone (die man nie wieder findet)
- 25 Browser-Tabs mit Rezepten
- Ein Gestell voller Kochbücher mit Post-its

- **Die Hürde:** *"Keine Lust, stundenlang Zutaten und Schritte abzutippen."*

- **Die Vision:** Eine zentrale, durchsuchbare Rezept-Datenbank für die ganze Familie. Mit Meal Planning und Einkaufsliste.



Die Lösung: Mealie

Die Werkzeuge

-  **Docker:** Portables Format für Server-Applikationen
 -  **Mealie:** Rezept-Datenbank.
 -  **Paperless-ngx:** Dokumenten-Archiv für PDFs/Bilder und OCR.
 -  **n8n:** Verbindet alles visuell per Drag-and-Drop.
 -  **Portainer:** Weboberfläche für einfaches Container-Management.)
-  **KI (LLM):** Das smarte Bindeglied für unstrukturierte Daten.
 - *In der Cloud:* ZB. Google Gemini 2.5 Flash (schnell, gratis, teils unverfügbar)
 - *Open-Source-Ausblick für zu Hause:* Ein lokales 8B-Modell (z. B. Llama 3.1 8B via Ollama) – benötigt GPU mit 6GB+ VRAM.

Die Infrastruktur (Docker)

Docker ist "virtuelle Werkstatt": Jedes Tool läuft isoliert in seiner Box (Container).

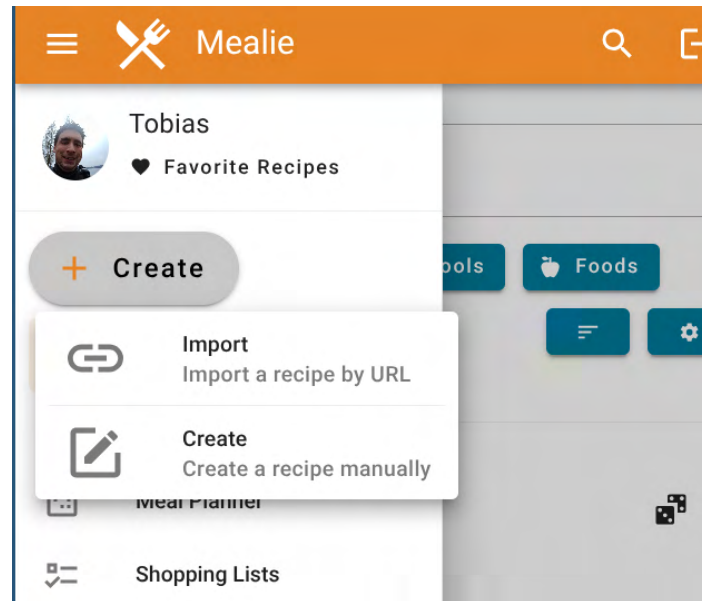
Ein Auszug aus `compose.yml`: (Konfiguration)

```
services:
  mealie:                # Das Kochbuch
    image: ghcr.io/mealie-recipes/mealie:latest
    ports: ["9925:80"]    # Erreichbar im Browser unter Port 9925
  n8n:                   # Der API-Dirigent
    image: docker.n8n.io/n8nio/n8n:latest
  ...
```

- Die Container kommunizieren isoliert über das interne Docker-Bridge-Netzwerk.
- Ermöglicht eine einfache, reproduzierbare Konfiguration und die volle Kontrolle über die installierten Container-Versionen (zB. `:1.4.2` anstatt `:latest`).

Mealie Rezept Import (Idealfall)

- Wie kommen Rezepte in die Mealie Datenbank?
 - **Import:** Rezept "Scraper", liest automatisch eine Website aus
 - **Create:** Manuelle Eingabe



Mealie Rezept Import (Idealfall)

- **Wie Mealie Webseiten liest (Rezept Scraper):** Viele moderne Koch-Blogs liefern standardisierte Metadaten für Google & Co.
 - Mealie nutzt ein Format namens **JSON-LD** (JavaScript Object Notation for Linked Data).
- **Das erwartete Schema (Beispiel):**

```
{
  "@context": "https://schema.org",
  "@type": "Recipe",
  "name": "Safran-Tagliatelle",
  "recipeIngredient": ["250g Tagliatelle", "0.1g Safran"],
  "recipeInstructions": [{"@type": "HowToStep", "text": "Pasta kochen und Safran einrühren."}]
}
```

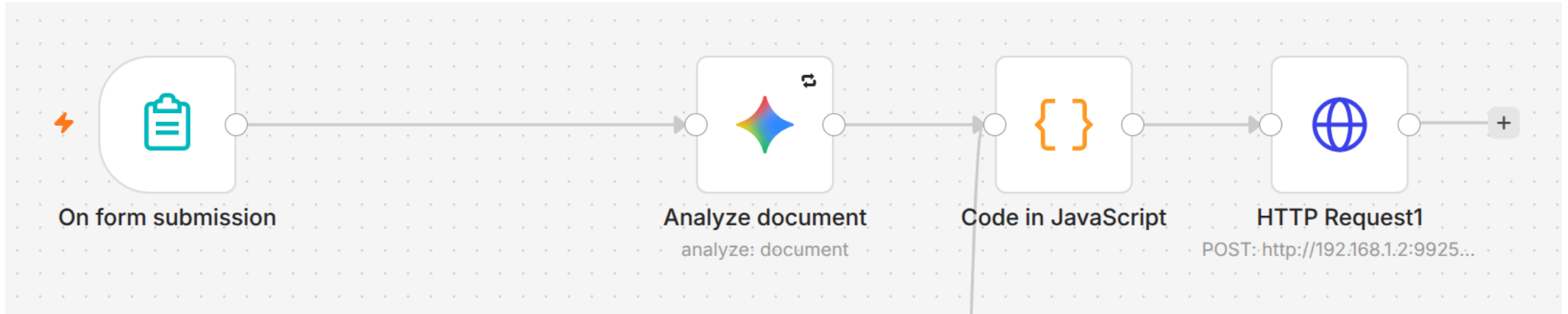
- Somit kann Mealie das Rezept in Sekunden perfekt einlesen

Die Hürde: Wenige Rezepte im JSON-LD Format

- Viele private Blogs, ältere oder absichtlich geschützte Webseiten bieten keine strukturierten Metadaten – der native Scraper sieht nichts.
- **Klassische Programmierung (Regex / CSS-Selektoren):**
 - Suche nach `<span class="ingredient">`
 - Unterschiedlich für jede Seite. Scheitert schon wenn Website-Design ändert.
- **Der KI-Ansatz (Semantische Übersetzung):**
 - Ein Large Language Model (LLM) liest HTML-Code wie ein Mensch.
 - Es versteht Kontext: `1 Zwiebel`, `one onion`, oder `Zwiebel (feingehackt): 1 Stk.` – für die KI ist es dasselbe Konzept.
 - Das Ziel ist Übersetzung von chaotischem Input in ein standardisiertes Schema (JSON-LD).

Exkurs: Automatisierung mit n8n

- Workflow um Rezepte nicht-unterstützter Websites trotzdem zu importieren:



1. Erzeuge ein Formular um die Rezept-URL entgegen zu nehmen.
2. n8n lädt den rohen HTML-Code der Website herunter.
3. Die KI liest den HTML-Salat und extrahiert Name, Zutaten und Schritte und wandelt die Daten ins JSON-LD Schema um.
4. Die Daten landen vollautomatisch in Mealie (Webhook).

Das Gehirn konfigurieren (AI Prompt)

- Wir füttern die KI nicht mit Fragen, sondern mit einem präzisen "Arbeitsauftrag" (System Prompt) als Rezept Parser.

```
"You are an expert culinary data extractor. Analyze the following web page text and extract the recipe. You MUST return ONLY a clean JSON object fitting the Mealie recipe schema. Do not write conversational text or code blocks."
```

Here is an example of the mealie target JSON structure:

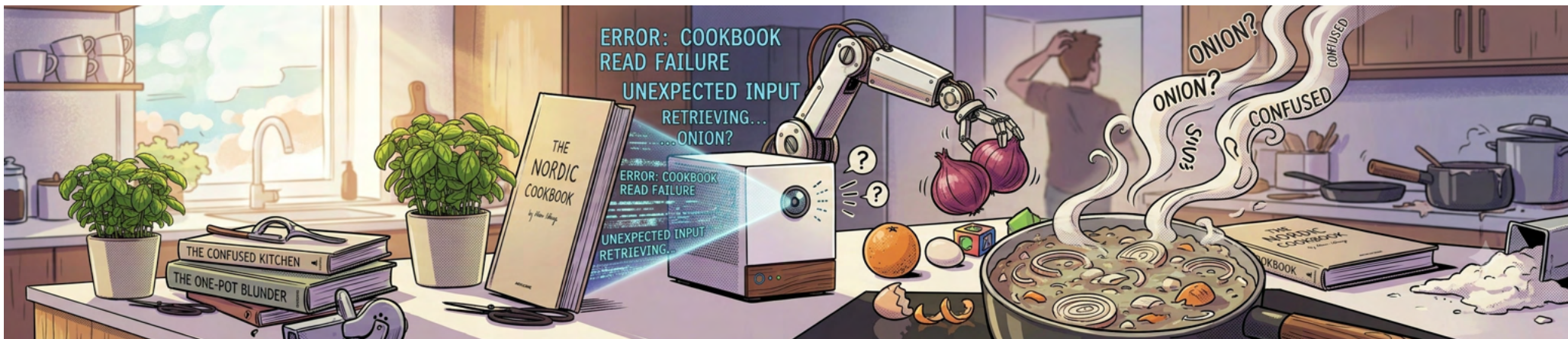
```
{
  "name": "Recipe Title",
  "description": "Short description",
  "recipeYield": "4 servings",
  "recipeIngredient": [
    {"note": "1 tbsp olive oil"},
    {"note": "2 cloves garlic, minced"}
  ],
  "recipeInstructions": [
    {"text": "Heat oil in a pan over medium heat."},
    {"text": "Add garlic and sauté until fragrant."}
  ]
}
```

Was passiert, wenn der Workflow einen Bug hat?

- **Der Fehler:** n8n übergab Gemini eine leere Variable (`undefined`) statt des Rezept-Texts.
- **Was passiert bei klassischer Programmierung?**

Was passiert, wenn der Workflow einen Bug hat?

- **Der Fehler:** n8n übergab Gemini eine leere Variable (`undefined`) statt des Rezept-Texts.
- **Die Reaktion der KI:** Statt eine Fehlermeldung auszugeben, wollte Gemini unbedingt den System-Prompt erfüllen ("Liefere ein gültiges Rezept-JSON").
- **Die Halluzination:** Aus dem Nichts erfand die KI ein makelloses Rezept für **französische Zwiebelsuppe** (inkl. Gruyère-Toast), da dies statistisch ein sehr "typisches" Rezept in den Trainingsdaten war!



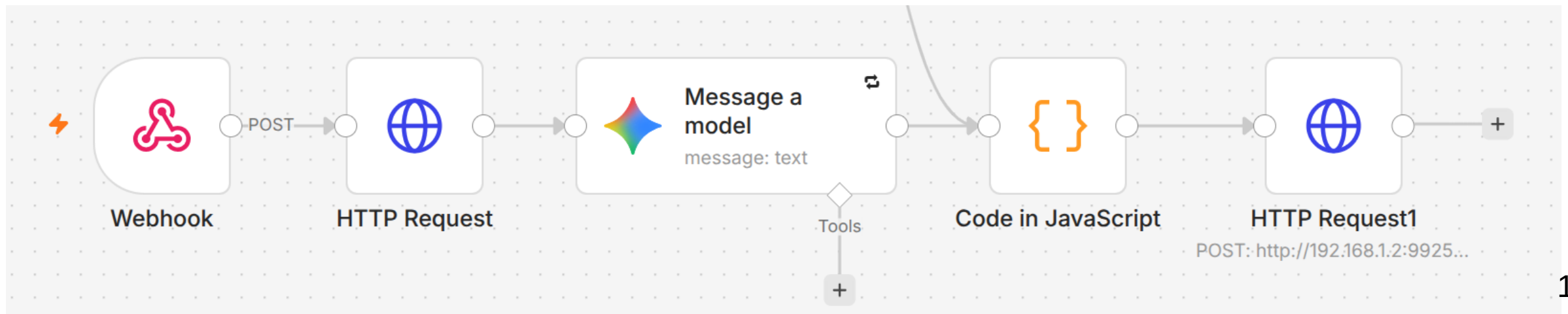
Prompt-Engineering

- KI Prompt ist eine schwammige Lösung für einen schwammigen Input.
- Wie wir die KI zähmen und Halluzinationen verhindern:
 1. **Strict Negative Constraints:** *"NO HALLUCINATIONS: You must extract ONLY the recipe described in the OCR text. Never substitute or invent a different recipe."*
 2. **Sprach-Verankerung (Language Locking):** *"Keep all recipe content in the ORIGINAL language (German). Do not translate."*
 3. **Umgang mit fehlerhaftem OCR:** *"Correct obvious scanning typos: '1Y2dl' means '1.5 dl', 'r Zwiebel' means '1/2 Zwiebel'."*

Rezept Scan per Handy

Wie wir gedruckte Kochbücher ohne Abtippen importieren:

1. Kochbuch aufschlagen, Foto mit der "Paperless Mobile" App schießen.
2. Paperless-ngx liest das Bild ein und macht den Text durchsuchbar (OCR).
3. Wir weisen dem Dokument den Tag `mealie` zu.
4. Paperless weiss nun dass das Dokumente ein Rezept ist und sendet es an n8n.
5. Der n8n Workflow (unten) macht den Text Mealie-tauglich mit AI.



AI Prompt für Umwandlung von OCR

You are a strict, literal OCR-to-Recipe data extraction engine. Your sole task is to extract the recipe from the provided messy OCR text and format it as a valid Schema.org/Recipe JSON object.

CRITICAL RULES:

NO HALLUCINATIONS: You must extract ONLY the recipe described in the OCR text below. Never substitute or invent a different recipe.

PRESERVE LANGUAGE: Keep all recipe content (name, description, ingredients, instructions) in the ORIGINAL language of the OCR text

(German or English). Do not translate German to English.

OCR CLEANUP: Correct obvious scanning typos sensibly:

"aaaen|" is noise, ignore it.

"1Y2dl" or "112 dl" means "1.5 dl" or "1 1/2 dl" (decide based on context).

"r Zwiebel" means "1 Zwiebel" or "1/2 Zwiebel".

Do NOT add ingredients (like beef broth, baguettes, cheese) that are not mentioned in the source OCR text.

FORMAT: Return ONLY the raw JSON object. Do not wrap it in markdown code blocks (no ```json). Do not add any conversational text.

SOURCE OCR TEXT:

```
{{ $json.content }}
```

Der Paperless → n8n Pipeline-Bau (Details..)

Die technischen Hürden beim Verbinden zweier lokaler Container:

- **Das Jinja2-Format in Paperless:** Wir nutzen den Parameter `{{doc_url}}` im Webhook-Body, um n8n mitzuteilen, welches Dokument verarbeitet werden soll.
- **Die ID-Extraktion mit Regex in n8n:**
Wir extrahieren die Dokumenten-ID elegant im URL-Feld des HTTP-Nodes:

```
http://192.168.1.2:8010/api/documents/{{  
$json.body.document_url.match(/documents\/(\d+)/)[1]  }}/
```
- **Das Token-Geheimnis:** Authentifizierung via API-Token im HTTP-Header (
`Authorization: Token <key>`).

Live-Demo

1. **Demo 1:** Import einer Website in Mealie mit JSON-LD.
2. **Demo 2:** Import einer Website über das n8n-Formular.
3. **Demo 3:** Rezept mit "Paperless Mobile" App scannen; automatische Auslösung des Workflows.

Fazit & Ausblick

- **Was ich gelernt habe:**

- Docker Container als stabile Grundlage für vielfältige Open-Source Projekte.
- n8n ist ein Allzweck-Tool für API-Integrationen.
- KI (LLMs) eignet sich als Parser für unstrukturierte Daten; mit Vorsicht..

- **Wie weiter?**

- **Rezept Austausch:** Bei Interesse könnte ich Zugang ermöglichen.
- **100% lokal:** Cloud-KI durch lokale Modelle ersetzen (Ollama auf Homeserver).
- **Rezept Foto:** Automatisiertes scrapen des Fotos direkt ins Mealie Rezept.
- **Anleitung** auf <https://github.com/kr4ch/smarte-rezept-datenbank>

Fragen? Diskussion?

