



KI-Community Praxisworkshop, Vortrag, Online-Webinar

Fokus: KI-Landkarte, Agenten, Workflows, Entwicklungsumgebungen für KI-“Coder”

Link: <https://github.com/rakidakiXYZ/101-Coherent-HPI-KI>



Autor: Ralf Kluth

Web: <https://from-scratch.ai> | X: @fromscratchai | LinkedIn: <https://www.linkedin.com/in/rakidakixyz/>



Agenda

- 01 KI + ADG-Community**
- 02 KI-Landkarte und News**
- 03 Überblick Agenten, Workflows, ...**
- 04 100 Zeilen Python Code – Basis für KI-Agenten**
- 05 Cursor/ Claude Code und 100 Zeilen Python**
- 06 KI-Entwicklung: Manus, Lovable, Dyad, Kimi, CC**
- 07 Claude Code Setup (u.a. BMAD)**

1

ADG & Atruvia KI-Community

Kurse, Best-Practices, Prompts, Vorlagen, News, Webinare, ...

ADG & Atruvia KI-Community

<https://genobank.ai>



ADG ATRUVIA

Genobank.AI

KI-Community

Herzlich Willkommen in der KI-Community "Genobank.AI", powered by ADG & Atruvia

Austausch, Weiterbildungen und Best-Practices rund um KI.

Hier findest du alles, was dich im Bereich KI weiterbringt: Austausch, Vernetzung & kuratierte Bildungsangebote.

E-Mail [Passwort vergessen?](#)

Dieses Feld darf nicht leer sein

Passwort [Passwort](#)

Erinnere dich an mich

[Einloggen](#)

Neu auf unserer Plattform? Ein Konto erstellen

[Impressum](#) [Datenschutz](#) [AGB](#)

Start Dezember 2025 / Januar 2025



<https://genobank.ai>

- Kurse
- Praxisbeispiele GFG
- Prompt-Beispiele
- KI-Assistenten
- Monatliche Webinare
- Einblicke in Projekte
- Sprechstunden ...

ADG & Atruvia KI-Community

<https://genobank.ai>

The screenshot displays the Genobank.AI KI-Community platform. On the left, a sidebar menu includes 'MEINE KURSE' (selected), 'COMMUNITY', 'Kalender', 'LINKS', 'Bildungsangebote', and 'BVR KI-Kompetenz...'. The main content area features a welcome message: 'Herzlich Willkommen in der KI-Community "Genobank.AI" by ADG & Atruvia. Austausch, Bildung und Best-Practice zu KI.' Below this is a section titled 'MEINE KURSE' with two course cards: 'Zertifizierter KI-Compliance Experte ADG (KI-Beauftragter) - Certified by Atruvia' (17 Modules, 41 Lessons) and 'KI-Selbstcheck - Wie fit bist du im Thema KI?' (1 Modul, 4 Lektionen). A central callout box says 'Schön, dass du Teil unserer Community bist. Entdecke deine Inhalte und werde aktiv – viel Spaß!' To the right, a large orange box highlights 'KOSTENLOSE KURSE' with five course cards: 'Prompts, Prompt-Generatoren, Prompt-Frameworks für Texte, Bilder und Videos | Praxisbeispiele GFG' (Kursdetails), 'Anthropic Claude - Alternative Lösung zu OpenAI ChatGPT' (Kursdetails), 'OpenAI ChatGPT' (Kursdetails), 'Bilder mit KI generieren' (Kursdetails), and 'Videos mit KI generieren' (Kursdetails). An orange callout box on the right side lists '• Monatliche Updates' and '• Wöchentliche Deep-Dives'.

- Monatliche Updates
- Wöchentliche Deep-Dives

2

KI-News und KI-Landkarte (Produkte, Lösungen, LLMs)

Neuigkeiten, OpenAI – Anthropic Claude – Google Gemini => Neue Funktionen

Landkarte der KI-Lösungen

Open-Source Community und Lösungen

KI-News

Rechenzentren, Investitionen, Firmenbewertungen



Bau von neuen gigantischen Rechenzentren

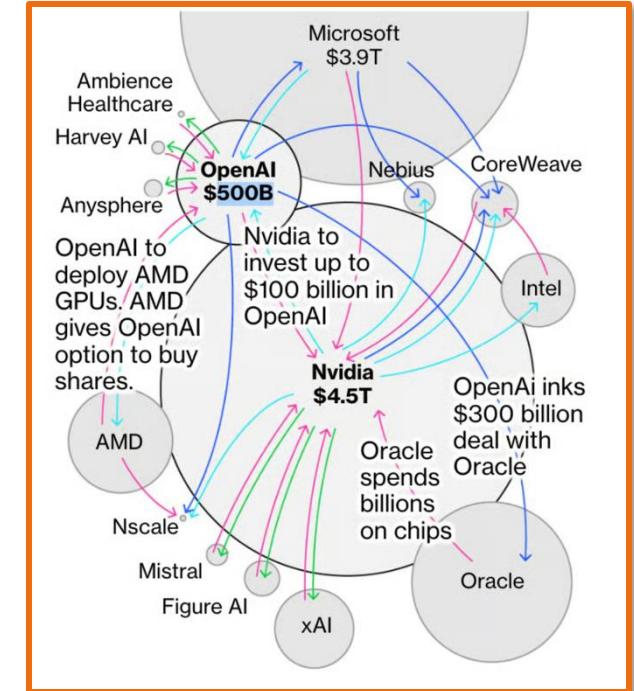
Anthropic verdreifacht seine Bewertung auf 183 Milliarden Dollar

Damit wird der OpenAI-Konkurrent zu einem der höchstbewerteten Start-ups der Welt. Der KI-Entwickler will nun international expandieren und vermehrt zu Sicherheit forschen.

Überbewertung von KI-Firmen?



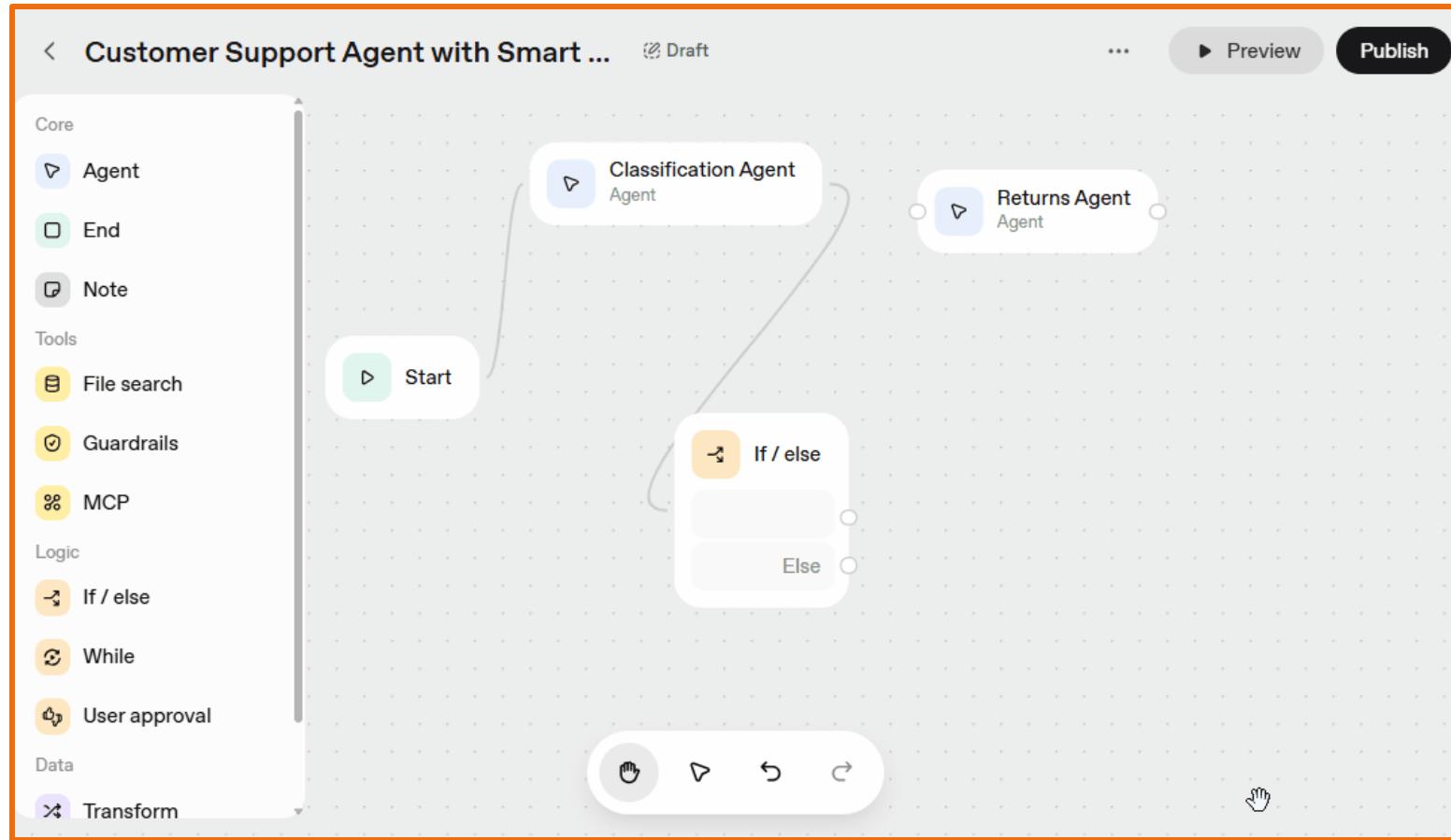
**KI verändert die Zukunft
Ängste vs. Visionen**



Geldflüsse in der KI-Bubble

KI-News

OpenAI, Anthropic, Google, ...



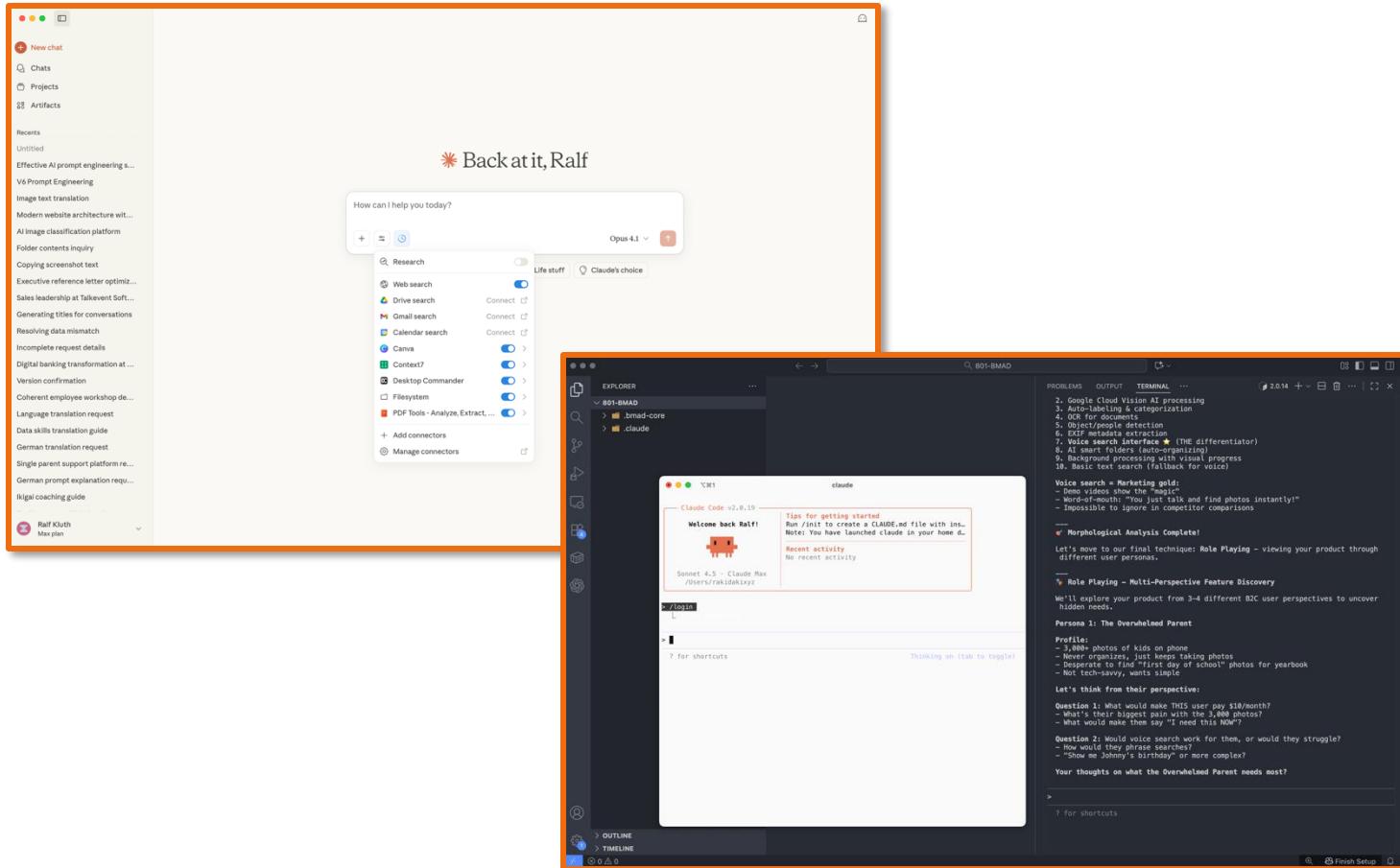
OpenAI:

- Tools und Konnektoren
- MCP
- Apps und App Store
- Agent Builder
- ChatKit
- Guardrails
- AgentsSDK

OpenAI => komplettes Ökosystem
(geschlossen/ offen??)
Analogie zu Apple (iOS, App-Store, ...)

KI-News

Anthropic – Neue Modelle, neue Funktionen



Anthropic – Claude, Claude Code

- Neue Sprachmodell Versionen
- Claude Code – KI-Software Entwickler
- MCP („Erfinder“ ... aber eigentlich sind es „nur“ Schnittstellen – APIs)
- Konnektoren (Gmail, Outlook, Canva, Teams, Sharepoint, ...)
- SDK für die Entwicklung von KI-Anwendungen

KI-News

Anthropic Skills (Capabilities)

The screenshot shows the 'Skills' section of the Anthropic interface. On the left, a sidebar menu includes 'Usage', 'Capabilities' (which is selected and highlighted in grey), 'Connectors', and 'Claude Code'. The main area is titled 'Skills' with a 'Preview' link. It contains a list of skills, each with a brief description, an 'Upload skill' button, and a toggle switch:

- algorithmic-art**: Creating algorithmic art using p5.js with seeded randomness and interactive parameter exploration. Use this when users request creating art using code, generative art, algorithmic art, flow fields, or particle systems....
Anthropic
- artifacts-builder**: Suite of tools for creating elaborate, multi-component claudia.ai HTML artifacts using modern frontend web technologies (React, Tailwind CSS, shadcn/ui). Use for complex artifacts requiring state management, routing, ...
Anthropic
- brand-guidelines**: Applies Anthropic's official brand colors and typography to any sort of artifact that may benefit from having Anthropic's look-and-feel. Use it when brand colors or style guidelines, visual formatting, or company design...
Anthropic
- canvas-design**: Create beautiful visual art in .png and .pdf documents using design philosophy. You should use this skill when the user asks to create a poster, piece of art, design, or other static piece. Create original visual designs, never...
Anthropic
- internal-comms**: A set of resources to help me write all kinds of internal communications, using the formats that my company likes to use. Claude should use this skill whenever asked to write some sort of internal communications (status...
Anthropic

Anthropic – Skills + Capabilities

- Anweisungen und Wissen, um Fähigkeiten für eine Aufgabe im Chat zu aktivieren, um eine bessere Antwort zu generieren
- Anstelle von Projekt- oder GPT- Instruktionen werden über ZIP-Dateien Anweisungen im Markdown-Format und Wissen in das Anthropic Account hochgeladen und diese neue „Fähigkeit“ ist dann über den Namen im Prompt nutzbar/ aktivierbar.

KI-News

Google

The screenshot shows the Google AI Studio interface. On the left, there's a sidebar with links like Home, Chat, Build, Dashboard, and Documentation. The main area has three tabs: Chat, Build, and Documentation. The Chat tab is active, showing a "Chat prompt" section with a text input field containing "Design a REST API for a social media platform." Below it is a "Run settings" panel for Gemini 2.5 Pro, which includes sliders for Temperature and Media resolution, and checkboxes for Thinking mode and Set thinking budget. The Build tab shows sections for Chat with models in the Playground, Vibe code GenAI enabled apps in Build, and Monitor usage and more in the Dashboard. The Documentation tab shows "What's new" sections for Nano Banana, Veo 3.1, Fetch information with URL context, and Generate native speech with Gemini. At the bottom, there's a "Get started with Gemini" section with a Python code example:

```
Python
from google import genai
client = genai.Client()
```

Google Ökosystem

- **Gemini 2.5**
 - TOP Sprachmodell
- **Nano Banana**
 - TOP KI-Bildgenerator
- **Veo 3.1**
 - TOP KI-Videogenerator
- **Notebook LM**
 - TOP Wissensmanagement

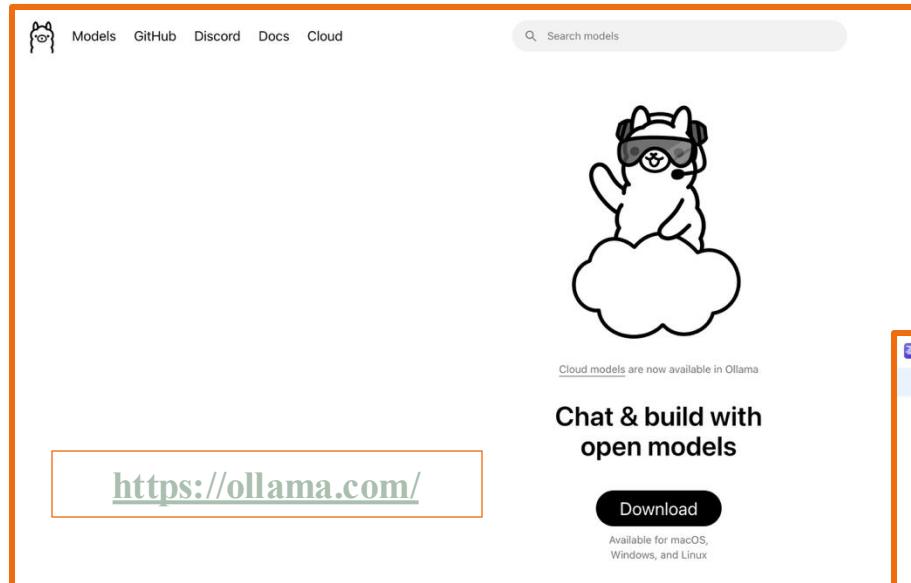
KI-News

OpenAI, Anthropic, Google, ...

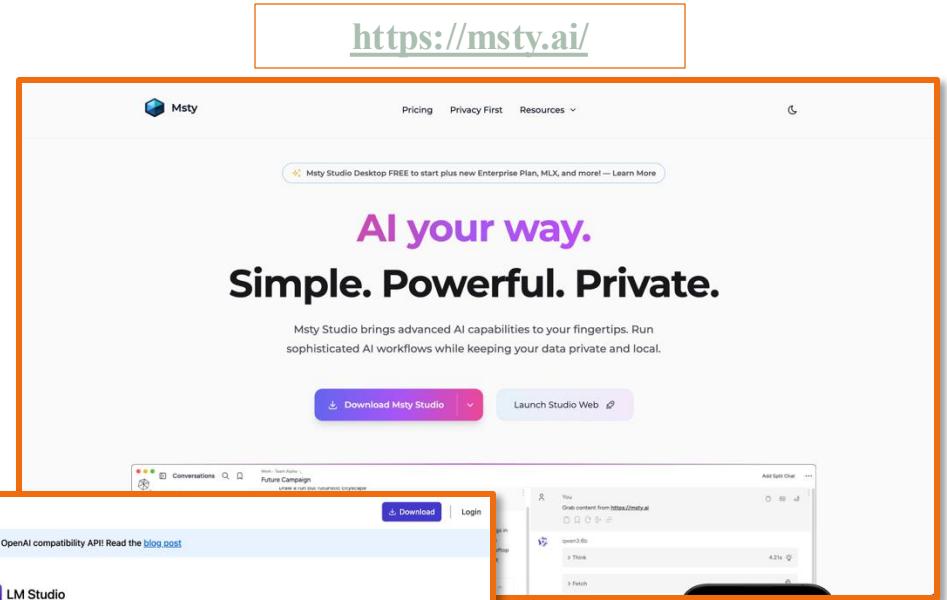
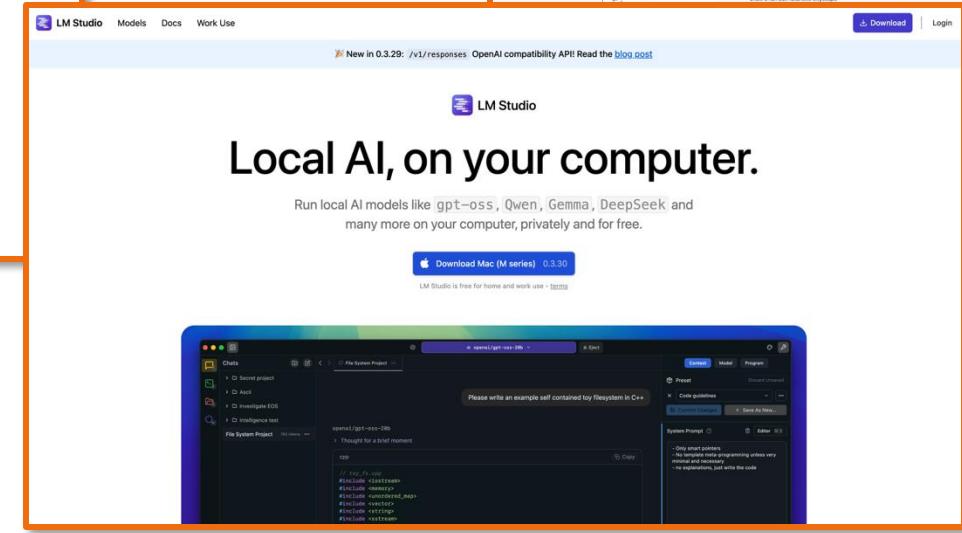
- KI steckt in der „Kinderschuhen“, wird aber schnell „erwachsen“
 - neue Funktionen, Features, ... fast wöchentlich
- Rechenzentren, Chip-Hersteller, Algorithmen, Daten, Talente
 - Kampf um Ressourcen
- Nischenlösungen mit Spezial-KI Lösungen (Texte, Bilder, Videos, Musik, ...)
 - StartUps und etablierte Unternehmen gewinnen Kunden mit KI-Lösungen in den vertikalen und horizontalen Märkten
- Regulatorik (EU-KI-Verordnung, DSGVO, Haftungsrichtlinien, ...)
 - Microsoft Azure AI Plattform Lösungen – Hosting in DE/ EU
 - Mistral Plattform – Hosting in EU
 - Hetzner, OVH, ... diverse Cloud-Anbieter mit Komplett-Lösungen
 - Open-Source Lösungen zum “Selber-Hosten”

Open Source Community

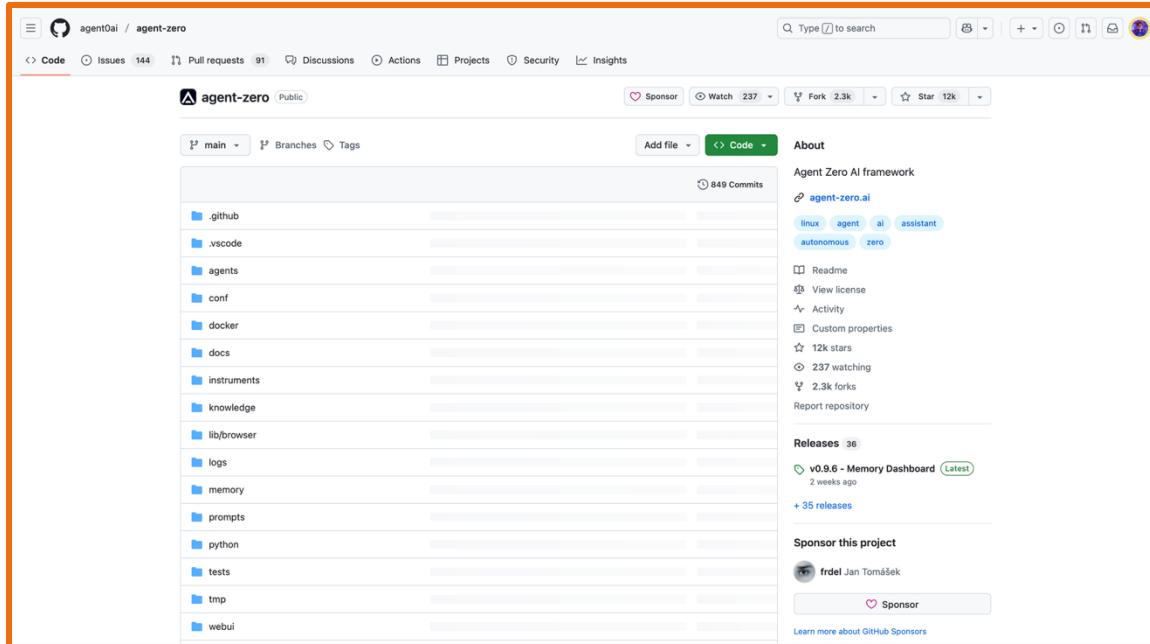
Offline arbeiten mit lokalen LLMs



Installation eines **lokalen** „ChatGPT“ auf dem eigenen Rechner mit Open-Source Sprachmodellen für die „**offline**“ Nutzung. Kann auch mit OpenAI ChatGPT und anderen „großen“ Sprachmodellen „**online**“ genutzt werden.

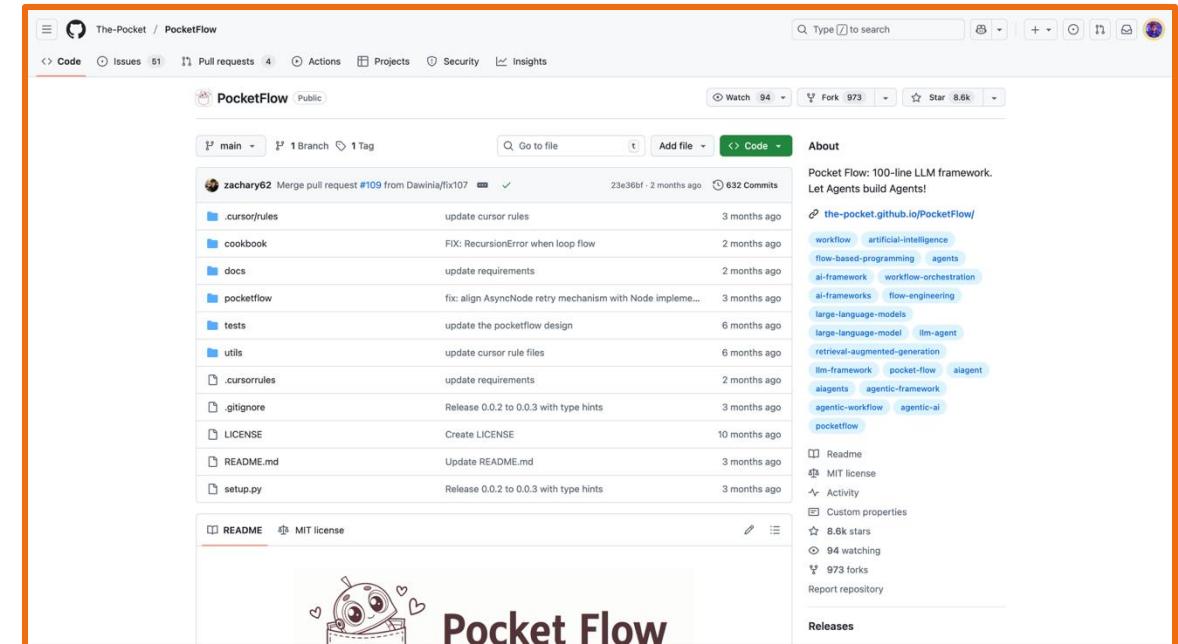


Open Source Community Agenten Systeme



The screenshot shows the GitHub repository page for 'agent0ai / agent-zero'. The repository has 144 issues, 91 pull requests, and 237 discussions. It has 2.3k forks and 12k stars. The repository is public and contains 849 commits. The codebase includes branches for 'main', 'Branches', and 'Tags'. Key files include .github, vscode, agents, conf, docker, docs, instruments, knowledge, libbrowser, logs, memory, prompts, python, tests, tmp, and webui. The 'About' section describes it as the 'Agent Zero AI framework' and lists tags such as linux, agent, ai, assistant, autonomous, and zero. It also includes a Readme, license, activity, custom properties, and a memory dashboard. There are 36 releases, with the latest being v0.9.6 - Memory Dashboard. A sponsorship section is present.

<https://github.com/agent0ai/agent-zero>



The screenshot shows the GitHub repository page for 'The-Pocket / PocketFlow'. The repository has 61 issues, 4 pull requests, and 94 discussions. It has 973 forks and 8.6k stars. The repository is public and contains 632 commits. The codebase includes main, Branches, and Tags. Key files include cursor/rules, cookbook, docs, pocketflow, tests, utils, cursorsrules, .gitignore, LICENSE, README.md, and setup.py. The 'About' section describes it as a '100-line LLM framework' for building Agents. It lists tags such as workflow, artificial-intelligence, flow-based-programming, agents, ai-framework, workflow-orchestration, ai-frameworks, flow-engineering, large-language-models, large-language-model, llm-agent, retrieval-augmented-generation, llm-framework, pocket-flow, alagent, agents, agentic-framework, agentic-workflow, agentic-ai, and pocketflow. It also includes a Readme, MIT license, activity, custom properties, and a memory dashboard. There are 36 releases, with the latest being Release 0.0.2 to 0.0.3 with type hints.

<https://github.com/The-Pocket/PocketFlow>

„Leichtgewichtige“ Python (Programmiersprache) Frameworks mit denen man Agenten und KI-Automatisierungen „programmieren“ kann.

Open Source Community

Senior Programmierer als KI-Mitarbeiter

<https://github.com/bmad-code-org/BMAD-METHOD>

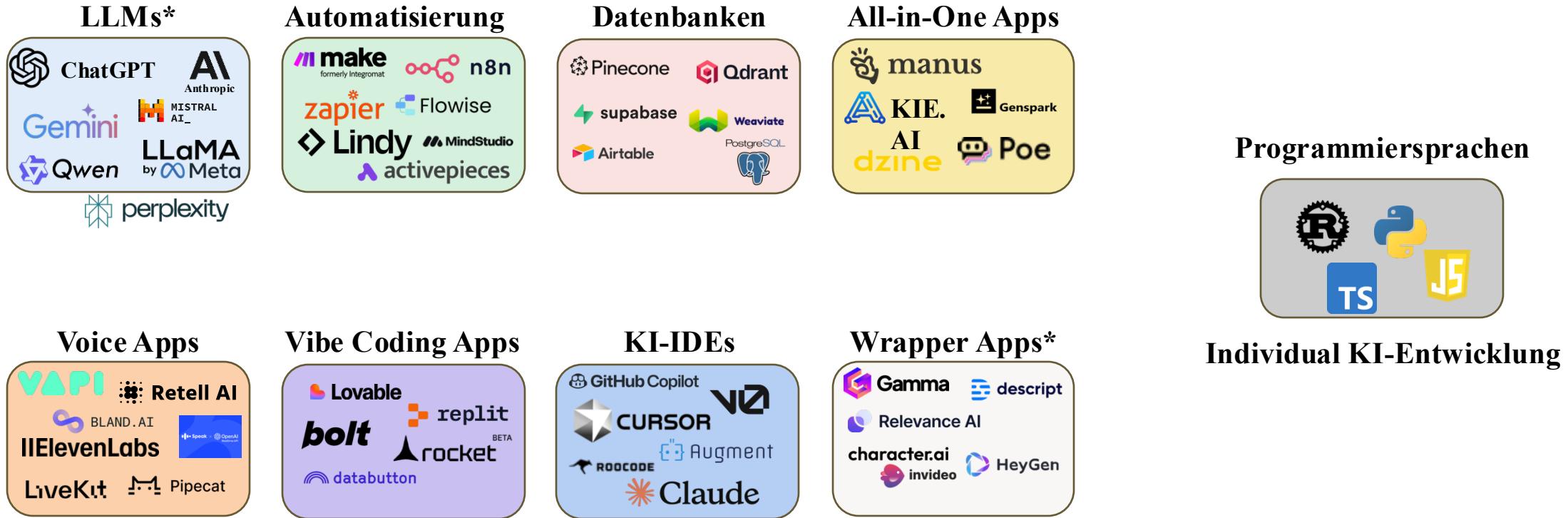
Agile Programmiermethode in Form von „Code“ als Grundlage für KI-Entwicklungstools bereitgestellt

Also per „Text- oder Spracheingabe“ wie mit einem echten Senior-Entwickler sprechen und dann:

**Per Brainstorming Ideen generieren
Ideen in einen Plan überführen
Plan in eine IT-Architektur überführen
Produktanforderungen für die Anwendung erstellen
Agile Entwicklungsmethode starten
Scrum-Master definieren
Sprint-Entwicklung
Automatisierte Tests der Anwendung
Automatisierte Dokumentation der Anwendung
...**

KI-Landkarte

Überblick



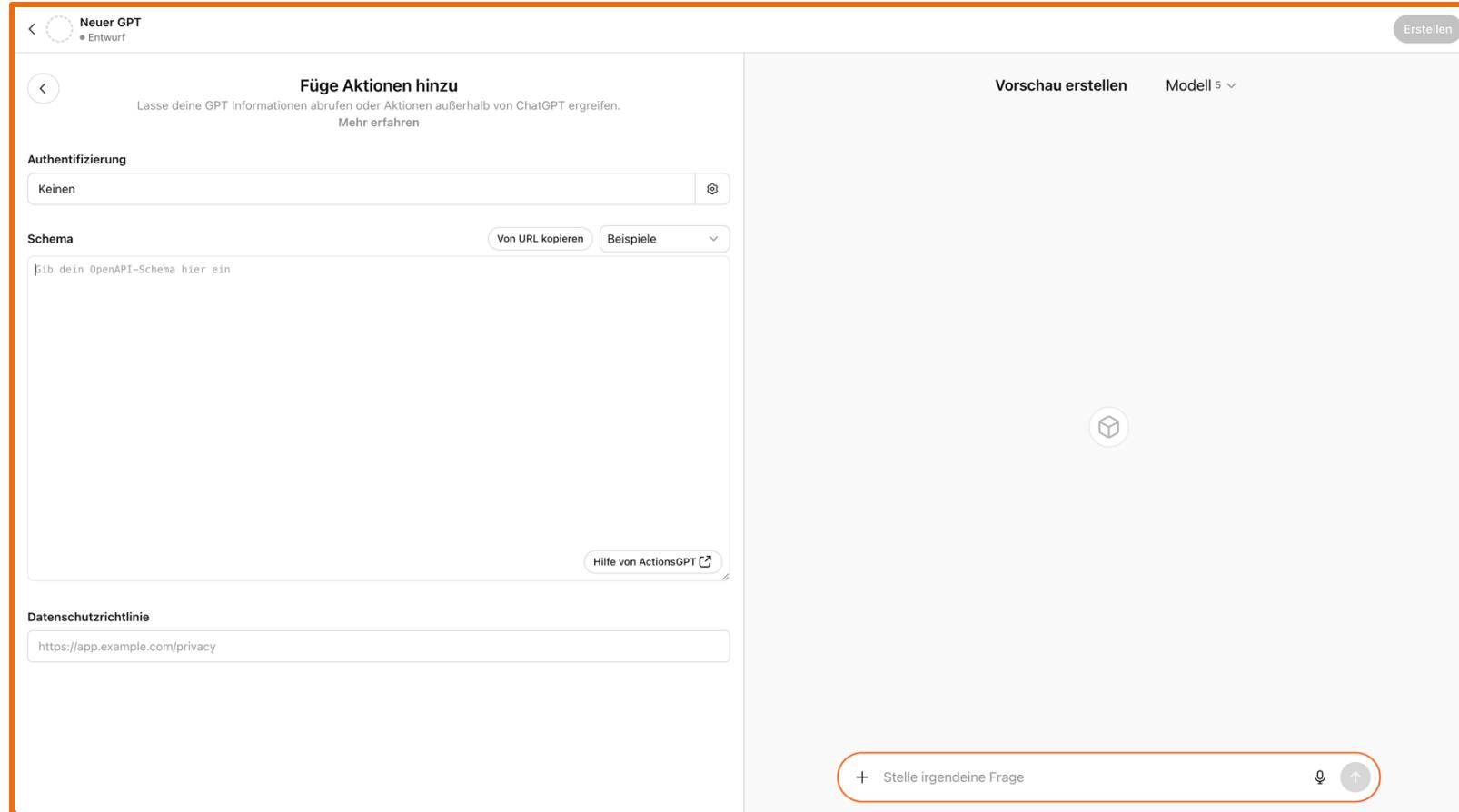
LLM = Large Language Model => Sprachmodelle wie ChatGPT, Wrapper = Webseite

Praxisbausteine

GPTs, Projekte, MCP, Aktionen, Agent Builder, Claude Skills, N8N, ...

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...



OpenAI -> GPTs -> Action:

- API/ Schema Abfrage
- Erweiterung von Funktionen von ChatGPT
- Datenabfrage

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

The screenshot shows the N8N interface. On the left, there's a sidebar with a user profile for "SEO Experte Rolf" and a "Füge Aktionen hinzu" (Add Actions) section. Below it are sections for "Authentifizierung" (Authentication) and "Schema". The "Schema" section displays a JSON configuration for the DataForSEO API. On the right, there's a "Vorschau erstellen" (Create Preview) section for "Modell 5" which generates a response about SEO functions. At the bottom, there's a "Verfügbare Aktionen" (Available Actions) table listing various API endpoints like "serp_organic_live_advanced", "keywords_data_google_ads_search_volume", and "on_page_content_parsing".

OpenAI -> GPTs -> Action:

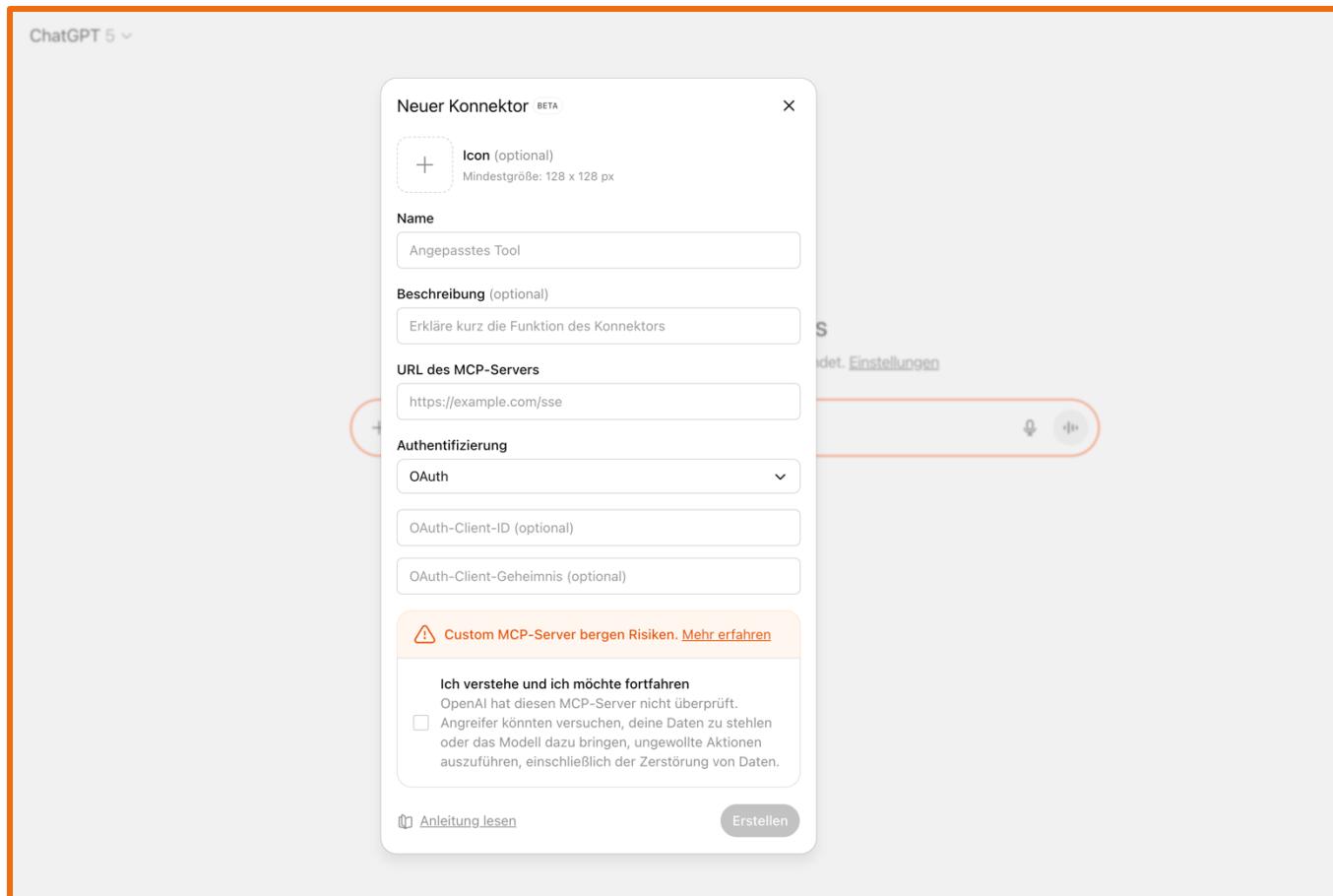
- API/ Schema Abfrage
- Erweiterung von Funktionen von ChatGPT
- Datenabfrage

Beispiel – Action -> DataSEO

- SEO-Funktionen von DataSEO über Actions (API) bereitgestellt

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...



OpenAI -> Konnektoren -> MCP

- Verbindung zu APIs, Tools, ...
- Hier ein Beispiel, um über Klavis (MCP-Marktplatz) mit OpenRouter zu arbeiten (um weitere LLMs einzubinden)

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

The screenshot shows the Klavis interface for managing connectors. On the left, a sidebar lists various categories: Allgemein, Benachrichtigungen, Personalisierung, Apps und Konnektoren (which is selected and highlighted in grey), Zeitpläne, Datenkontrollen, Sicherheit, Kindersicherung, and Konto. The main area displays a connector named "Klavis - >OpenRouter->LLMs MCP". It includes a "DEV" status indicator, a "Trennen" (Disconnect) button, and a three-dot menu. Below this, there's an "Info" section showing the connection status ("Verbunden am 11. Nov. 2025") and the URL ("https://openrouter-mcp-server.klavis.ai/mcp/?instance_i"). The "Aktionen" (Actions) section lists several functions: "openrouter_analyze_model_performance" (Analyze the performance of a single model across multiple test prompts), "openrouter_compare_models" (Compare multiple models by running the same prompt through each), "openrouter_create_chat_completion" (Create a chat completion using OpenRouter), and "openrouter_create_chat_completion_stream" (Create a streaming chat completion using OpenRouter). There are also "SCHREIBEN" (Write) buttons next to the last two actions.

OpenAI -> Konnektoren -> MCP

- Verbindung zu APIs, Tools, ...

Beispiel OpenRouter

- MCP Server Openrouter ermöglicht die Nutzung von z.B. Anthropic Claude aus ChatGPT heraus

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

The screenshot shows the OpenRouter interface. At the top, there's a search bar with the placeholder "Welche Sprachmodelle kann ich mit Openrouter nutzen?". Below it, a message says "Gespräch mit connector beendet". A button labeled "(✉) Tool aufgerufen" is present. A note reads: "Kurzfassung: Mit OpenRouter kannst du eine riesige Auswahl an Modellen verschiedenster Anbieter nutzen – von „Frontier“-LLMs über günstige Schnellläufer bis hin zu Coding-, Vision- und Research-Spezialisten. Hier ein übersichtlicher Überblick (Auszug der wichtigsten Familien):". A section titled "Top-Anbieter & Modellfamilien" lists providers: OpenAI, Anthropic, Google, Meta, Mistral, DeepSeek, and Qwen. Each provider has a list of specific models. At the bottom, there's a text input field with placeholder "Stelle irgendeine Frage" and a "Create" button.

OpenAI -> Konnektoren -> MCP

- Klavis MCP Marktplatz -> OpenRouter
- Mehrwert:**
- Zugriff auf andere Sprachmodelle aus OpenAI ChatGPT heraus

The screenshot shows a tool creation dialog. It starts with a message "(✉) ChatGPT möchte mit call_tool sprechen". Below it, a red box contains "Klavis->OpenRouter->LLMs MCP" and "Allow creation of 5 customer complaint reply emails?". A note explains: "Allow creation of 5 professional, empathetic, solution-focused German reply emails addressing customer complaints like late delivery, defective goods, unfriendly support, wrong or incomplete items. [Details](#)". At the bottom, there are "Create" and "Verweigern" buttons, and a note about tool usage: "Die Verwendung von Tools ist mit Risiken verbunden. [Mehr erfahren](#)".

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

The screenshot shows the Claude AI interface with a sidebar on the left containing navigation links: Allgemein, Konto, Datenschutz, Abrechnung, Nutzung, **Fähigkeiten** (which is highlighted with a blue border), Konnektoren, and Claude Code. The main area is titled "Skills" and "Vorschau". It lists several skills with their descriptions and toggle switches:

- bwa-voba-rk: Automatisierte Betriebswirtschaftliche Auswertung (BWA) für Geschäftsberichte der VR-Bank. Verarbeitet PDF-Bilanzien und Jahresabschlüsse, extrahiert Kennzahlen, führt Branchenvergleiche durch und erstellt... (von dir hinzugefügt • vor 4 Tagen)
- audio-meeting-analyzer: Transcribe audio recordings and generate comprehensive meeting analysis including protocols, summaries, action items, and meeting quality evaluation. Use when user uploads audio files (MP3, WAV, M4A), mentions... (von dir hinzugefügt • vor 20 Tagen)
- algorithmic-art: Creating algorithmic art using p5.js with seeded randomness and interactive parameter exploration. Use this when users request creating art using code, generative art, algorithmic art, flow fields, or particle systems.... (Anthropic)
- artifacts-builder: Suite of tools for creating elaborate, multi-component claudia.ai HTML artifacts using modern frontend web technologies (React, Tailwind CSS, shadcn/ui). Use for complex artifacts requiring state management, routing, ... (Anthropic)
- brand-guidelines: Applies Anthropic's official brand colors and typography to any sort of artifact that may benefit from having Anthropic's look-and-feel. Use it when brand colors or style guidelines, visual formatting, or company design... (Anthropic)
- canvas-design: Create beautiful visual art in .png and .pdf documents using design philosophy. You should use this skill when the user asks to create a poster, piece of art, design, or other static piece. Create original visual designs, never... (Anthropic)

Anthropic Claude -> Skills:

- Erweiterungen des Funktionsumfangs eines Sprachmodells
- Zusätzliche „Fähigkeiten“ hinzufügen
- Skills können über natürliche Sprache „konfiguriert“ und erstellt werden

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

Einstellungen

Allgemein

Konto

Datenschutz

Abrechnung

Nutzung

Fähigkeiten

Konnektoren

Claude Code

Konnektoren

Erlauben Sie Claude, andere Apps und Dienste für mehr Kontext zu referenzieren.

Konnektor	Status	Aktionen
Google Drive	Verbunden	...
Github	Verbunden	...
Canva	Konfigurieren	...
Gmail	Getrennt	Verbinden
Google Calendar	Getrennt	Verbinden
Infranodus	BENUTZERDEFINIERT https://server.smithery.ai/infranodus/mcp-serv...	Konfigurieren ...

Connectors durchsuchen

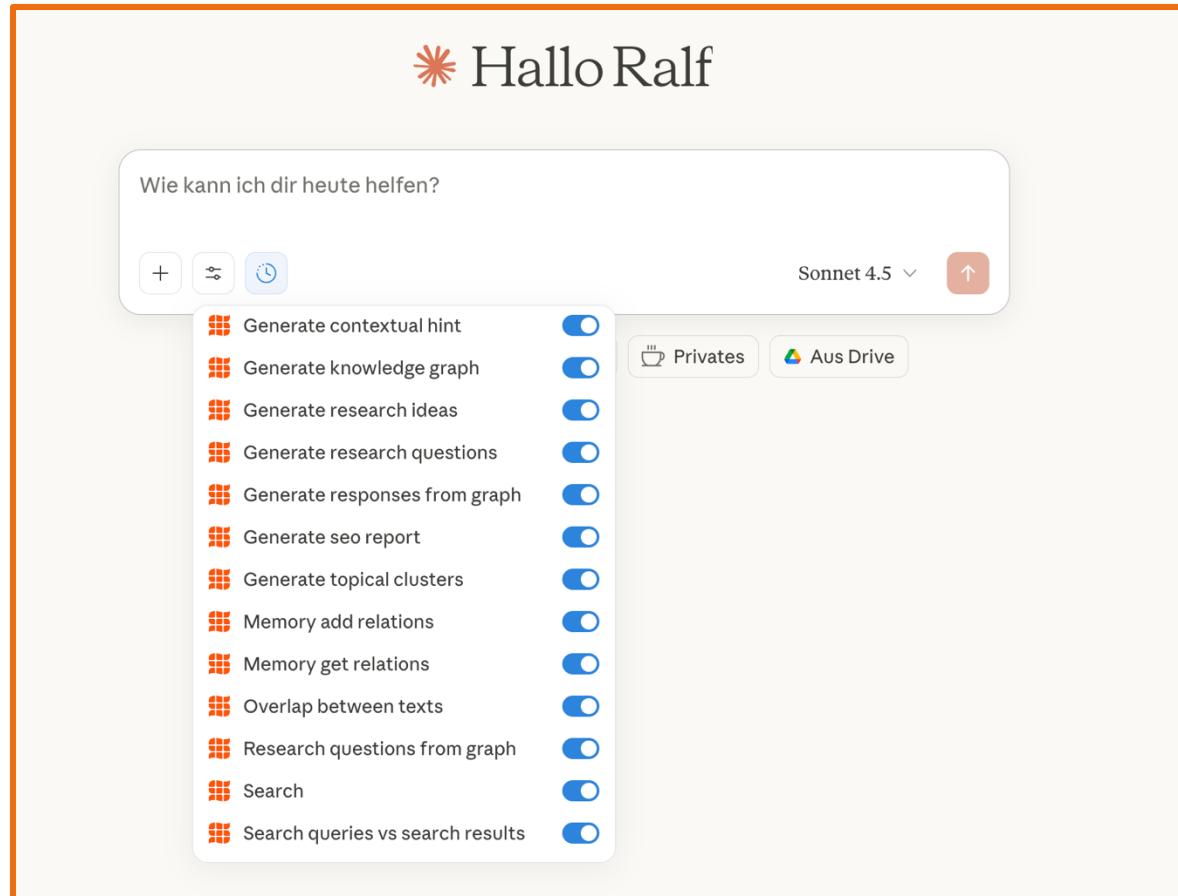
Benutzerdefinierten Connector hinzufügen

Antropic Claude -> Konnektoren (MCPs):

- Verbindung (APIs) zu anderen Services u.a. Tools, Daten, ...
- Hier das Beispiel Infranodus (Graph-Wissensmanagement-Technologien)

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

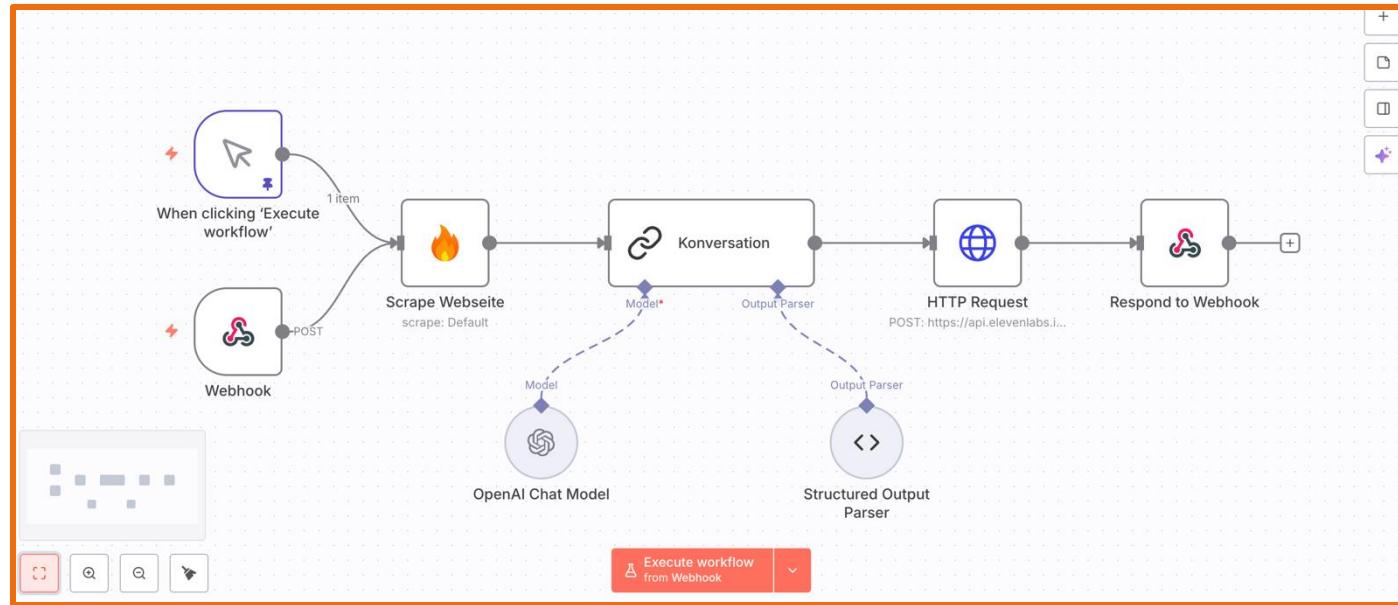


Claude -> MCP Infranodus -> Neue „Werkzeuge“

- MCP-Konnektoren ermöglichen den Zugriff über API Keys auf Daten und Werkzeuge
- Mit den zuvor genannten Skills kann man damit „komplexe“ Agenten erstellen.

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...

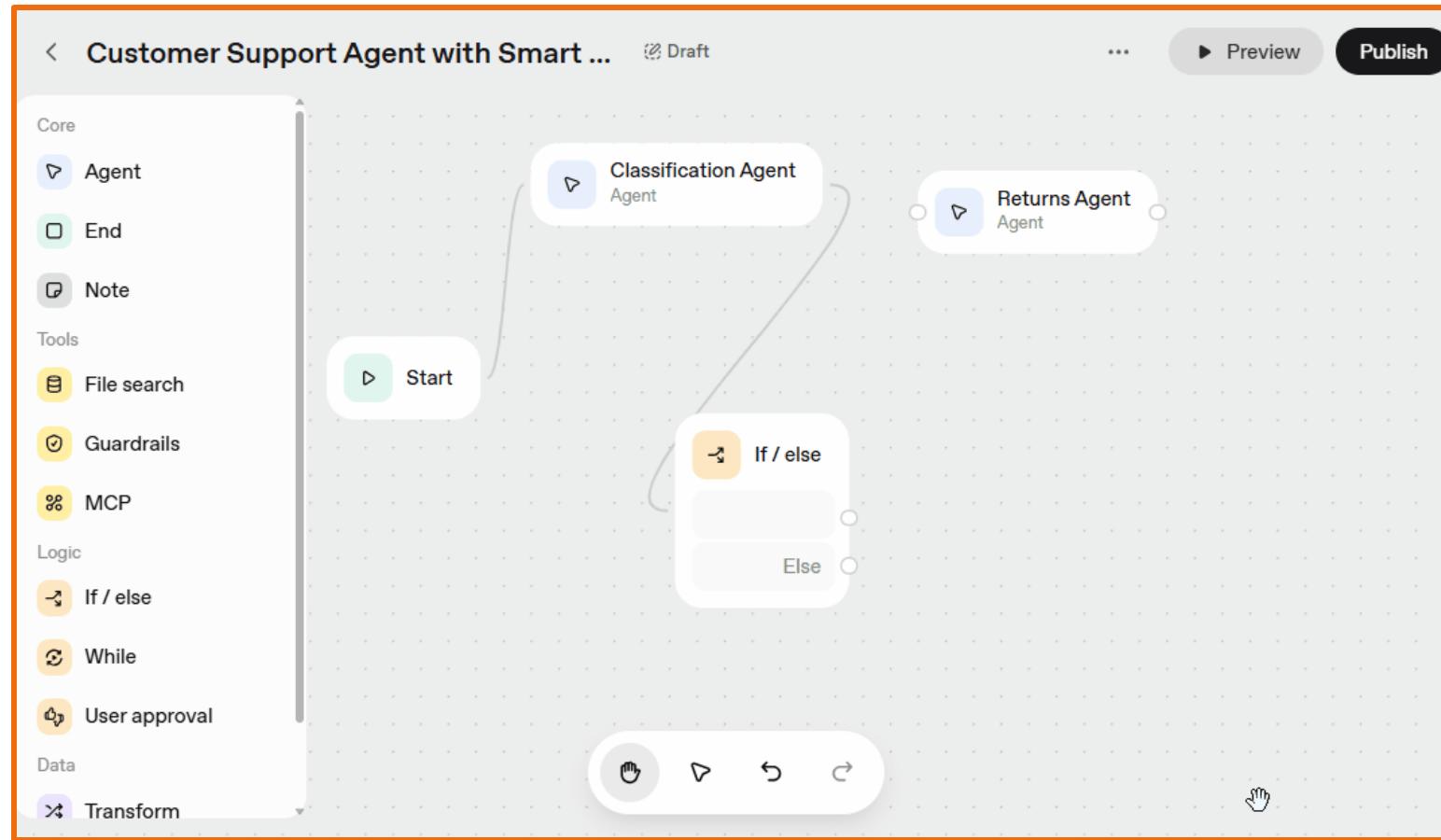


N8N – Automatisierungsplattform:

- Klassische Automatisierungsplattform
- Integration von KI-Nodes
- Integration von APIs von Drittanbietern (Elevenlabs, Firecrawl, ...)

GPTs, Projekte, Skills, Agenten, Workflows

OpenAI, Anthropic, N8N, ...



OpenAI -> Agent Builder & ChatKit & ...

- Agent Builder
- ChatKit
- Guardrails
- AgentsSDK

OpenAI => komplettes Ökosystem
(geschlossen/ offen??)
Analogie zu Apple (iOS, App-Store, ...)

Agenten (Node, Graph, Flow), ...

Ein paar Grundlagen

LLM-Framework mit 100 Zeilen Python Code



Ein paar Begriffe am Anfang klären

- **LLM** steht für **Large Language Model**, also ein großes Sprachmodell wie ChatGPT
- Neben **ChatGPT** von **OpenAI** gibt es zahlreiche andere **Sprachmodelle**: Claude, Gemini, Grok, DeepSeek, Kimi, ... (>1 Mio. Sprachmodelle siehe u.a. HuggingFace Plattform)
- Für die **Automatisierung** von digitalen Prozessen gibt es Frameworks wie **N8N**, **Zapier**, **Make** und viele mehr, die auch Sprachmodelle integrieren – über **Nodes**
- Ein **Node** (Knoten) in der Automatisierung ist ein Element, dass eine bestimmte Aufgabe erfüllt und könnte z.B. ein Sprachmodell sein
- Ein **Graph** ist ein Prozessablauf, wie z.B. der Prozess für einen Kreditantrag, der in einer Prozessbeschreibung, Verfahrensanweisung, ... festgelegt ist
- Ein **Flow** ist der konkrete Durchlauf eines Prozesses für den jeweiligen Input (Kunde, Trigger, ...)

LLM-Framework mit 100 Zeilen Python Code



Node: ein einzelner Bearbeitungsschritt

- Ein **Node** ist wie ein Schreibtisch/Abteilung in deiner Volksbank, wo EINE bestimmte Aufgabe erledigt wird:

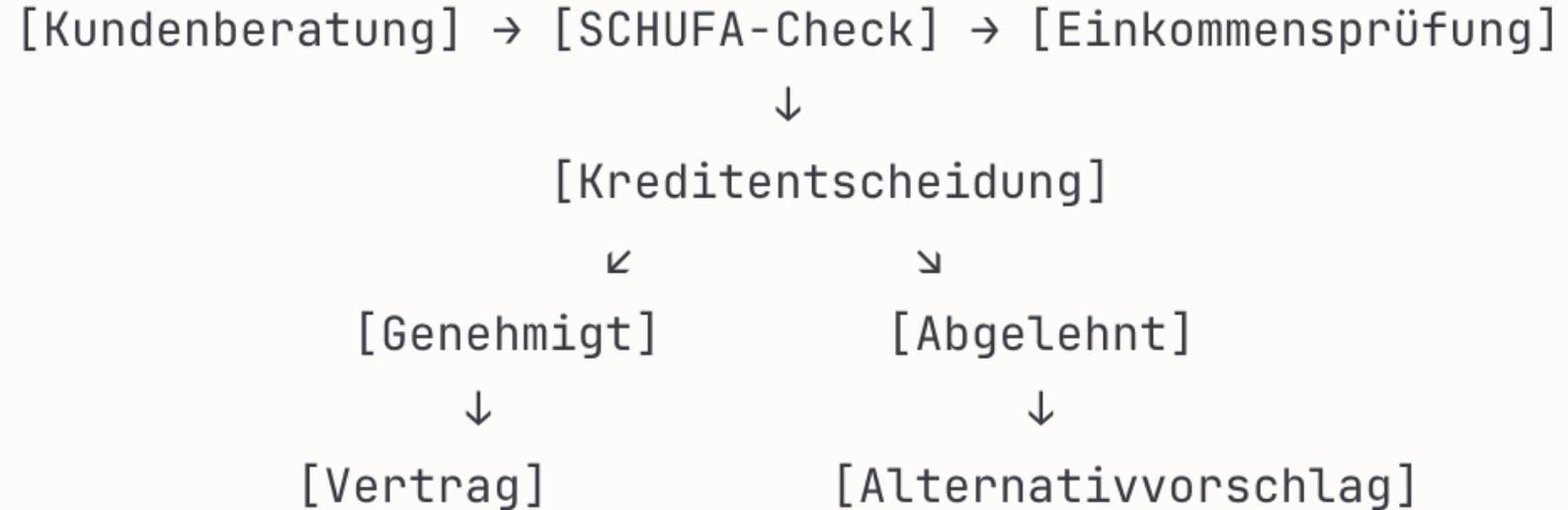
- 👤 Node 1: "Kundenberatung"
→ Aufgabe: Kundenwunsch erfassen (Kredithöhe, Zweck)
- 📊 Node 2: "SCHUFA-Prüfung"
→ Aufgabe: Bonität checken, SCHUFA-Score abrufen
- 💰 Node 3: "Einkommensprüfung"
→ Aufgabe: Gehaltsnachweise prüfen, Haushaltsrechnung
- ✓ Node 4: "Kreditentscheidung"
→ Aufgabe: Bewilligen oder Ablehnen basierend auf Prüfungen
- 📄 Node 5: "Vertragsabteilung"
→ Aufgabe: Kreditvertrag erstellen und versenden

- **Wichtig:** Jeder **Node** ist wie ein **Spezialist** - macht **NUR** seinen Job perfekt!

LLM-Framework mit 100 Zeilen Python Code

Graph: der komplette Kreditprozess

- Ein **Graph** ist der Prozessablauf, wie er in eurer Verfahrensanweisung steht:



- Der Graph ist wie ein **internes Prozesshandbuch** - zeigt alle Stationen und Entscheidungswege.

LLM-Framework mit 100 Zeilen Python Code

Flow: ein konkreter “Kunde“ durchläuft den Prozess

- Der **Flow** ist, wenn ein Kunde „Herr Müller“ WIRKLICH für seinen Autokredit durch alle Abteilungen läuft:

```
START: Herr Müller möchte 25.000€ für einen VW Golf
↓
1. Beratungsgespräch: 14:00 Uhr, Büro 3 (Node 1 wird ausgeführt)
↓
2. SCHUFA-Abfrage: Score 95, sehr gut! (Node 2 wird ausgeführt)
↓
3. Gehaltsprüfung: 3.500€ netto, passt! (Node 3 wird ausgeführt)
↓
4. Entscheidung: GENEHMIGT (Node 4 wird ausgeführt)
↓
5. Vertrag wird erstellt (Node 5 wird ausgeführt)
↓
ENDE: Herr Müller unterschreibt, Auto kann gekauft werden! 🚗
```

LLM-Framework mit 100 Zeilen Python Code



Wie sieht das als Code jetzt aus?

- Der Kreditantragsprozess als KI-System:

```
# Die NODES (Bearbeitungsstationen)
class KundenberatungNode(Node):
    def exec(self, kundenantrag):
        # Wie dein Beratungstool
        return {
            "kunde": "Herr Müller",
            "betrag": 25000,
            "zweck": "Autokredit"
        }

class SchufaPruefungNode(Node):
    def exec(self, kundendaten):
        # Wie eure SCHUFA-Schnittstelle
        schufa_score = schufa_api.check(kundendaten["kunde"])
        return {"score": 95, "bewertung": "sehr gut"}

class EinkommenspruefungNode(Node):
    def exec(self, antrag_mit_schufa):
        # Wie euer Haushaltsrechner
        return {
            "netto": 3500,
            "verfuegbar": 800,
            "rate_moeglich": 450
        }
```

```
# Der GRAPH (euer Kreditprozess)
kredit_prozess = Flow(
    nodes=[
        KundenberatungNode(),      # Station 1
        SchufaPruefungNode(),      # Station 2
        EinkommenspruefungNode(),  # Station 3
        KreditentscheidungNode(),  # Station 4
        VertragsNode()             # Station 5
    ]
)

# Der FLOW (Herr Müller's Antrag läuft durch)
ergebnis = kredit_prozess.run({"kunde": "Herr Müller", "wunsch": "25000€"})
# Ausgabe: "Kreditvertrag erstellt - GENEHMIGT"
```

.... weitere Nodes

LLM-Framework mit 100 Zeilen Python Code

 Zusammengefasst – Kreditantrag

Begriff	Volksbank-Beispiel	KI-System Entsprechung
Node 	Ein Bearbeitungsschritt (z.B. SCHUFA-Prüfung)	Ein Code-Baustein für genau diese Prüfung
Graph 	Eure Kreditrichtlinie/Prozesshandbuch	Alle Prüfschritte und deren Reihenfolge
Flow 	Ein konkreter Kunde durchläuft den Prozess	Computer arbeitet einen echten Antrag ab

LLM-Framework mit 100 Zeilen Python Code



Warum ist das so spannend?

- Compliance konforme Lösung

```
# Jeder Node protokolliert automatisch für die BaFin
class SchufaNode(Node):
    def post(self, shared, prep_res, exec_res):
        # Automatische Dokumentation für Prüfungspfad
        shared["pruefprotokoll"].append({
            "zeitstempel": datetime.now(),
            "pruefung": "SCHUFA",
            "ergebnis": exec_res
        })
```

LLM-Framework mit 100 Zeilen Python Code



Warum ist das so spannend?

- Verschiedene Kreditarten – gleiche Nodes

```
# Baufinanzierung? Nutze die gleichen Bausteine!
baufi_prozess = Flow([
    KundenberatungNode(),
    SchufaPruefungNode(),
    EinkommenspruefungNode(),
    GrundbuchNode(),          # NEU: Nur bei Baufi
    KreditentscheidungNode(),
    VertragsNode()
])

# Dispokredit? Vereinfachter Prozess!
dispo_prozess = Flow([
    SchufaPruefungNode(),    # Weniger Nodes!
    KreditentscheidungNode()
])
```

LLM-Framework mit 100 Zeilen Python Code



Warum ist das so spannend?

- Risikomanagement eingebaut

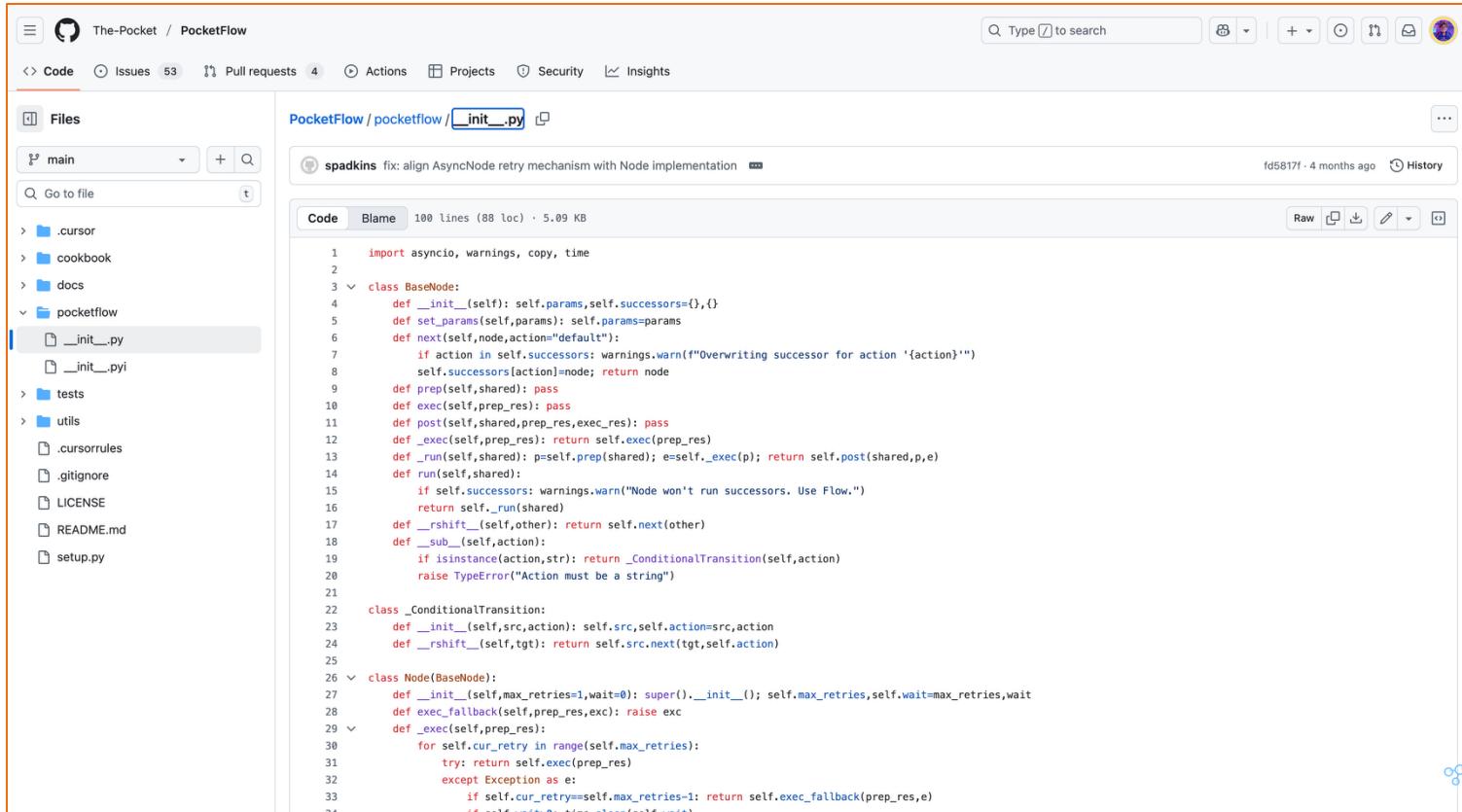
```
class RisikoNode(Node):  
    def post(self, shared, prep_res, exec_res):  
        if exec_res["score"] < 50:  
            return "zur_filialleitung" # → Automatische Eskalation  
        else:  
            return "standard_prozess" # → Normal weiter
```

Programmierung mit KI

100 Zeilen in Python – kurzer Blick auf die Elemente und den Code

LLM-Framework mit 100 Zeilen Python Code

Nodes



The screenshot shows a GitHub repository page for 'The-Pocket / PocketFlow'. The left sidebar lists files in the 'main' branch, including .cursor, cookbook, docs, pocketflow (with __init__.py selected), tests, utils, .cursorrules, .gitignore, LICENSE, README.md, and setup.py. The main content area displays the code for __init__.py. The code is as follows:

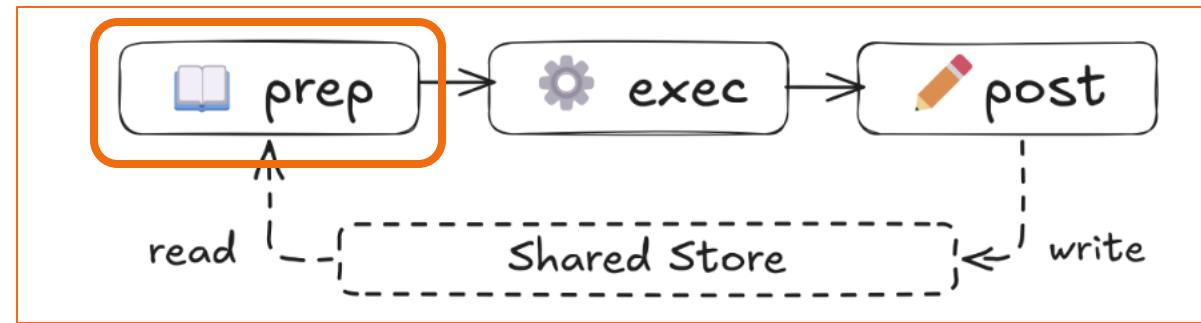
```
1 import asyncio, warnings, copy, time
2
3 class BaseNode:
4     def __init__(self):
5         self.params = {}
6         self.successors = {}
7         self.action = "default"
8
9     def next(self, node):
10        if node in self.successors:
11            warnings.warn(f"Overwriting successor for action '{node}'")
12        self.successors[node] = node
13
14    def prep(self, shared):
15        pass
16
17    def exec(self, prep_res):
18        pass
19
20    def post(self, shared, prep_res, exec_res):
21        pass
22
23    def _exec(self, prep_res):
24        return self.exec(prep_res)
25
26    def _run(self, shared):
27        p = self.prep(shared)
28        e = self._exec(p)
29        self.post(shared, p, e)
30
31    def run(self, shared):
32        if self.successors:
33            warnings.warn("Node won't run successors. Use Flow.")
34        else:
35            self._run(shared)
36
37    def __rshift__(self, other):
38        return self.next(other)
39
40    def __sub__(self, action):
41        if isinstance(action, str):
42            return _ConditionalTransition(self, action)
43        raise TypeError("Action must be a string")
44
45    class _ConditionalTransition:
46        def __init__(self, src, action):
47            self.src = src
48            self.action = action
49
50        def __rshift__(self, tgt):
51            return self.src.next(tgt, self.action)
52
53    class Node(BaseNode):
54        def __init__(self, max_retries=1, wait=0):
55            super().__init__()
56            self.max_retries = max_retries
57            self.wait = wait
58
59        def exec(self, prep_res, exc):
60            self._exec(self, prep_res)
61
62            for self.cur_retry in range(self.max_retries):
63                try:
64                    self._exec(self, prep_res)
65                except Exception as e:
66                    if self.cur_retry == self.max_retries - 1:
67                        return self._exec(self, prep_res, e)
68
69                if self.wait > 0:
70                    time.sleep(self.wait)
```

<https://github.com/The-Pocket/PocketFlow/tree/main>

LLM-Framework mit 100 Zeilen Python Code

Nodes

Ein **Node** ist der kleinste Baustein. Jeder Node hat 3 Schritte **prep->exec->post**:



1. prep(shared)

Lese und bereite Daten vor aus dem gemeinsamen Speicher.

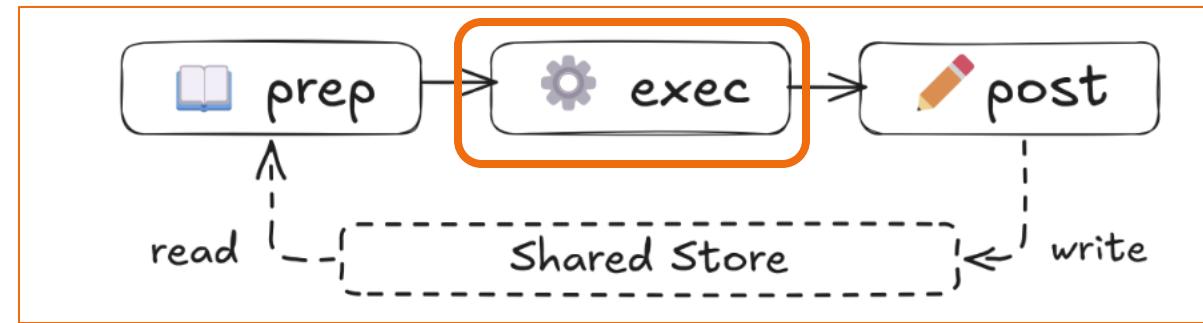
Beispiele: Datenbank abfragen, Dateien lesen oder Daten in einen String serialisieren.

Gibt prep_res zurück, welches von exec() und post() verwendet wird.

LLM-Framework mit 100 Zeilen Python Code

Nodes

Ein Node ist der kleinste Baustein. Jeder Node hat 3 Schritte prep->exec->post:



2. exec(prep_res)

Führe Berechnungslogik aus, mit optionalen Wiederholungen und Fehlerbehandlung.

Beispiele: (hauptsächlich) LLM-Aufrufe, externe APIs, Tool-Nutzung (MCP, ...)

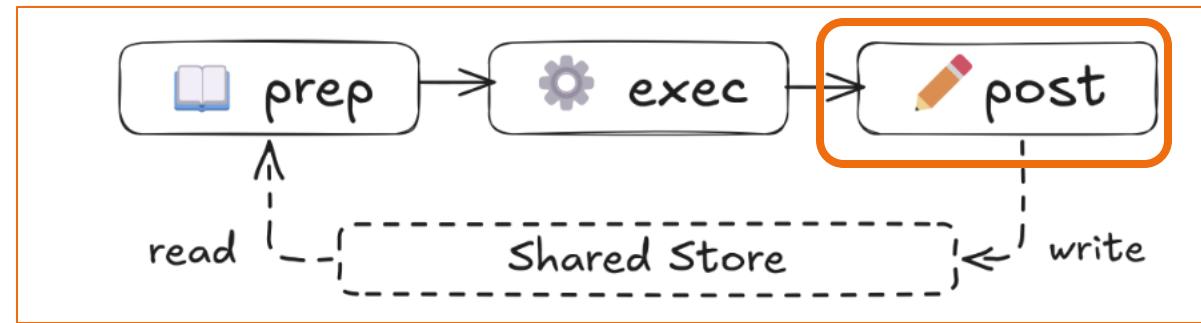
- ⚠ Dies soll NUR für Berechnungen sein und greift NICHT auf shared Store zu.
- ⚠ Falls Wiederholungen aktiviert sind, stelle idempotente Implementierung sicher.
- ⚠ Überlasse Fehlerbehandlung dem eingebauten Wiederholungsmechanismus des Nodes.

Gibt exec_res zurück, welches an post() weitergegeben wird.

LLM-Framework mit 100 Zeilen Python Code

Nodes

Ein Node ist der kleinste Baustein. Jeder Node hat 3 Schritte prep->exec->post:



3. post(shared, prep_res, exec_res)

Nachbearbeitung und schreibe Daten zurück zu shared.

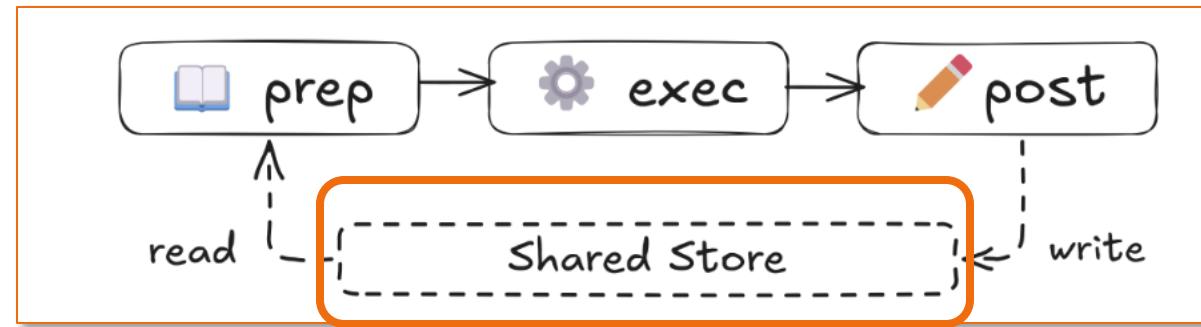
Beispiele: Datenbank aktualisieren, Status ändern, Ergebnisse protokollieren.

Entscheide die nächste Aktion durch Rückgabe eines string (action = "default" falls None).

LLM-Framework mit 100 Zeilen Python Code

Nodes

Ein Node ist der kleinste Baustein. Jeder Node hat 3 Schritte prep->exec->post:



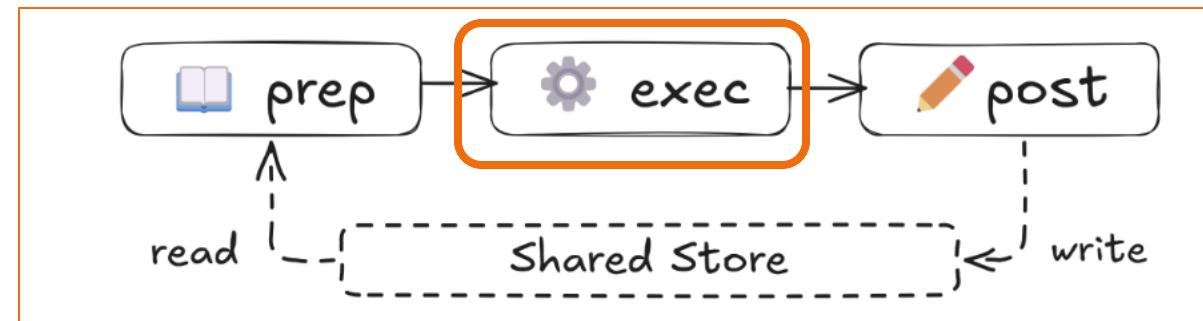
Shared Store (Gemeinsamer Speicher)

Die gestrichelte Box in der Mitte des Diagramms
prep liest aus dem gemeinsamen Speicher
post schreibt in den gemeinsamen Speicher
exec hat **KEINEN** direkten Zugriff

LLM-Framework mit 100 Zeilen Python Code

Nodes

Ein Node ist der kleinste Baustein. Jeder Node hat 3 Schritte prep->exec->post:



Du kannst exec() wiederholen lassen, wenn es eine Ausnahme (Exception) auslöst, über zwei Parameter beim Definieren des Nodes:

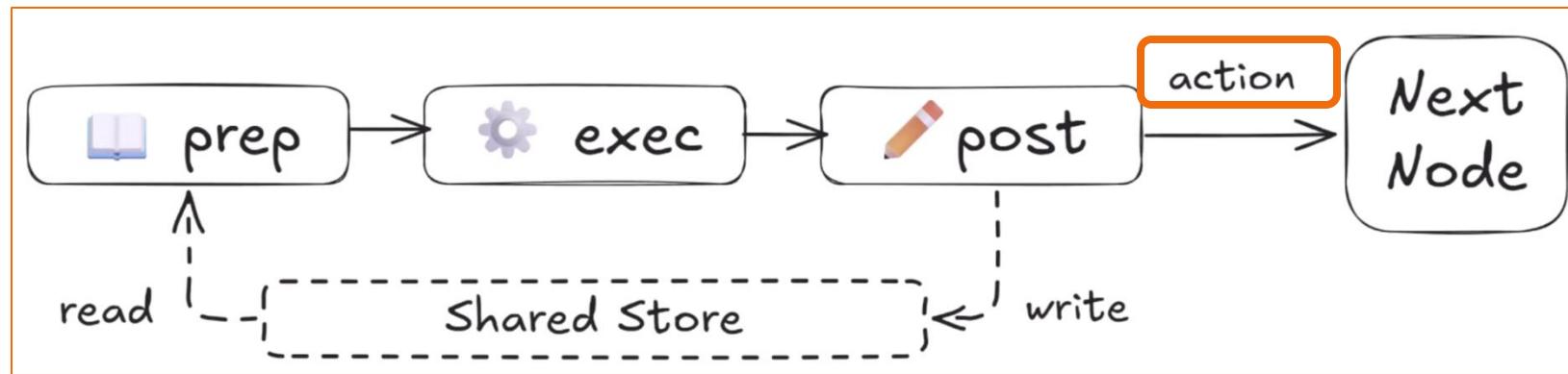
max_retries (Ganzzahl): Maximale Anzahl, wie oft exec() ausgeführt wird. Der Standard ist 1 (keine Wiederholung).

wait (Ganzzahl): Die Wartezeit (in Sekunden) vor dem nächsten Versuch. Standardmäßig ist wait=0 (keine Wartezeit). wait ist hilfreich, wenn du auf Ratenlimits oder Kontingentfehler deines LLM-Anbieters stößt und eine Pause einlegen musst.

LLM-Framework mit 100 Zeilen Python Code

🤔 Flows – Node A >> Node B >> Node C >>...

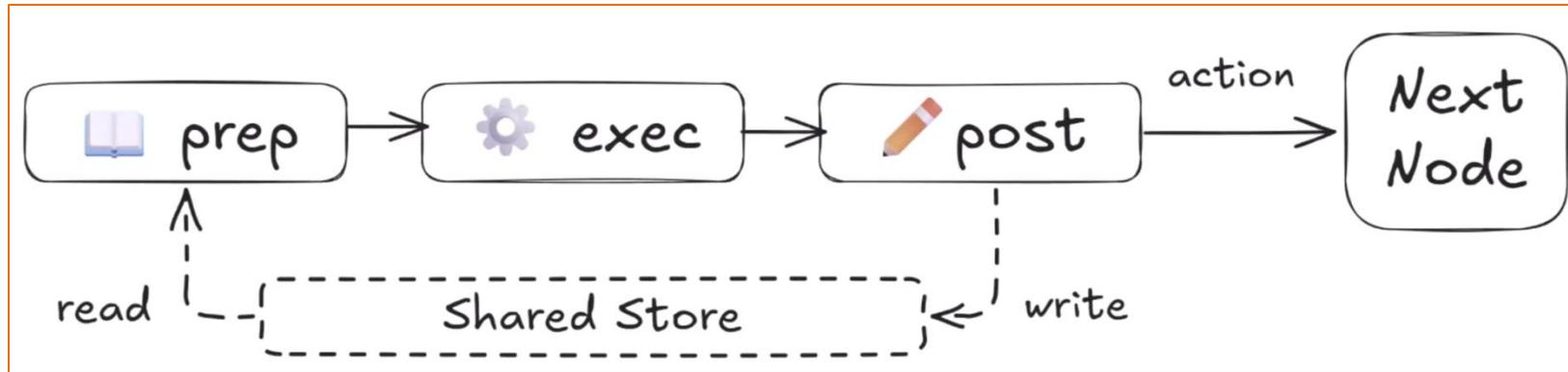
Wie funktioniert die Überleitung zum nächsten Node: per action im post Ergebnis des vorhergehenden Nodes:



LLM-Framework mit 100 Zeilen Python Code

⚠️ Flows: Node A >> Node B >> ...

Ein Flow orchestriert einen Graphen von Nodes. Du kannst Nodes in einer Sequenz verketteten oder Verzweigungen erstellen, abhängig von den Actions, die von jedem Node's post() zurückgegeben werden.



1. Action-basierte Übergänge

Jeder Node's post() gibt einen Action-String zurück. Standardmäßig, wenn post() nichts zurückgibt, behandeln wir das als "default".

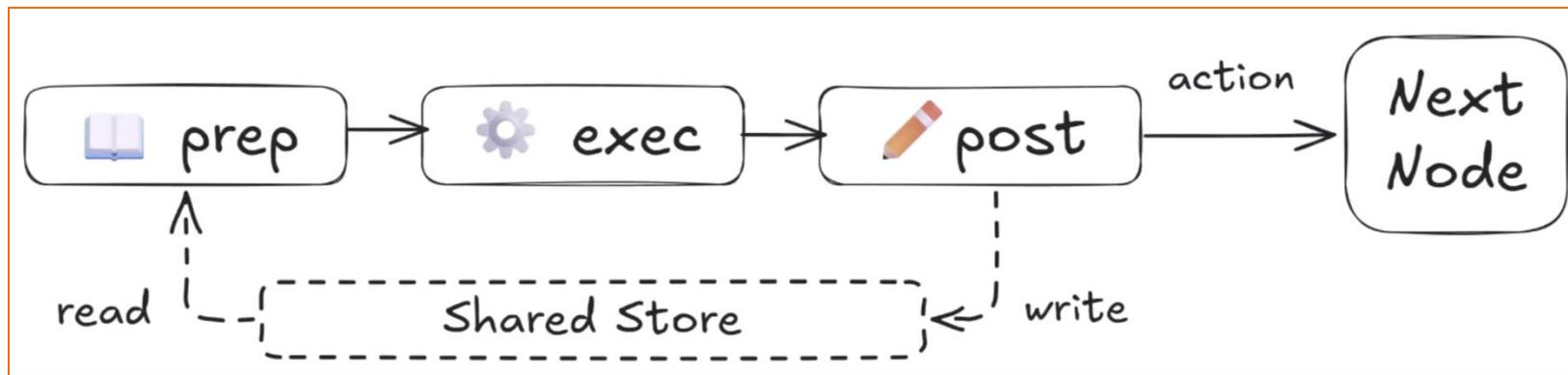
Du definierst Übergänge mit folgender Syntax:

1. Einfacher Standard-Übergang: node_a >> node_b Das bedeutet: Wenn node_a.post() "default" zurückgibt, gehe zu node_b. (Äquivalent zu node_a - "default" >> node_b)

LLM-Framework mit 100 Zeilen Python Code

 Flows: Node A >> Node B >> ...

Ein Flow orchestriert einen Graphen von Nodes. Du kannst Nodes in einer Sequenz verketteten oder Verzweigungen erstellen, abhängig von den Actions, die von jedem Node's post() zurückgegeben werden.



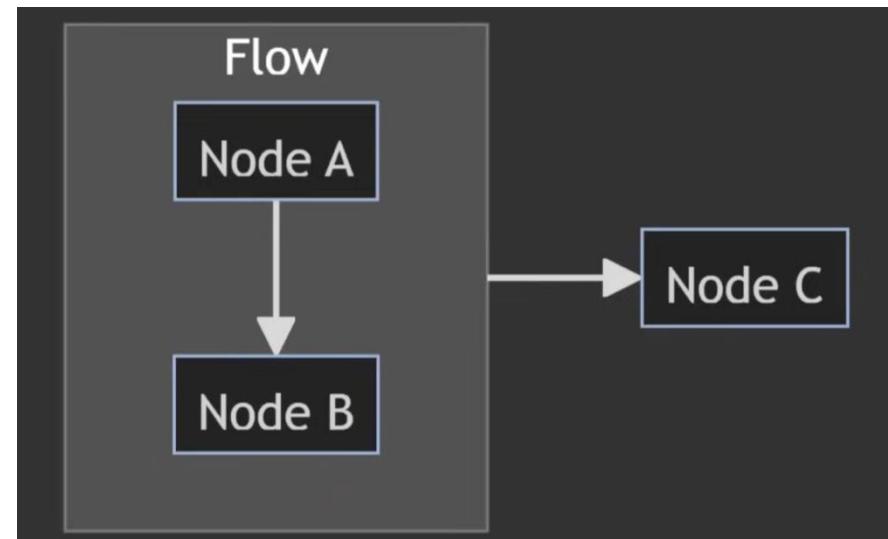
2. Benannter Action-Übergang: node_a - "action_name" >> node_b Das bedeutet: Wenn node_a.post() "action_name" zurückgibt, gehe zu node_b.

Es ist möglich, Schleifen, Verzweigungen oder mehrstufige Flows zu erstellen.

LLM-Framework mit 100 Zeilen Python Code

 Flows = Nodes?

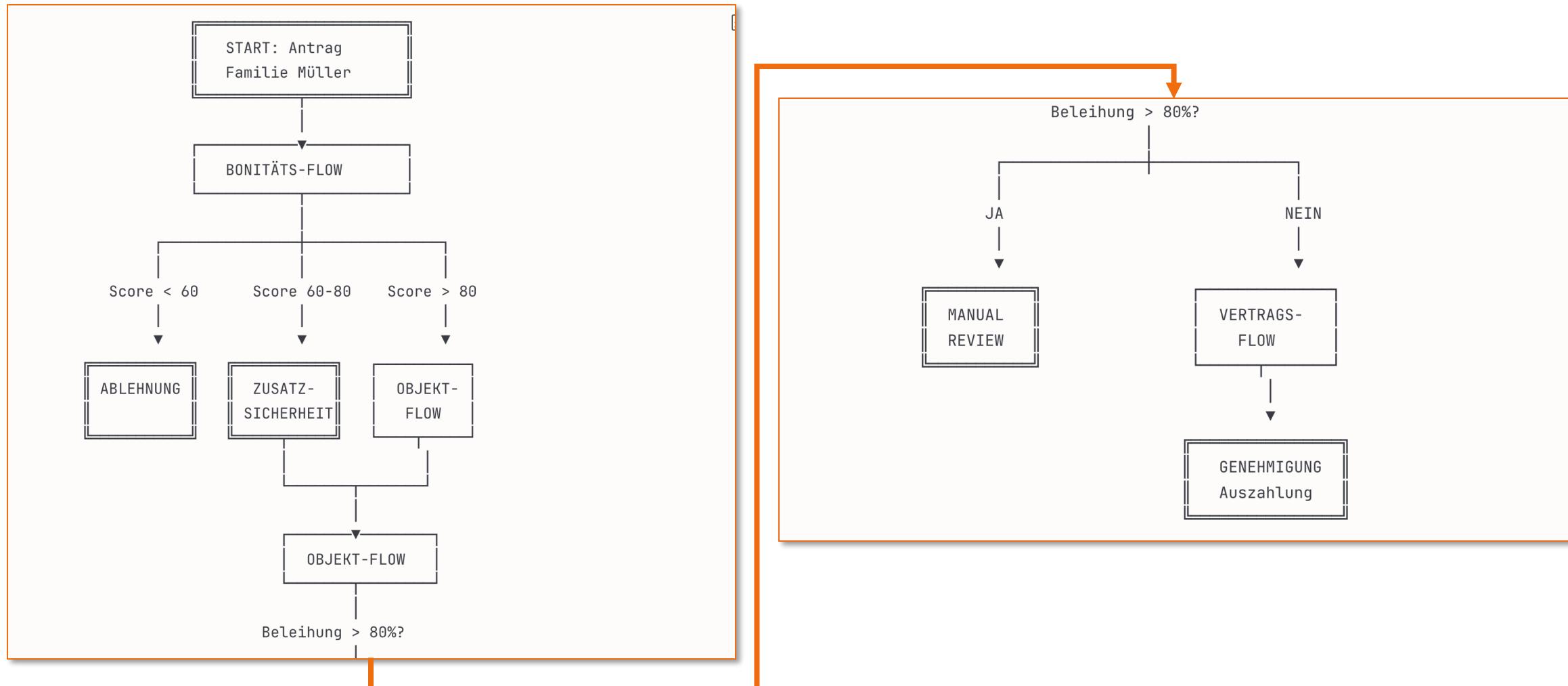
Ein Flow wie hier z.B. Node A >> Node B kann wiederum zu einem Node definiert werden, so dass komplexe Workflows/ Prozesse/ Vorgänge umgesetzt werden können



LLM-Framework mit 100 Zeilen Python Code



Beispiel für einen komplexen Flow



LLM-Framework mit 100 Zeilen Python Code



Beispiel für einen komplexen Flow

Flow	Node	Aufgabe	Output
◆ Bonitäts-Flow	SCHUFA-Check	Bonitätsprüfung	Score 0-100
	Einkommens-Prüfung	Haushaltsrechnung	Verfügbares Einkommen
	Eigenkapital-Check	Vermögensprüfung	EK-Quote
🏠 Objekt-Flow	Grundbuch-Prüfung	Lastenfreiheit prüfen	Grundschulden
	Gutachten-Analyse	Verkehrswert ermitteln	Marktwert in €
	Beleihungswert	Sicherheit berechnen	Max. Kredithöhe
📄 Vertrags-Flow	Konditionen-Rechner	Zinssatz bestimmen	Effektivzins
	Vertrag-Generator	Dokumente erstellen	PDF-Vertrag
	Notar-Termin	Beurkundung planen	Termin-Datum

LLM-Framework mit 100 Zeilen Python Code



Beispiel für einen komplexen Flow

```
# =====
# BAUFINANZIERUNG - Hauptbeispiel
# =====

# Bonitäts-Flow
schufa_check = SchufaCheckNode()
einkommens_pruefung = EinkommensPruefungNode()
eigenkapital_check = EigenkapitalCheckNode()

schufa_check >> einkommens_pruefung >> eigenkapital_check
bonitaets_flow = Flow(start=schufa_check)

# Objekt-Flow
grundbuch_pruefung = GrundbuchPruefungNode()
gutachten_analyse = GutachtenAnalyseNode()
beleihungswert = BeleihungswertNode()

grundbuch_pruefung >> gutachten_analyse >> beleihungswert
objekt_flow = Flow(start=grundbuch_pruefung)

# Vertrags-Flow
konditionen_rechner = KonditionenRechnerNode()
vertrag_generator = VertragGeneratorNode()
notar_termin = NotarTerminNode()

konditionen_rechner >> vertrag_generator >> notar_termin
vertrags_flow = Flow(start=konditionen_rechner)
```

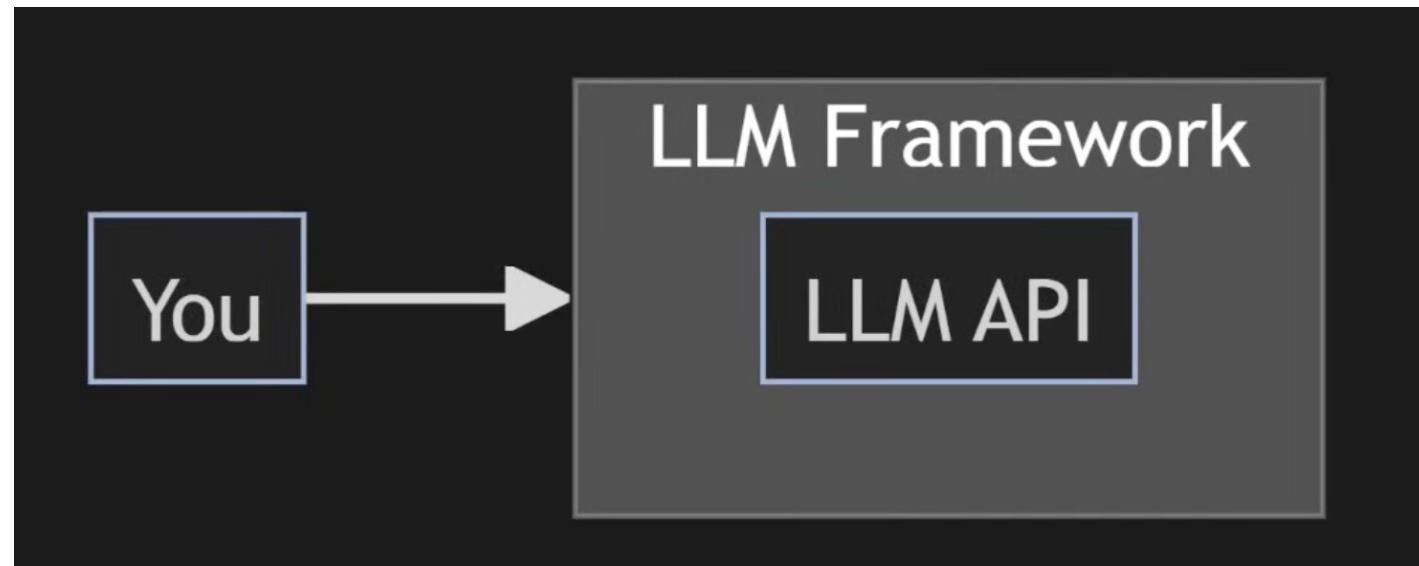
```
# Master-Flow mit Verzweigungen
ablehnung_node = AblehnungNode()
zusatz_sicherheit_node = ZusatzSicherheitNode()

baufi_pipeline = Flow(
    nodes=[bonitaets_flow, objekt_flow, vertrags_flow, ablehnung_node, zusatz_sicherheit_node],
    edges=[
        # Haupt-Pipeline
        bonitaets_flow >> objekt_flow >> vertrags_flow,
        # Alternative Pfade
        bonitaets_flow - "sofort_ablehnung" >> ablehnung_node,
        bonitaets_flow - "zusatz_sicherheit" >> zusatz_sicherheit_node,
        zusatz_sicherheit_node >> objekt_flow
    ]
)
```

LLM-Framework mit 100 Zeilen Python Code

🤔 OK, aber wo ist jetzt das Sprachmodell???

Viele Frameworks kommen mit zahlreichen Wrappers, um LLMs einzubinden



In diesen 100 Zeilen Python Code ist keine Liste an LLMs mit ihren Wrappern integriert, da dies für jedes LLM mit 10-15 Zeilen Code umgesetzt werden kann und diese Zeilen Code öffentlich zugänglich sind.

LLM-Framework mit 100 Zeilen Python Code



OK, aber wo ist jetzt das Sprachmodell???

Wie bindet man jetzt z.B. OpenAI Sprachmodelle ein:

```
from openai import OpenAI
import os

def call_llm(messages):
    client = OpenAI(api_key=os.environ.get("OPENAI_API_KEY"))

    response = client.chat.completions.create(
        model="gpt-4o",
        messages=messages
    )

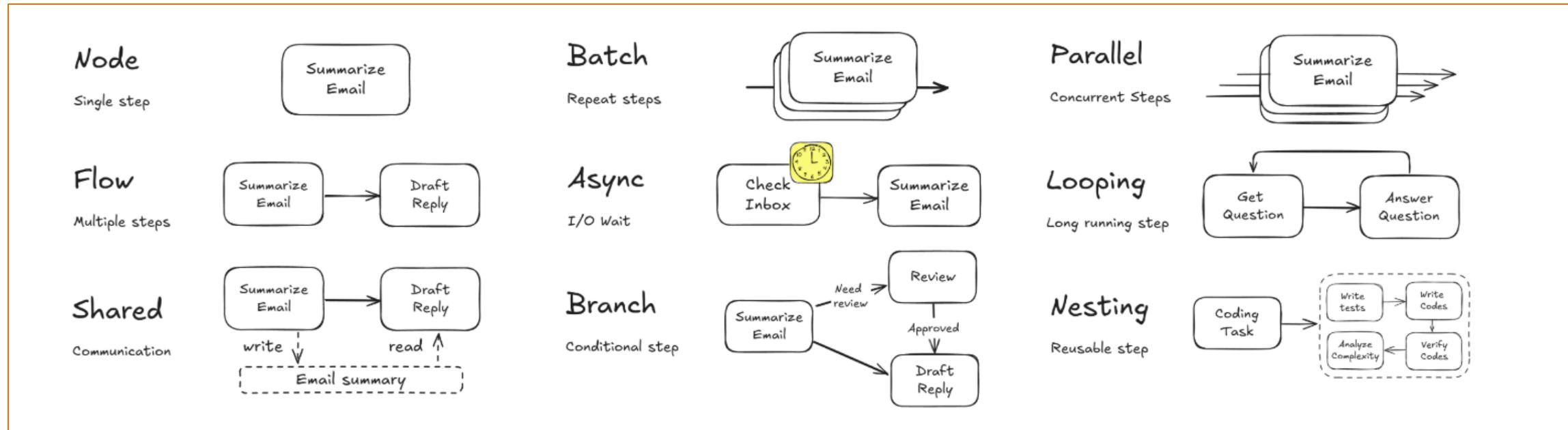
    return response.choices[0].message.content
```

<https://platform.openai.com/docs/quickstart>

LLM-Framework mit 100 Zeilen Python Code



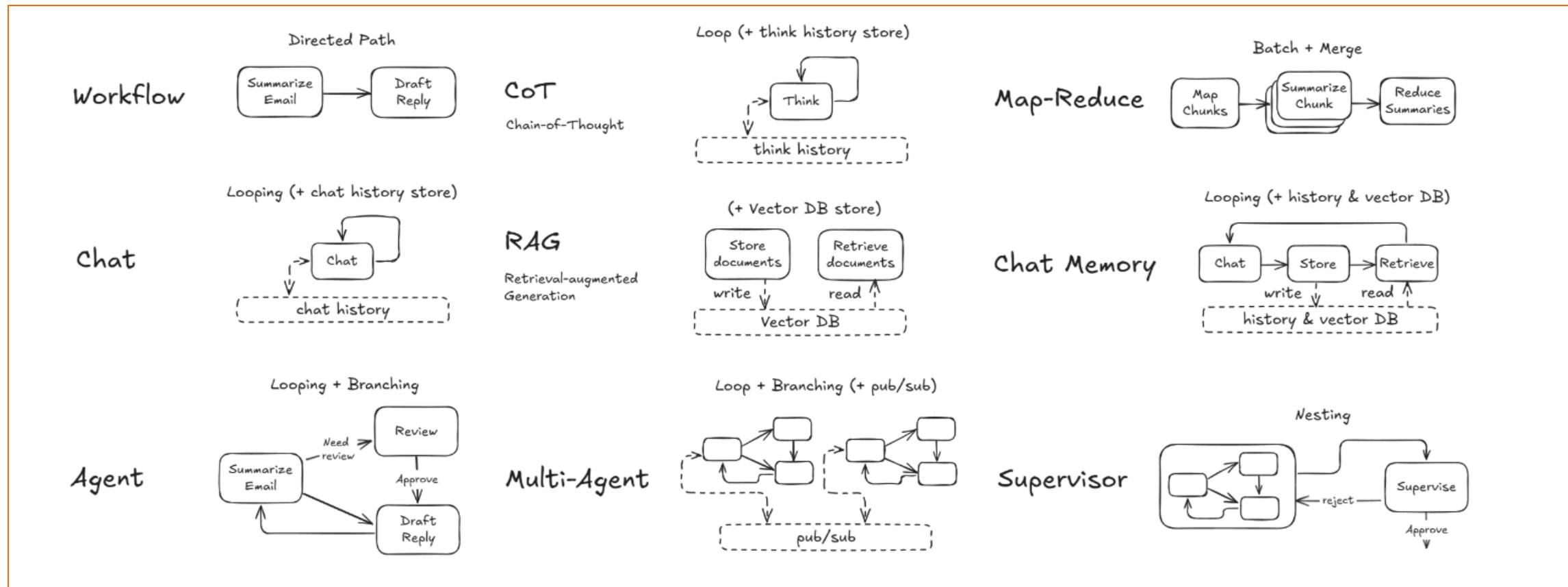
Workflows mit 100 Zeilen Python Code + LLM Integration



LLM-Framework mit 100 Zeilen Python Code



Workflows mit 100 Zeilen Python Code + LLM Integration

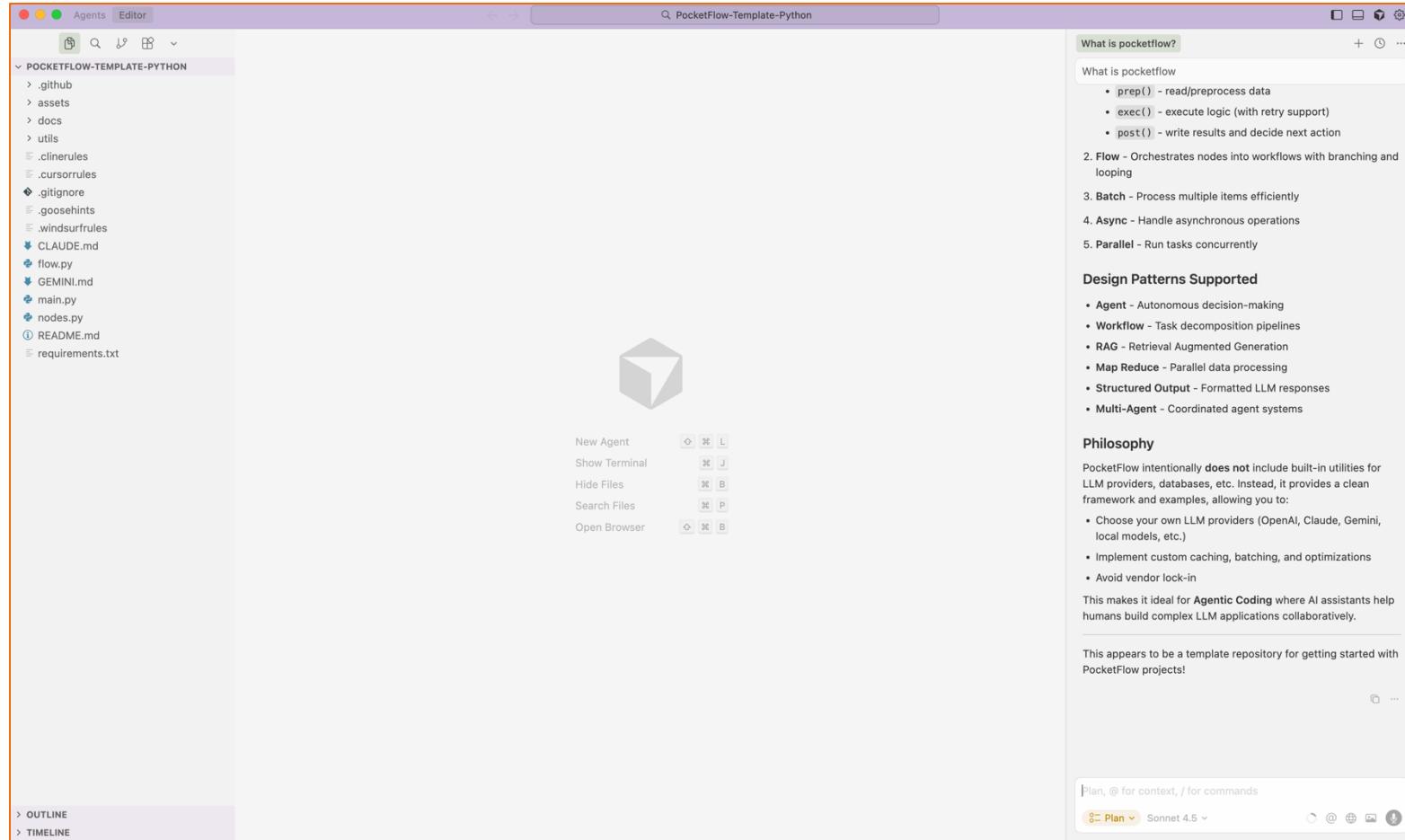


Cursor + 100 Zeilen Python Code

... kurzer Blick auf Cursor + Pocketflow – 100 Zeilen Python Code

LLM-Framework mit 100 Zeilen Python Code

 Cursor: KI-Entwicklung mit 100 Zeilen Code



<https://github.com/The-Pocket/PocketFlow-Template-Python>

Wir erstellen einen „Klon“ des Python Codes für Cursor (und später dann auch für Claude Code).

Cursor und Claude Code: Entwicklungsumgebungen für LLM's, um sozusagen mit “KI” Software, Apps, ... zu entwickeln.

LLM-Framework mit 100 Zeilen Python Code

Cursor: KI-Entwicklung mit 100 Zeilen Code

```
POCKETFLOW-TEMPLATE... .github assets docs utils clinerules .gitignore .goosehints .windsurfrules CLAUDE.md flow.py GEMINI.md main.py nodes.py README.md requirements.txt

.curorrules > .curorrules > ...
1  --
2  layout: default
3  title: "Agentic Coding"
4  --
5
6  # Agentic Coding: Humans Design, Agents code!
7
8  > If you are an AI agent involved in building LLM Systems, read this guide **VERY, VERY** carefully! This is the most important chapter in the entire document. Throughout development, you should always (1) start with a small and simple solution, (2) design at a high level ('docs/design.md') before implementation, and (3) frequently ask humans for feedback and clarification.
9  { .warning }
10
11  ## Agentic Coding Steps
12
13  Agentic Coding should be a collaboration between Human System Design and Agent Implementation:
14
15  | Steps | Human | AI | Comment |
16  |:-----|:-----|:-----|:-----|
17  | 1. Requirements | *** High | ** Low | Humans understand the requirements and context. |
18  | 2. Flow | ** Medium | ** Medium | Humans specify the high-level design, and the AI fills in the details. |
19  | 3. Utilities | ** Medium | ** Medium | Humans provide available external APIs and integrations, and the AI helps with implementation. |
20  | 4. Data | ** Low | *** High | AI designs the data schema, and humans verify. |
21  | 5. Node | ** Low | *** High | The AI helps design the node based on the flow. |
22  | 6. Implementation | ** Low | *** High | The AI implements the flow based on the design. |
23  | 7. Optimization | ** Medium | ** Medium | Humans evaluate the results, and the AI helps optimize. |
24  | 8. Reliability | ** Low | *** High | The AI writes test cases and addresses corner cases. |
25
26  1. **Requirements**: Clarify the requirements for your project, and evaluate whether an AI system is a good fit.
27  - Understand AI systems' strengths and limitations:
28  - **Good for**: Routine tasks requiring common sense (filling forms, replying to emails)
29  - **Good for**: Creative tasks with well-defined inputs (building slides, writing SQL)
30  - **Not good for**: Ambiguous problems requiring complex decision-making (business strategy, startup planning)
31  - **Keep It User-Centric**: Explain the "problem" from the user's perspective rather than just listing features.
32  - **Balance complexity vs. impact**: Aim to deliver the highest value features with minimal complexity early.
33
34  2. **Flow Design**: Outline at a high level, describe how your AI system orchestrates nodes.
35  - Identify applicable design patterns (e.g., [Map Reduce](./design_pattern/mapreduce.md), [Agent](./design_pattern/agent.md), [RAG](./design_pattern/rag.md)).
36  - For each node in the flow, start with a high-level one-line description of what it does.
37  - If using **Map Reduce**, specify how to map (what to split) and how to reduce (how to combine).
38  - If using **Agent**, specify what are the inputs (context) and what are the possible actions.
39  - If using **RAG**, specify what to embed, noting that there's usually both offline (indexing) and online (retrieval) workflows.
40  - Outline the flow and draw it in a mermaid diagram. For example:
41  ````mermaid
42  flowchart LR
43  start[Start] --> batch[Batch]
44  batch --> check[Check]
45  check -->|OK| process[Process]
46  check -->|Error| fix[Fix]
47  fix --> check
```

Mit dem Klon des Github Repos wurde zusätzlich eine .curorrules Datei angelegt, in der die gesamte „Anleitung“ zur Nutzung von „Pocketflow“ – den 100 Zeilen Python Code mit integriert ist.

Cursor + Pocketflow (100 Zeilen Python Code – Node, Graph, Flow)



Cursor: Youtube Summarizer „coden“

Unser erster Prompt für Cursor:

„Hey Cursor AI, hilf mir ein Projekt zu erstellen, das YouTube-Videos zusammenfasst:
Es nimmt einen YouTube-Link als Eingabe.

Es extrahiert interessante Themen und generiert auch Fragen und Antworten für jedes Thema.

Es erklärt all das oben Genannte auf eine sehr freundliche Art, als wären wir KI-Anfänger und berücksichtigt dabei, dass wir für eine Volksbank arbeiten im Marketing und Vertrieb und in der IT-Organisation.

Generiere eine HTML-Seite zur Visualisierung der Zusammenfassung.

Lies zuerst die Dokumentation sehr sorgfältig. Dann beginne nur mit Anforderungen und Hilfsfunktionen. Höre dort auf“

Cursor + Pocketflow (100 Zeilen Python Code – Node, Graph, Flow)



Cursor: Youtube Summarizer „coden“

The screenshot shows the Cursor AI application interface. On the left, there is a file tree for a project named "POCKETFLOW-TEMPLATE-PYTHON". The file ".cursorrules" is selected and highlighted in green. The file tree also includes ".github", ".assets", ".docs", ".util", ".clinerules", ".gitignore", ".goosehints", ".windsurfrules", "CLAUDE.md", "flow.py", "GEMINI.md", "main.py", "nodes.py", "README.md", and "requirements.txt". At the bottom of the file tree, there are buttons for "OUTLINE" and "TIMELINE". In the center, there is a large workspace area with a cube icon and a terminal-like interface with keyboard shortcuts for "New Agent", "Show Terminal", "Hide Files", "Search Files", and "Open Browser". On the right, there is a "New Chat" window titled "Hey Cursor AI, hilf mir ein Projekt zu erstellen, das YouTube-Videos zusammenfasst: Es nimmt einen YouTube-Link als Eingabe." Below this, it says "Es extrahiert interessante Themen und generiert auch Fragen und Antworten für jedes Thema." and "Es erklärt all das oben Genannte auf eine sehr freundliche Art, als wären wir KI-Anfänger und berücksichtigt dabei, dass wir für eine Volksbank arbeiten im Marketing und Vertrieb und in der IT-Organisation." The "Plan" tab is selected in the chat window. At the bottom of the screen, there is a "Cursor Tab" button.

Wir haben den Prompt hier eingegeben und den Modus in Cursor auf „**Plan**“ gestellt, damit noch kein Code generiert wird.

Cursor + Pocketflow (100 Zeilen Python Code – Node, Graph, Flow)



Cursor: Youtube Summarizer „coden“

The screenshot shows the Pocketflow interface for a YouTube video summarization project. The sidebar lists various files and folders. The main content area is titled "YouTube Video Zusammenfassung - Requirements & Utility Functions". It contains sections for Requirements, Benutzer-Story, Funktionaler Anforderungen (with steps 1-4), Nicht-funktionaler Anforderungen, and Utility Functions. The right panel displays a plan titled "Projekt zur Zusammenfassung von YouTube-Videos" with a message from Claude Sonnet 4.5, a "Thought for 5s" section, and a "To-dos" list.

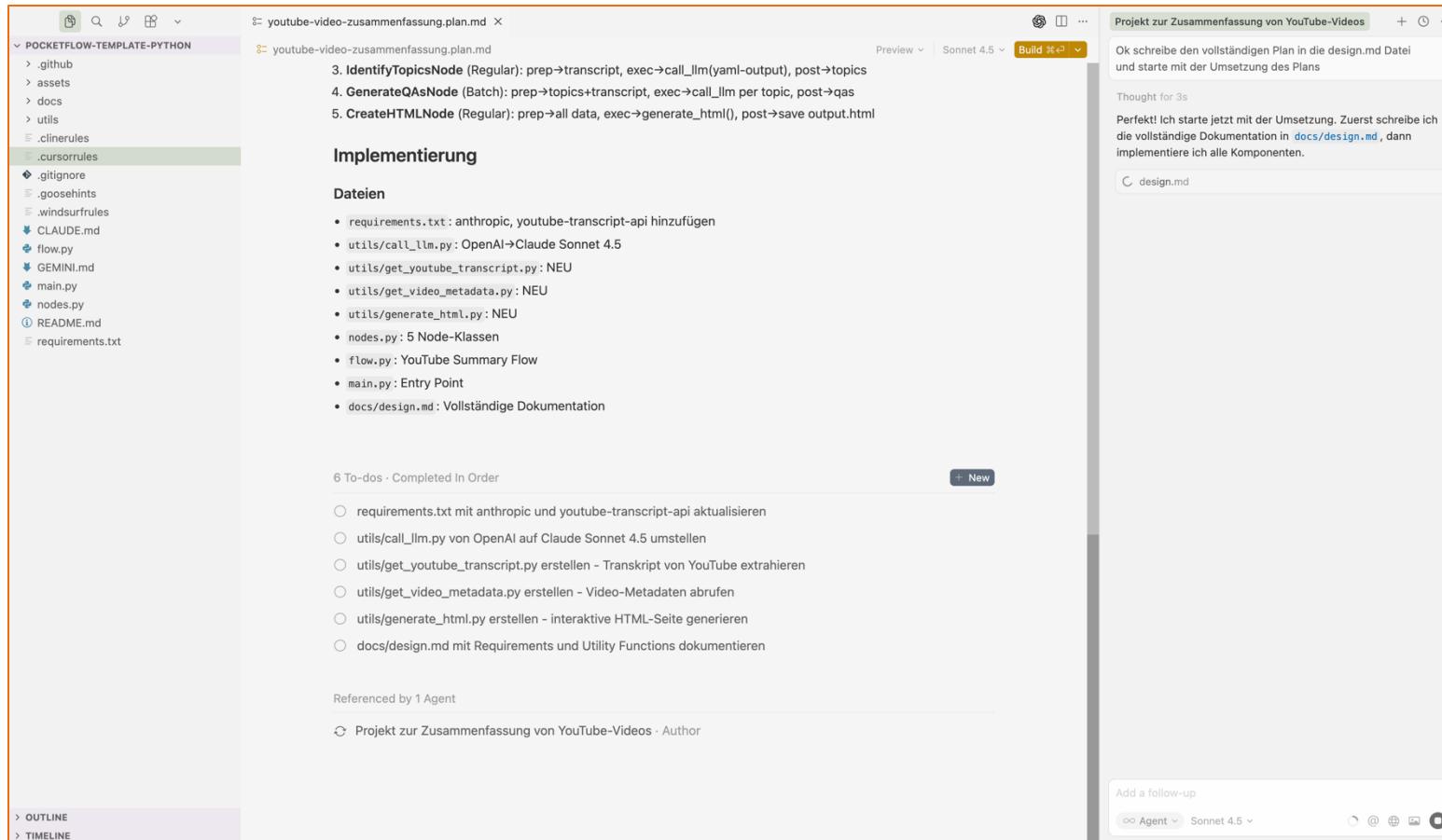
Cursor plant jetzt die Aufgabe Schritt für Schritt.

Wir können interaktiv mit Cursor den Plan anpassen, überarbeiten usw.

Nach diesen Iterationen lassen wir dann den Plan ausführen und über Cursor Code schreiben.

Cursor + Pocketflow (100 Zeilen Python Code – Node, Graph, Flow)

Cursor: Youtube Summarizer „coden“



Nach der kompletten Planung des Projekts lassen wir diesen Plan in eine md-Datei schreiben (Markdown-Format).

Dann wechseln wir auf den **Agenten Modus**, so dass Cusor „**Code**“ schreiben kann und los geht's mit der KI-Entwicklung.

Cursor + Pocketflow (100 Zeilen Python Code – Node, Graph, Flow)



Cursor: Youtube Summarizer „coden“

The screenshot shows the Pocketflow interface with a flow titled "YouTube Video Zusammenfassungs-Tool". The flow consists of several nodes connected by arrows, forming a process. The nodes include:

- Input node: "YouTube Video Zusammenfassungs-Tool" (with a file icon)
- Text node: "Ein automatisches Tool zur Erstellung freundlicher, interaktiver Zusammenfassungen von YouTube-Videos - speziell entwickelt für Volksbank-Mitarbeiter in Marketing, Vertrieb und IT ohne KI-Vorkenntnisse."
- Text node: "Erstellt mit [PocketFlow] (<https://github.com/The-Pocket/PocketFlow>) und Claude Sonnet 4.5."
- Text node: "## Features
 - ✅ **Automatische Transkript-Extraktion** von YouTube-Videos
 - ✅ **Freundliche Zusammenfassungen** in einfacher, verständlicher Sprache
 - ✅ **Themenidentifikation** - erkennt 3-5 Hauptthemen
 - ✅ **Q&A-Generierung** - 2-3 relevante Fragen & Antworten pro Thema
 - ✅ **Interaktive HTML-Visualisierung** mit modernem Design und klappbaren Sektionen
- Text node: "## Voraussetzungen
 - Python 3.8 oder höher
 - Anthropic API Key (Claude Sonnet 4.5)
 - Internetverbindung
- Text node: "## Installation
 - 1. **Repository klonen oder herunterladen**
 - 2. **Dependencies installieren**
```bash  
pip install -r requirements.txt
  - 3. \*\*Anthropic API Key setzen\*\*  
Erstelle einen API Key bei [Anthropic] (<https://console.anthropic.com/>) und setze ihn als Umgebungsvariable:  
\*\*macOS/Linux:\*\*  
```bash  
export ANTHROPIC_API_KEY="dein-api-key-hier"
Windows:
```cmd  
set ANTHROPIC\_API\_KEY=dein-api-key-hier
- Text node: "## Verwendung
  - ### Einfache Verwendung  
```bash  
python main.py "https://www.youtube.com/watch?v=VIDEO_ID"
 - ### Interaktive Verwendung

On the right side of the interface, there is a sidebar with the following sections:

- API Key setzen: "Ok schreibe den vollständigen Plan in die design.md Datei und starte mit der Umsetzung des Plans", "pip install -r requirements.txt", "export ANTHROPIC_API_KEY="dein-api-key"
- Features: A list of checked features including automatic transcript extraction, friendly summaries, topic identification, Q&A generation, and interactive HTML visualization.
- Project Status: "Das Projekt ist produktionsbereit und folgt allen PocketFlow Best Practices!"
- File Changes: "10 Files Edited Review", listing changes for design.md, requirements.txt, call_llm.py, get_youtube_transcript.py, get_video_metadata.py, generate_html.py, nodes.py, flow.py, and main.py.
- Bottom status bar: "10 Files", "Undo All", "Keep All", "Plan, @ for context, / for commands", "Agent", "Sonnet 4.5", and other icons.

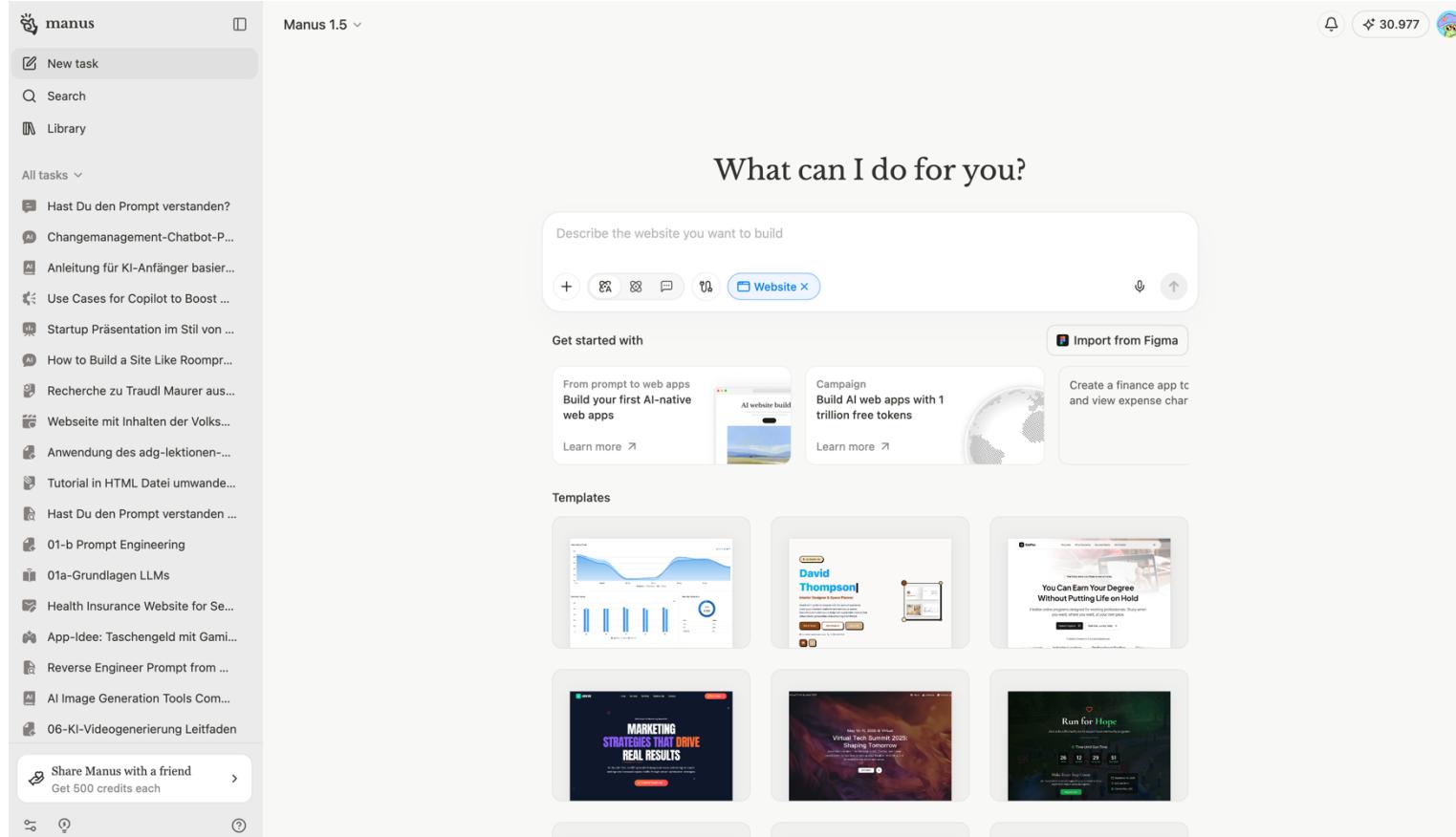
Nach einigen Minuten ist Cursor fertig und erstellt uns eine kleine Zusammenfassung mit dem Ergebnis plus einer Readme.md Datei, die uns die Nutzung des Codes erklärt.

KI-Landkarte

Manus, Lovable, Kimi, Dyad, ..., Claude Code

All-in-One Wrapper

 Manus, Lovable, Kimi, Dyad, ...



<https://manus.im/app>

All-in-One Wrapper



Manus, Lovable, Kimi, Dyad, ...

The screenshot shows the Manus AI web app landing page. At the top, there's a navigation bar with links for Features, Resources, Events, Pricing, and a language selector set to English. A prominent 'Get started' button is also visible. The main headline reads 'Build AI web apps with 1 trillion free tokens'. Below it, a subtext states 'A shared pool of 1 trillion tokens for all the AI features your web apps need.' Two buttons are present: 'Build now' and 'Explore: Manus web app'. A progress bar indicates 'Used 177,630,289,627 / 1,000,000,000,000 LLM Tokens'. The bar is mostly blue with a small white segment at the beginning. The text 'Last updated November 6, 2025 - 5:00 AM PST' is shown below the bar. A friendly reminder note is at the bottom: 'Friendly reminder: In this campaign, "tokens" refer to free AI usage within the Manus web app, not your Manus account credits. Building website will still consume credits from your account.' The background features a dark globe graphic.

<https://manus.im/app>

Manus-App:

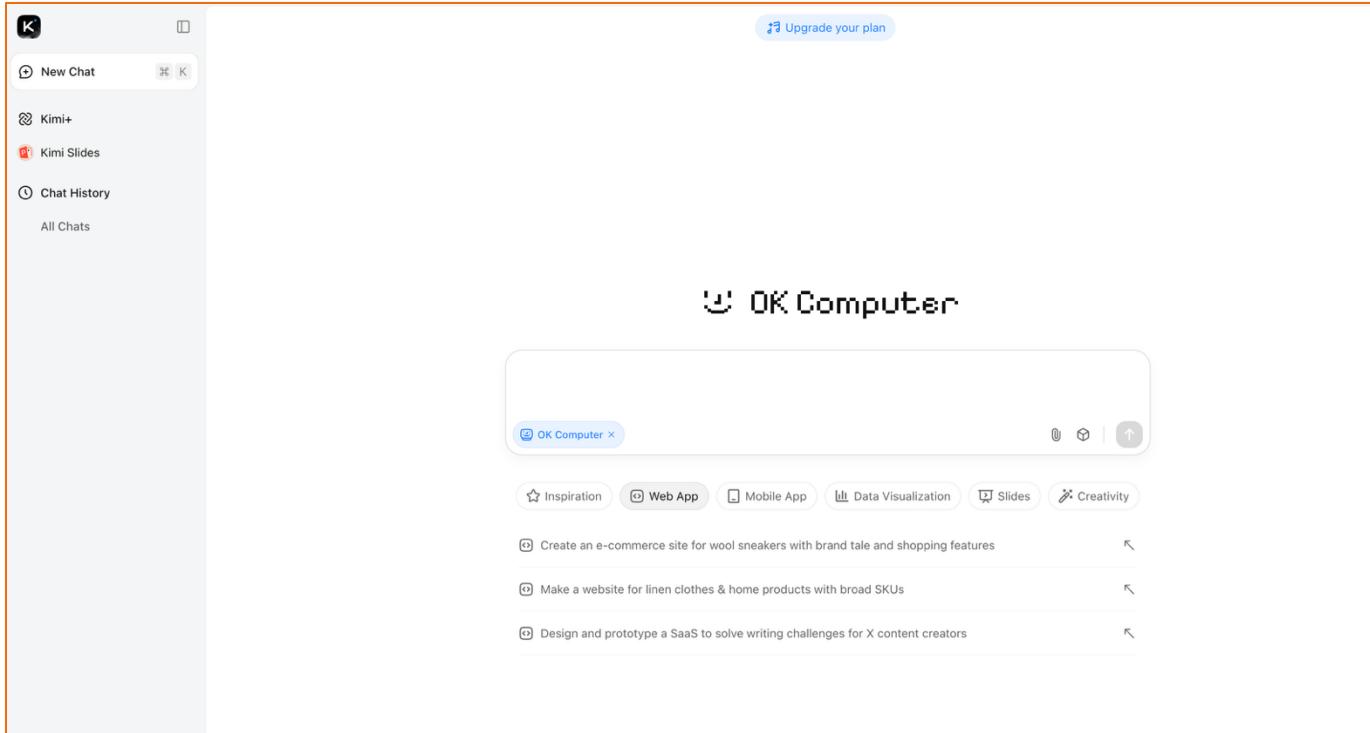
Wrapper Anwendungen und native Apps bieten diverse Funktionen, unter anderem auch KI-Software-Entwicklung.

Über API Keys können dann die Sprachmodelle für die KI-Entwicklung ausgewählt werden und auch lokale Modelle (Ollama, LM-Studio, ...)

All-in-One Wrapper



Manus, Lovable, Kimi, Dyad, ...



<https://www.kimi.com/>

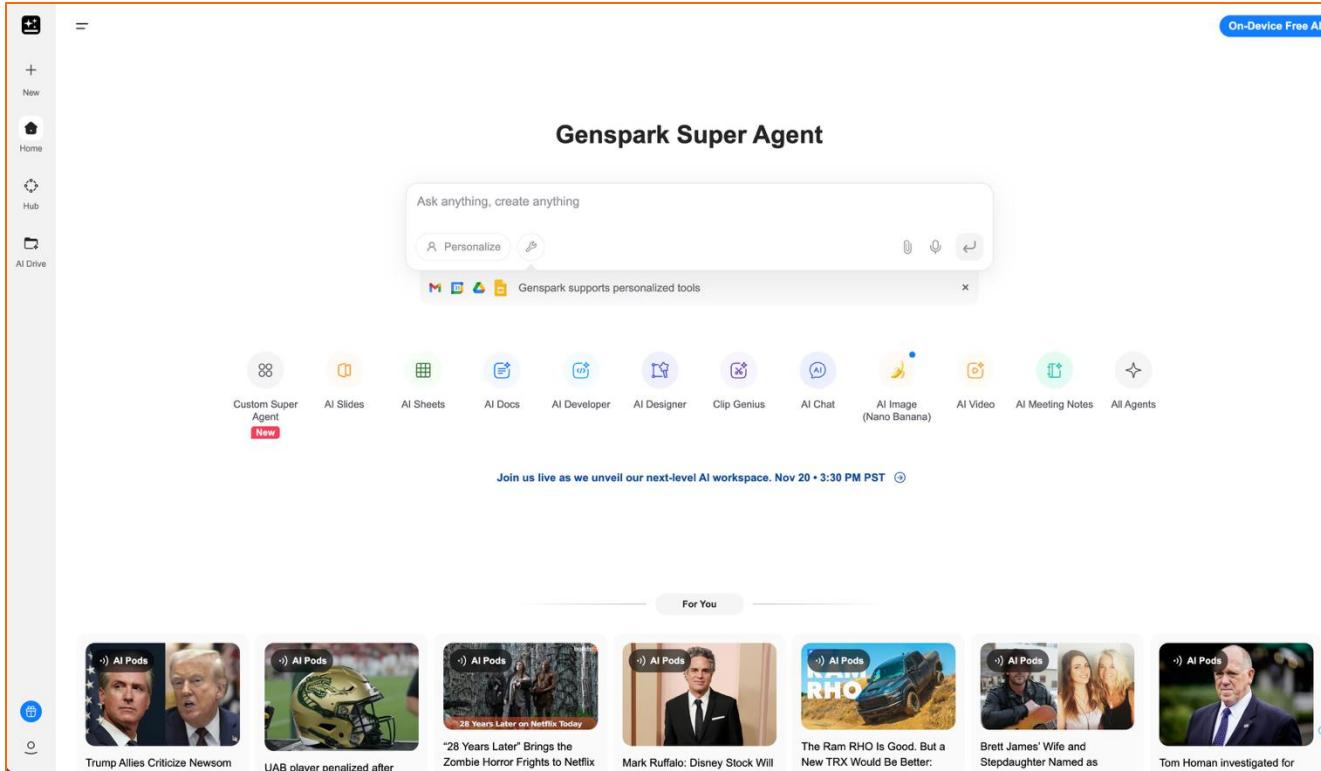
Kimi/ Kimi-2 Web App:

Wrapper Anwendungen und native Apps bieten diverse Funktionen, unter anderem auch KI-Software-Entwicklung.

Über API Keys können dann die Sprachmodelle für die KI-Entwicklung ausgewählt werden und auch lokale Modelle (Ollama, LM-Studio, ...)

All-in-One Wrapper

Manus, Lovable, Kimi, Dyad, ...



<https://www.genspark.ai/>

All-in-One Wrapper



Manus, Lovable, Kimi, Dyad, ...

The screenshot shows the Lovable web application interface. At the top, there is a navigation bar with the Lovable logo, Community, Pricing, and Enterprise links. On the right side of the header, there are icons for a gift, an envelope, and a user profile labeled "rakidakiXYZ". The main area has a blue-to-orange gradient background. In the center, the text "Build something ❤️ Lovable" is displayed, followed by "Create apps and websites by chatting with AI". Below this is a white input field containing the placeholder "Ask Lovable to create an internal tool that...". At the bottom left, there is a user profile for "rakidakiXYZ" and a "View all >" button. The entire screenshot is enclosed in a red border.

<https://lovable.dev/>

Lovable Web-App:

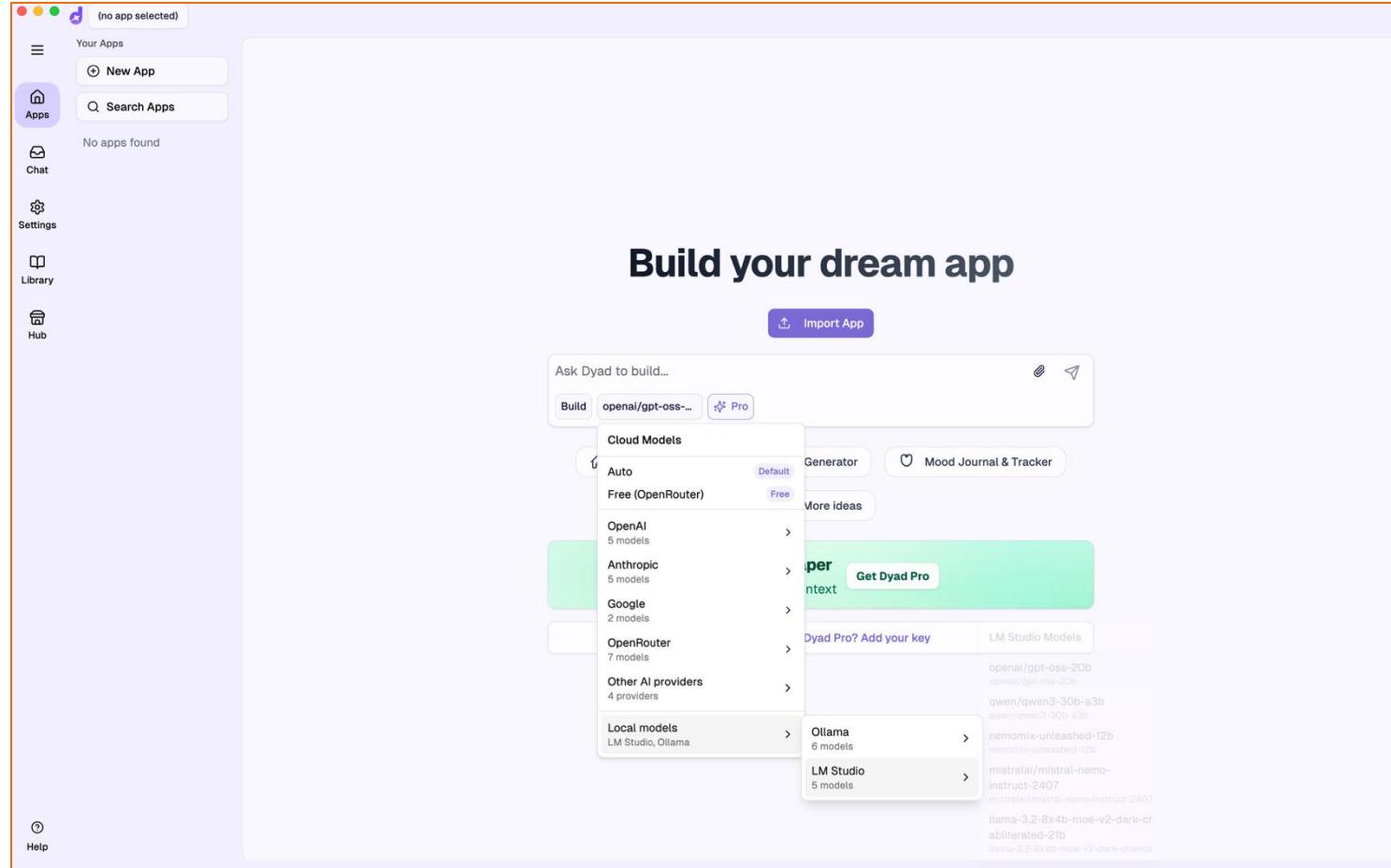
Wrapper Anwendungen und native Apps bieten diverse Funktionen, unter anderem auch KI-Software-Entwicklung.

Über API Keys können dann die Sprachmodelle für die KI-Entwicklung ausgewählt werden und auch lokale Modelle (Ollama, LM-Studio, ...)

All-in-One Wrapper



Manus, Lovable, Kimi, Dyad, ...



<https://www.dyad.sh/>

DYAD-App:

Wrapper Anwendungen und native Apps bieten diverse Funktionen, unter anderem auch KI-Software-Entwicklung.

Über API Keys können dann die Sprachmodelle für die KI-Entwicklung ausgewählt werden und auch lokale Modelle (Ollama, LM-Studio, ...)

Tech-Stacks, IT-Infrastruktur, Anforderungen, ...

Kurzer Überblick

Vibe Coding



Typische Produkte, die eingesetzt werden

<https://www.inngest.com/>

Local Debugging Tool, Warteschleifen, Fehlermanagement ...

<https://polar.sh/>

Anstelle von Stripe für Zahlungen von Kunden

<https://posthog.com/>

OpenSource Analytics

<https://nextjs.org/>

Framework to build Apps, SaaS, ...

<https://tailwindcss.com/> <https://ui.shadcn.com/> <https://daisyui.com/>

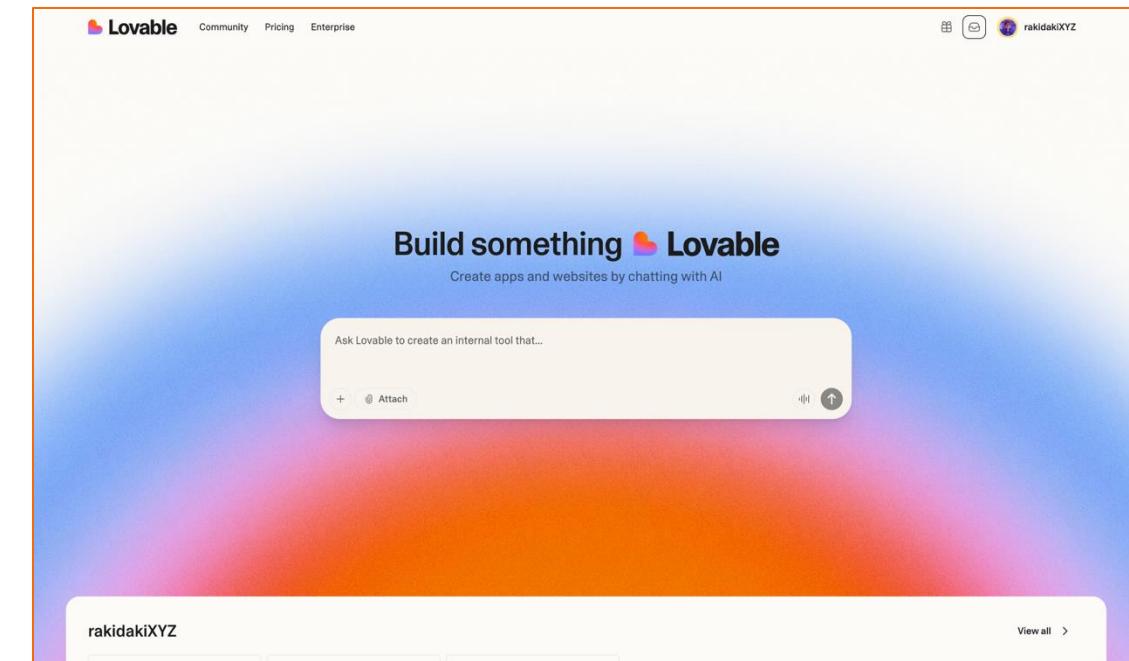
ShadCN Daisy UI, Tailwind CSS for the design System

<https://www.better-auth.com/>

User Authentication

<https://www.postgresql.org/>

Datenbank PostgreSQL (oder auch Supabase, Qdrant, ...)



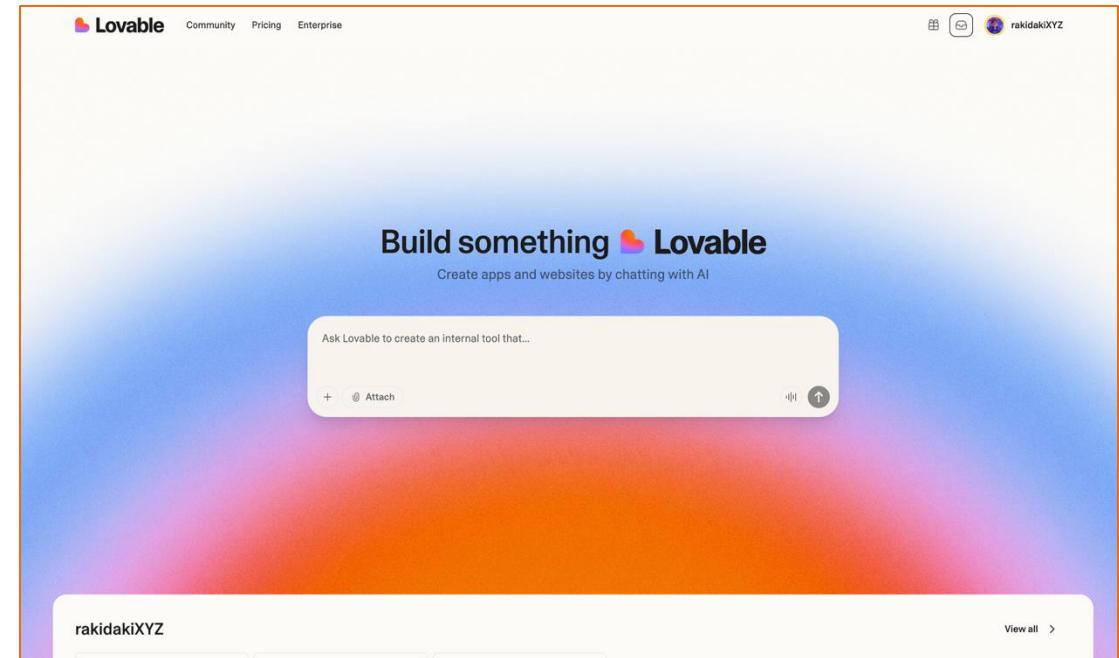
Vibe Coding

🌐 Webseite - Anforderungen

<https://openrouter.ai/> und <https://ai-sdk.dev/>

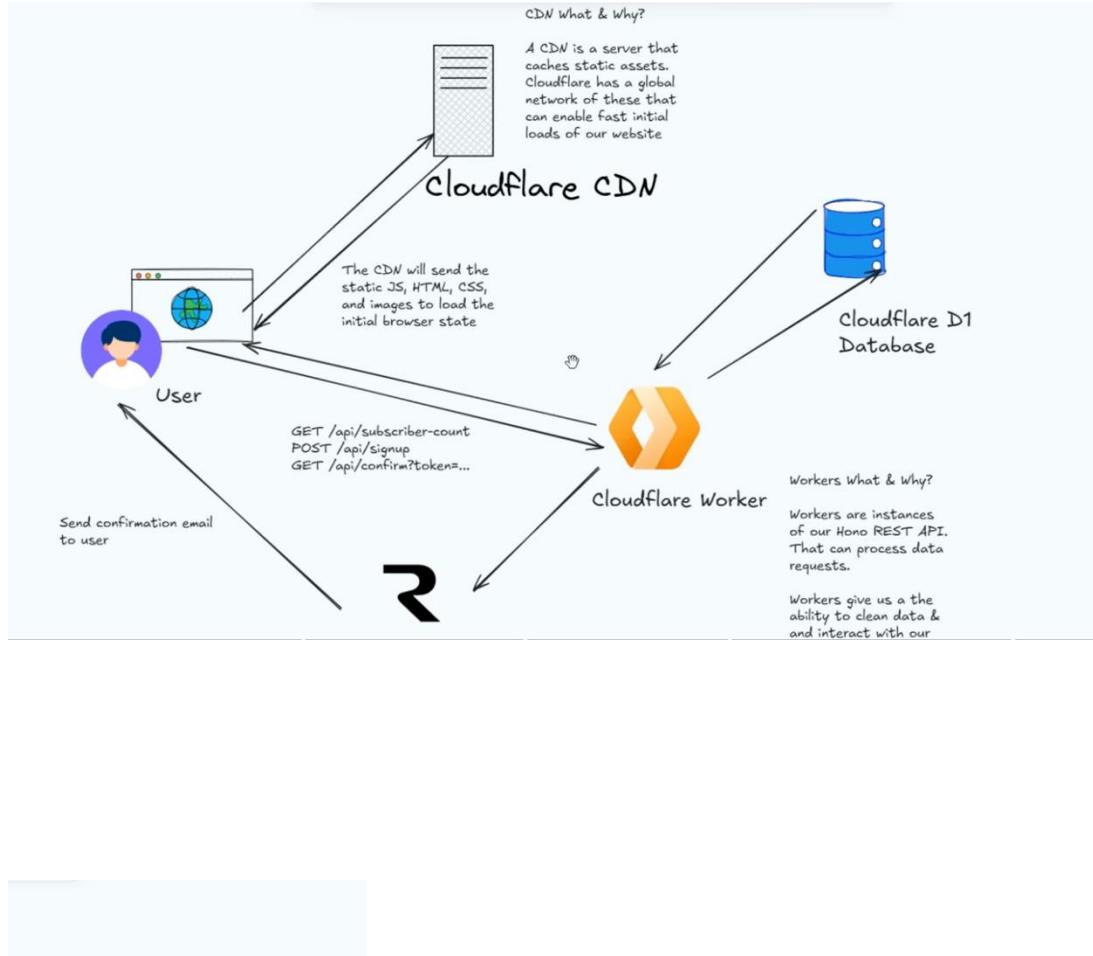
Integration von KI, LLMs, ...

und natürlich noch viel mehr



Claude Code – Senior Entwickler

Beispiel Infrastruktur



Cloudflare CDN,

Ein Content Delivery Network (CDN) sorgt dafür, dass statische Ressourcen wie HTML, CSS und JavaScript weltweit mit hoher Geschwindigkeit ausgeliefert werden.

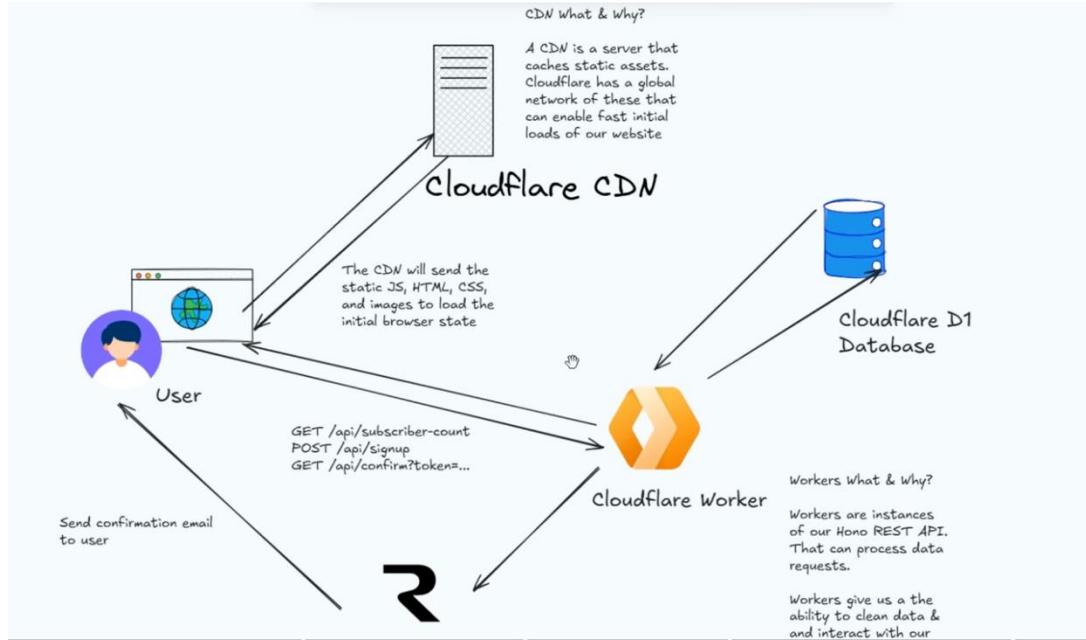
Cloudflare CDN verbessert die Ladezeiten, Verfügbarkeit und Sicherheit unserer Anwendung, indem Inhalte über ein globales Netzwerk von Edge-Servern bereitgestellt werden.

Cloudflare Workers

Verwendung als „Wächter“ unserer API:
Cloudflare Workers dienen als Sicherheits- und Ausführungsschicht, um unsere Hono API zu hosten. Sie schützen unsere Datenbank vor unbefugten Zugriffen und ermöglichen **skalierbare serverlose Funktionen** direkt am Edge.

Claude Code – Senior Entwickler

Beispiel Infrastruktur



Cloudflare D1

Datenbankdesign:

Wir entwerfen ein SQLite-Schema für eine Tabelle `subscribers`, die Abonnenteninformationen speichert. Diese D1-Datenbank ist leichtgewichtig, serverlos und perfekt für unsere Anforderungen.

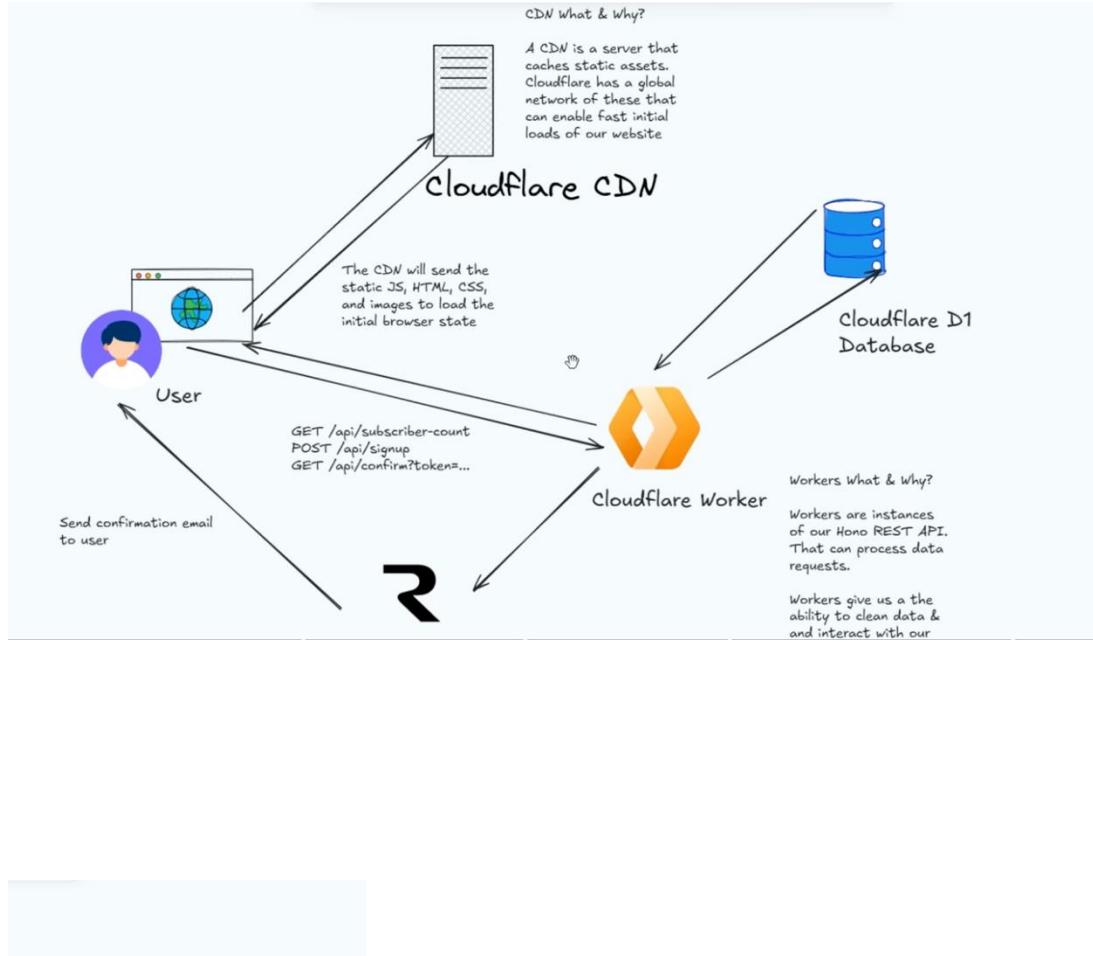
API-Endpunkte

Design der Hono API-Routen:

- `GET /api/subscriber-count` – Gibt die aktuelle Anzahl der Abonnenten zurück.
- `POST /api/signup` – Registriert einen neuen Abonnenten.
- `GET /api/confirm` – Bestätigt die E-Mail-Adresse im Double-Opt-in-Verfahren.

Claude Code – Senior Entwickler

Beispiel Infrastruktur



Sicherheit & Validierung

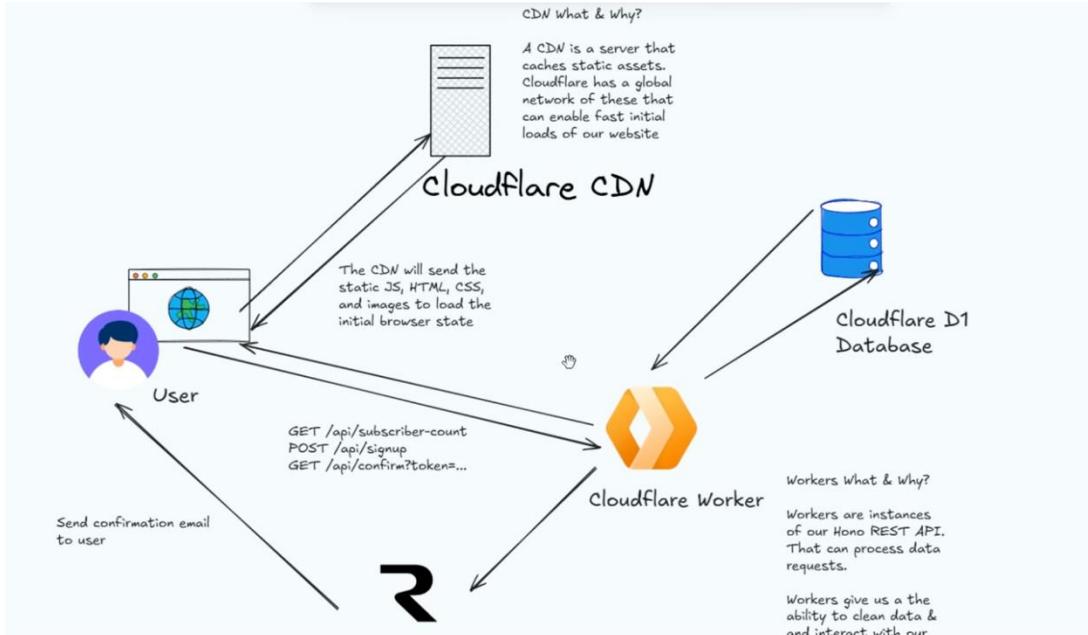
- Implementierung eines sicheren Double-Opt-in-Flows mithilfe von Resend
- Rate Limiting, um Spam zu verhindern.
- Verwendung von Zod zur E-Mail-Validierung.
- Umgang mit „Plus Addressing“, um doppelte Anmeldungen zu vermeiden (z. B. `user+test@gmail.com`).

Frontend UX

- Astro für Static Site Generation (SSG)
- React für dynamische „Islands“.
- Fokus auf Ladezustände, Fehlerbehandlung und Erfolgsmeldungen, um ein reibungsloses Nutzererlebnis zu schaffen.

Claude Code – Senior Entwickler

Beispiel Infrastruktur



Datenbank-Philosophie

Eine einzelne D1-Datenbank (ohne Read-Replicas) ist ideal für diesen Anwendungsfall.
Sie bietet die perfekte Balance aus Einfachheit, Zuverlässigkeit und Leistung – ohne unnötige Over-Engineering-Komplexität

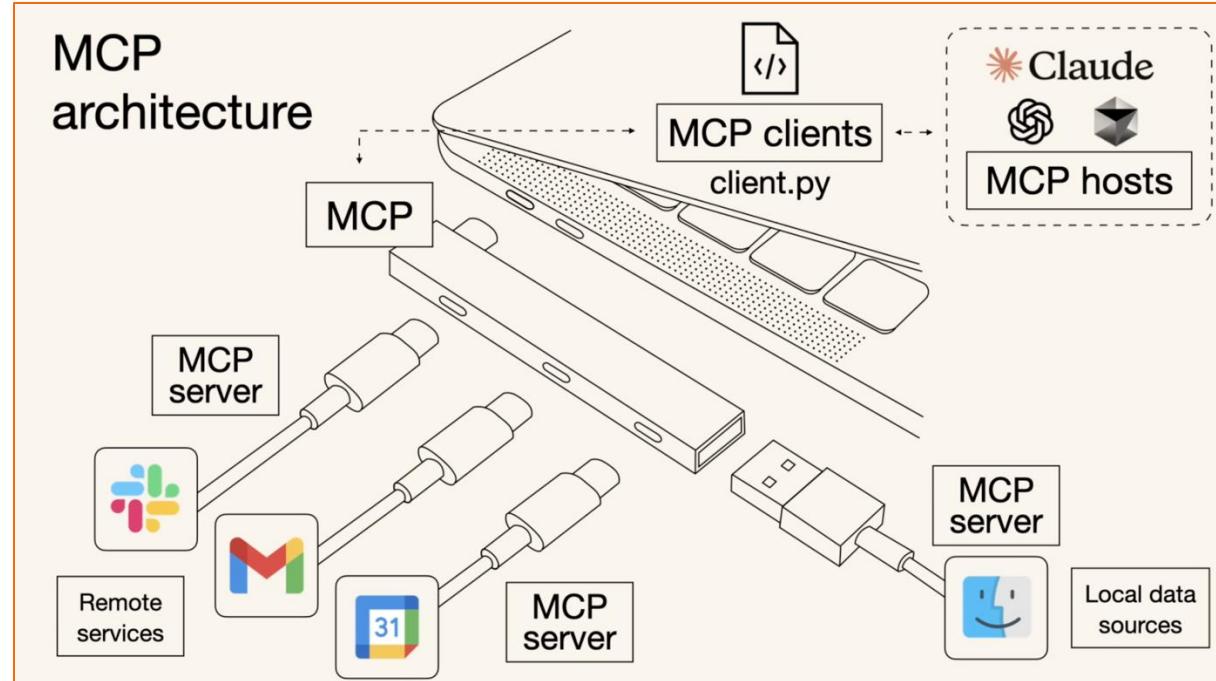
Claude Code als Seniorentwickler

Entwicklungsteam in Claude Code mit „Agenten, Commands, Templates, Memory, MCP, ...“

Claude Code



Wissen:Context7 MCP – aktuelle Infos zu Tech-Stacks ...



Context7

Sign in

Playground MCP Server + Add

Up-to-date Docs for LLMs and AI code editors

Copy latest docs & code — paste into Cursor, Claude, or other LLMs

Search a library (e.g. Next, React)

Popular Trending Recent

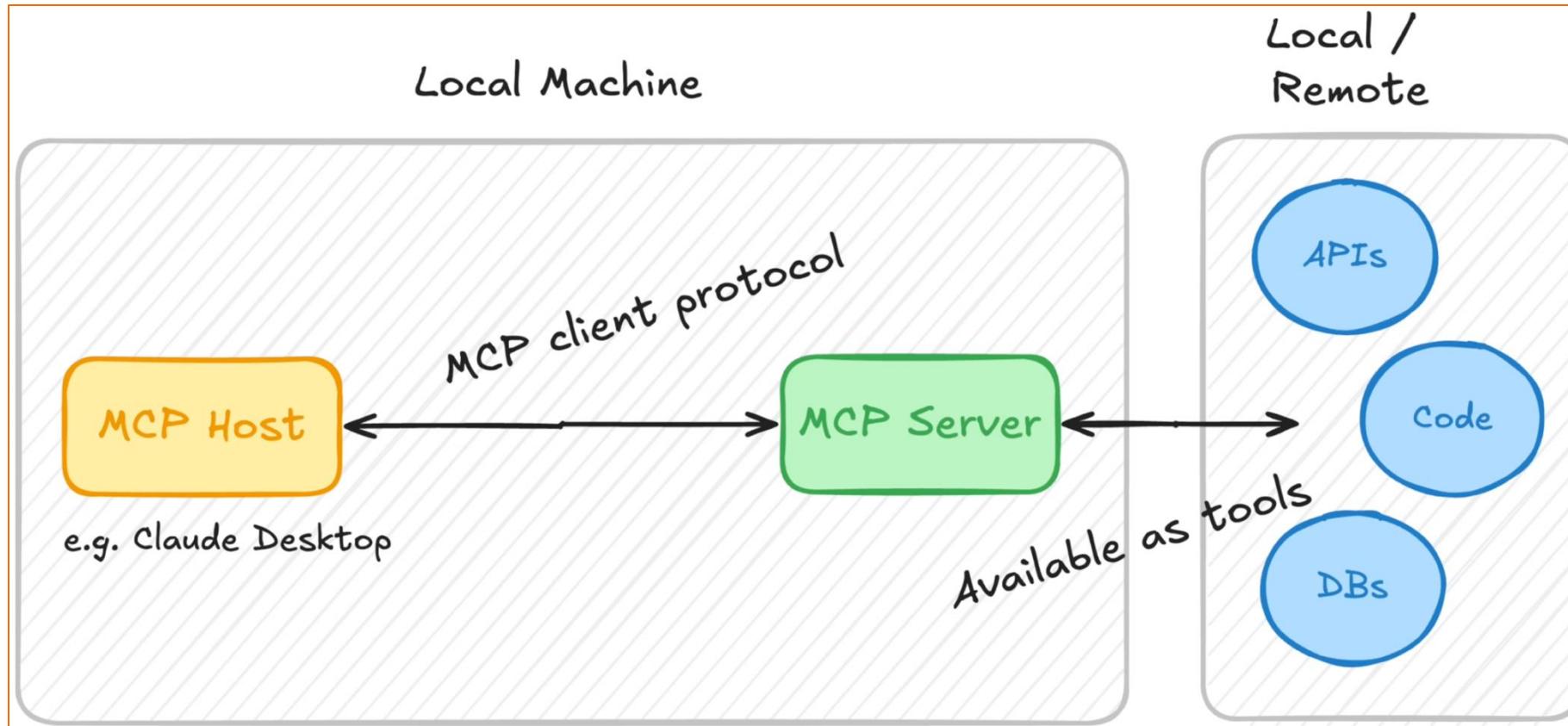
| SOURCE | TOKENS | SNIPPETS | UPDATE |
|---------------|--------|----------|---------|
| Next.js | 678K | 3K | 2 days |
| MongoDB | 15.8M | 131K | 3 weeks |
| Shadcn UI | 215K | 1.3K | 2 weeks |
| Vercel AI SDK | 559K | 2.4K | 1 week |
| React | 834K | 2.8K | 1 week |
| Better Auth | 266K | 1.3K | 2 days |
| Tailwind CSS | 261K | 1.7K | 1 week |
| Supabase | 1.4M | 4.6K | 2 days |

<https://context7.com/>

Claude Code



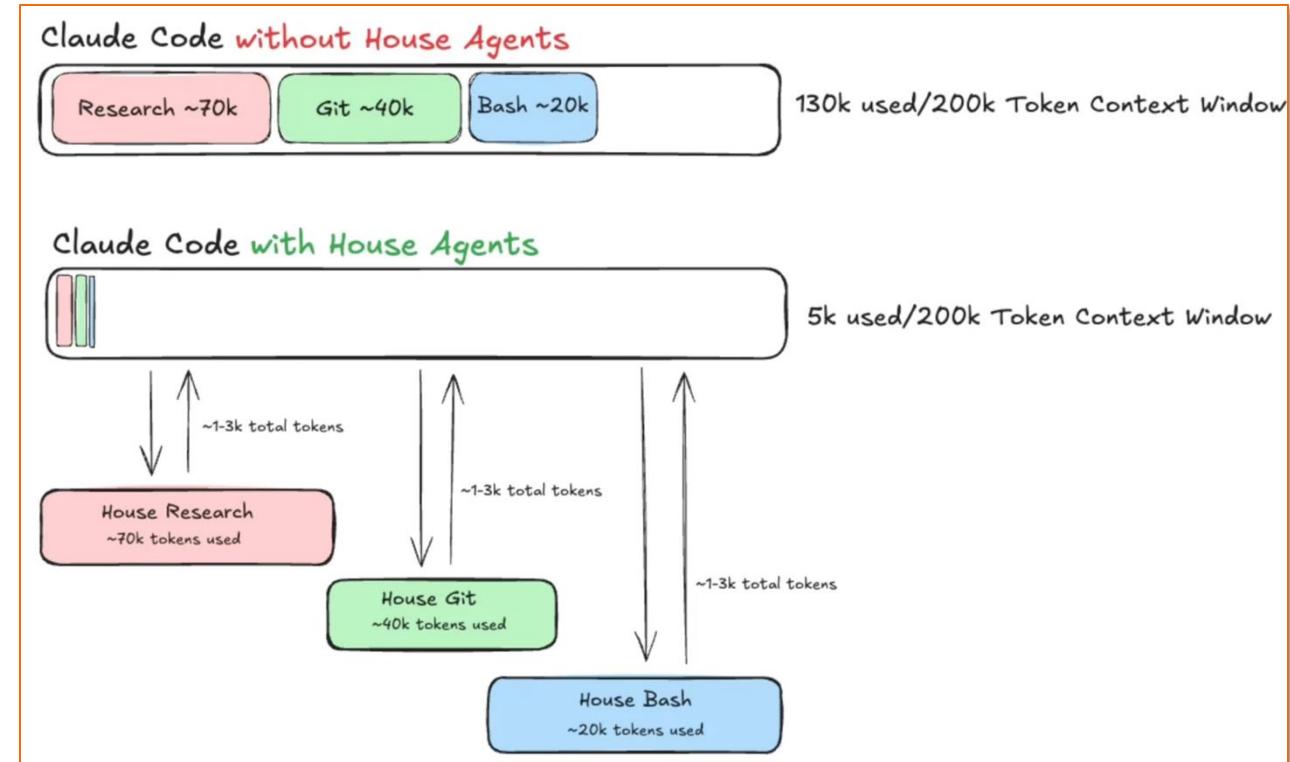
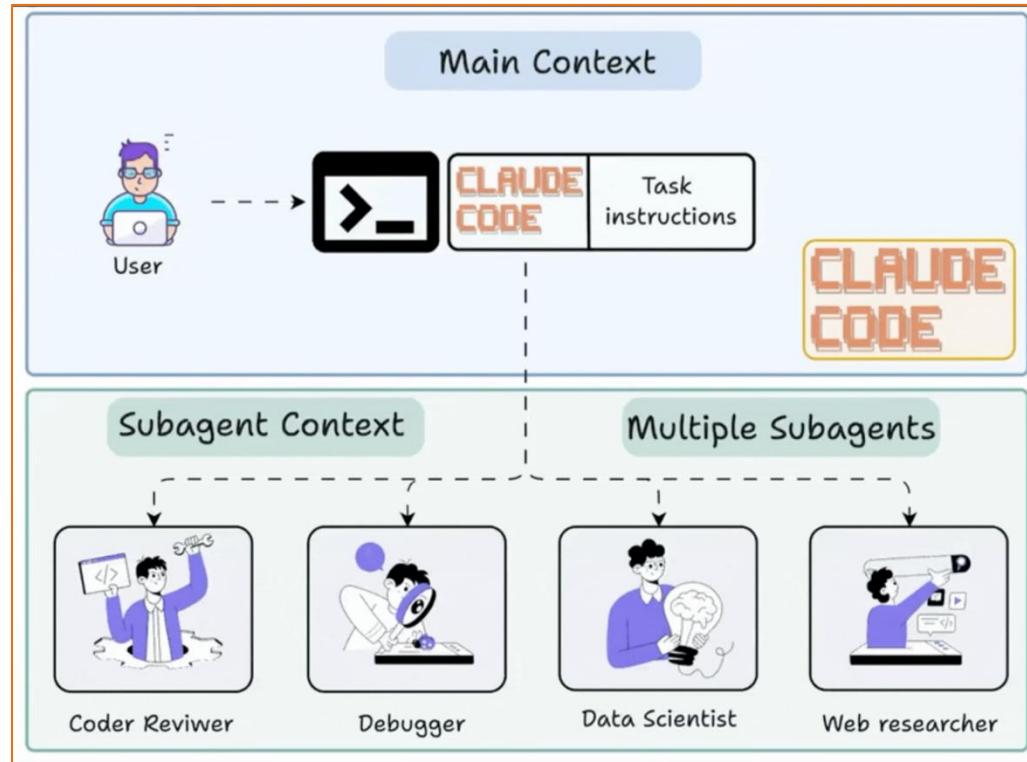
MCP: Externe Daten/ Services über „APIs“ verbinden



Claude Code



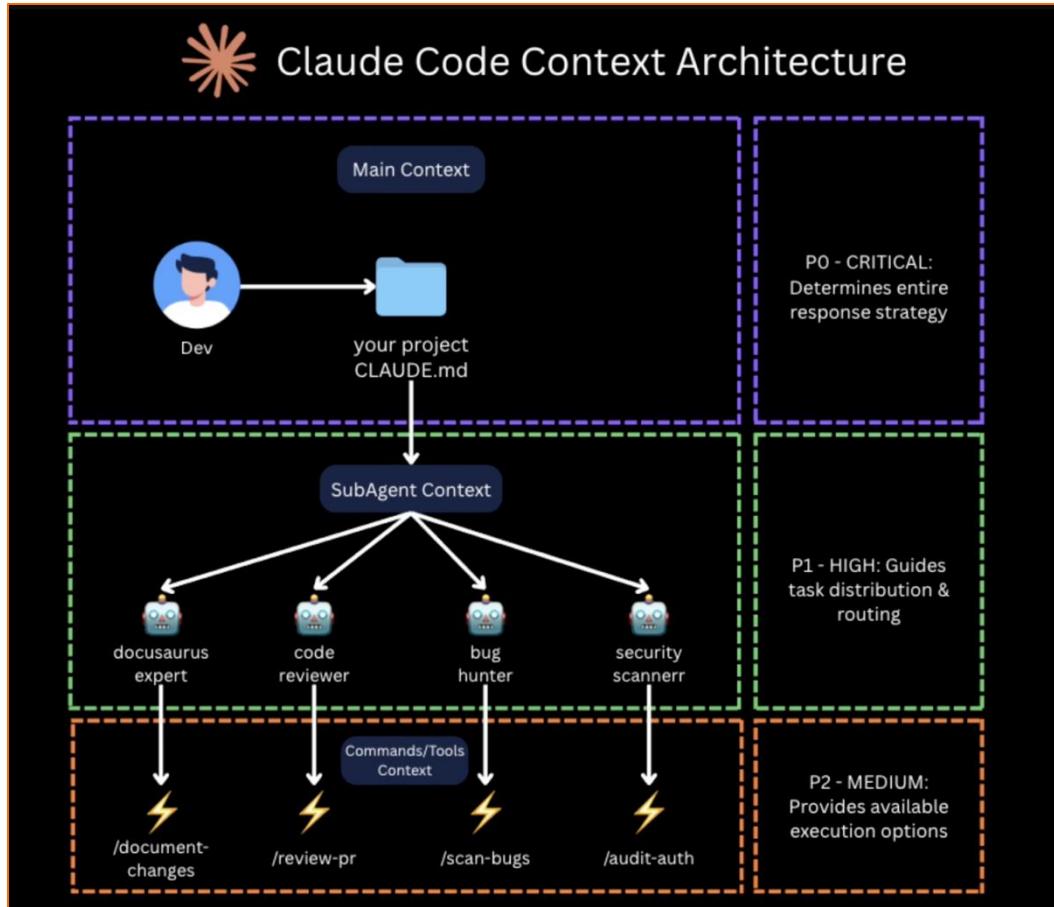
Agenten – Claude Code Subagents



<https://github.com/VoltAgent/awesome-claude-code-subagents>

Claude Code

Setup eines KI-Entwicklers



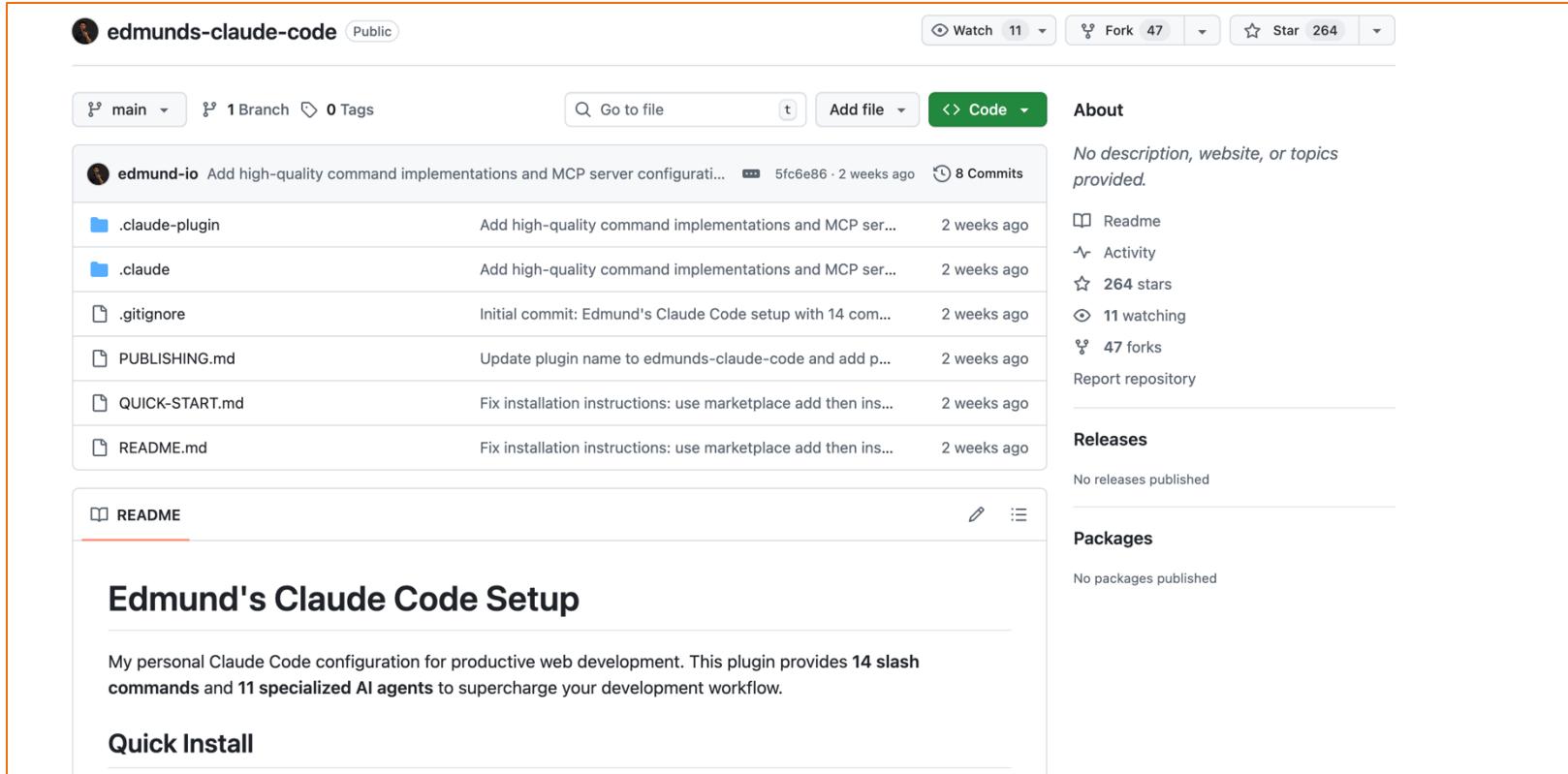
Screenshot of the Claude Code application interface:

- Header: `▶ + transcribr git:(main) x claude`
- Icon: Orange alien icon.
- Version: Claude Code v2.0.26
- Project: Sonnet 4.5 · Claude Max
- Path: /Users/edmund/Documents/GitHub/transcribr
- Section: `> /agents`
- Section: Agents
- Text: 14 agents
- Section: Create new agent
- Section: User agents (/Users/edmund/.claude/agents)
- List of agents:
 - system-architect · sonnet △ overridden by projectSettings
 - performance-engineer · sonnet △ overridden by projectSettings
 - refactoring-expert · sonnet △ overridden by projectSettings
 - requirements-analyst · sonnet △ overridden by projectSettings
 - learning-guide · sonnet △ overridden by projectSettings
 - backend-architect · sonnet △ overridden by projectSettings
 - security-engineer · sonnet △ overridden by projectSettings
 - deep-research-agent · sonnet △ overridden by projectSettings
 - frontend-architect · sonnet △ overridden by projectSettings
 - technical-writer · sonnet △ overridden by projectSettings

<https://github.com/mylee04/clause-code-subagents/blob/main/agents/quality/code-reviewer.md>

Claude Code

 Plugins = Marktplatz für Agenten, Templates, ...



The screenshot shows a GitHub repository page for 'edmunds-claude-code'. The repository is public and has 11 watchers, 47 forks, and 264 stars. It contains 1 branch and 0 tags. The main branch has 8 commits. The repository includes files like .claude-plugin, .claude, .gitignore, PUBLISHING.md, QUICK-START.md, and README.md. The README file is currently selected. The content of the README describes 'Edmund's Claude Code Setup' as a personal configuration for productive web development, featuring 14 slash commands and 11 specialized AI agents. A 'Quick Install' section provides instructions. The repository has no releases or packages published.

<https://github.com/edmund-io/edmunds-claude-code>

Claude Code + Senior Entwickler Know-How

🟡 BMAD Github Repo

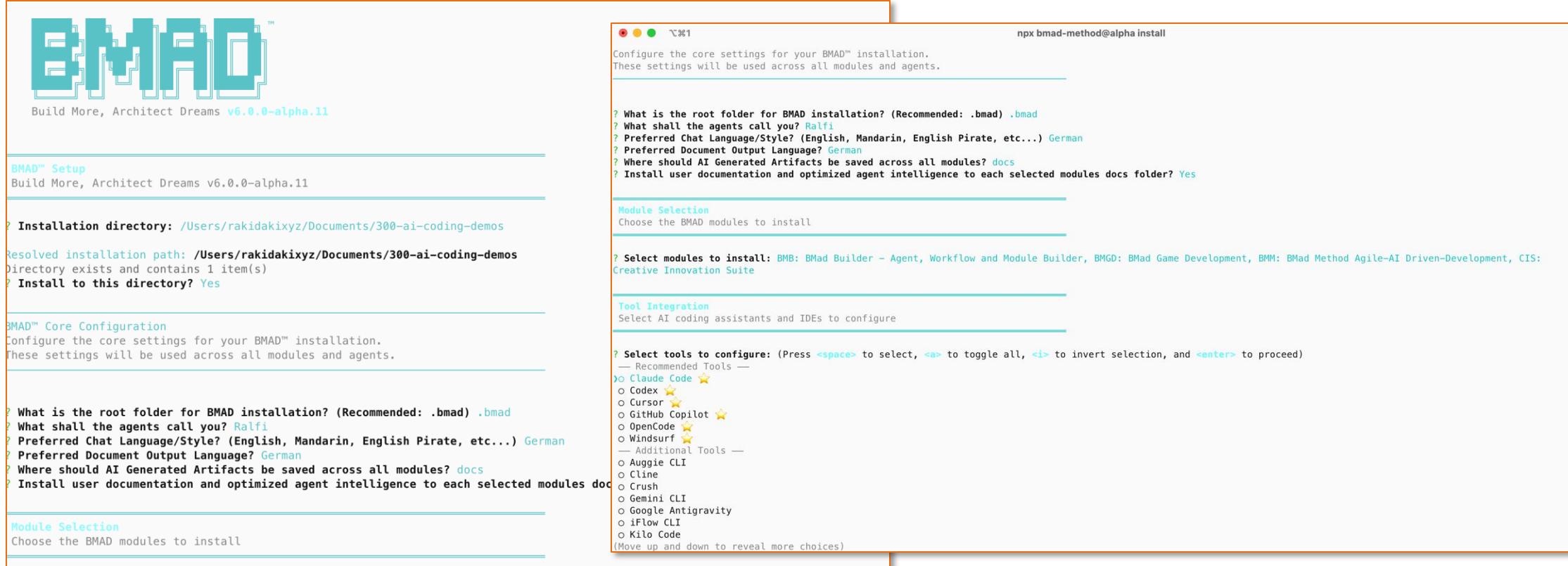
The screenshot shows the GitHub repository page for `BMAD-METHOD`. The repository is public and has 31 issues, 10 pull requests, and 22 branches. The main branch is `main`, which was last updated yesterday. There are 839 commits in total. The repository is described as "Breakthrough Method for Agile Ai Driven Development". It includes links to Readme, View license, Contributing, Activity, Custom properties, 21.5k stars, 308 watching, 3.1k forks, and a Report repository. There are 12 releases, with the latest being `BMad Method v4.44.1` from Sep 30. A "Sponsor this project" button links to `buyamecoffee.com/bmad`. No packages have been published. Contributors include Brian Madison and Brian Madison, with 75 total contributors.

| File/Folder | Description | Last Commit |
|------------------------------------|---|--------------|
| <code>.bmad</code> | Major Enhancements: | yesterday |
| <code>.claude/commands/bmad</code> | Major Enhancements: | yesterday |
| <code>.github</code> | chore: CC PR review with GH token (#874) | 5 days ago |
| <code>.husky</code> | chore: added CC PR review (#871) | last week |
| <code>.vscode</code> | Major Enhancements: | yesterday |
| <code>docs</code> | feat: Extract BMGD module and implement workflow vend... | 5 days ago |
| <code>src</code> | more updates to use ephemeral file location for phase 4 it... | yesterday |
| <code>test</code> | Major Enhancements: | yesterday |
| <code>tools</code> | agents now are not duplicated and instead cli command... | yesterday |
| <code>web-bundles</code> | Major Enhancements: | yesterday |
| <code>.gitignore</code> | chore: added CC PR review (#871) | last week |
| <code>.npmrc</code> | feat: v6.0.0-alpha.0 - the future is now | 2 months ago |
| <code>.nvmrc</code> | feat: v6.0.0-alpha.0 - the future is now | 2 months ago |
| <code>.prettierignore</code> | feat: add agent schema validation with comprehensive te... | 3 weeks ago |
| <code>CHANGELOG.md</code> | release: bump to v6.0.0-alpha.5 | last week |
| <code>CONTRIBUTING.md</code> | release: bump to v6.0.0-alpha.6 | last week |

<https://github.com/bmad-code-org/BMAD-METHOD>

Claude Code + Senior Entwickler Know-How

🟡 BMAD Github Repo



The screenshot shows the GitHub repository interface for the BMAD project. The repository name is 'BMAD' and it has a tag 'Build More, Architect Dreams v6.0.0-alpha.11'. The repository has 10 stars and 14 forks. A yellow emoji with a sad face is displayed next to the repository name.

BMAD™ Setup
Build More, Architect Dreams v6.0.0-alpha.11

? Installation directory: /Users/rakidakixyz/Documents/300-ai-coding-demos
Resolved installation path: /Users/rakidakixyz/Documents/300-ai-coding-demos
Directory exists and contains 1 item(s)
? Install to this directory? Yes

BMAD™ Core Configuration
Configure the core settings for your BMAD™ installation.
These settings will be used across all modules and agents.

? What is the root folder for BMAD installation? (Recommended: .bmad) .bmad
? What shall the agents call you? Ralfi
? Preferred Chat Language/Style? (English, Mandarin, English Pirate, etc...) German
? Preferred Document Output Language? German
? Where should AI Generated Artifacts be saved across all modules? docs
? Install user documentation and optimized agent intelligence to each selected modules docs folder? Yes

Module Selection
Choose the BMAD modules to install

? Select modules to install: BMB: BMad Builder – Agent, Workflow and Module Builder, BMGD: BMad Game Development, BMM: BMad Method Agile-AI Driven-Development, CIS: Creative Innovation Suite

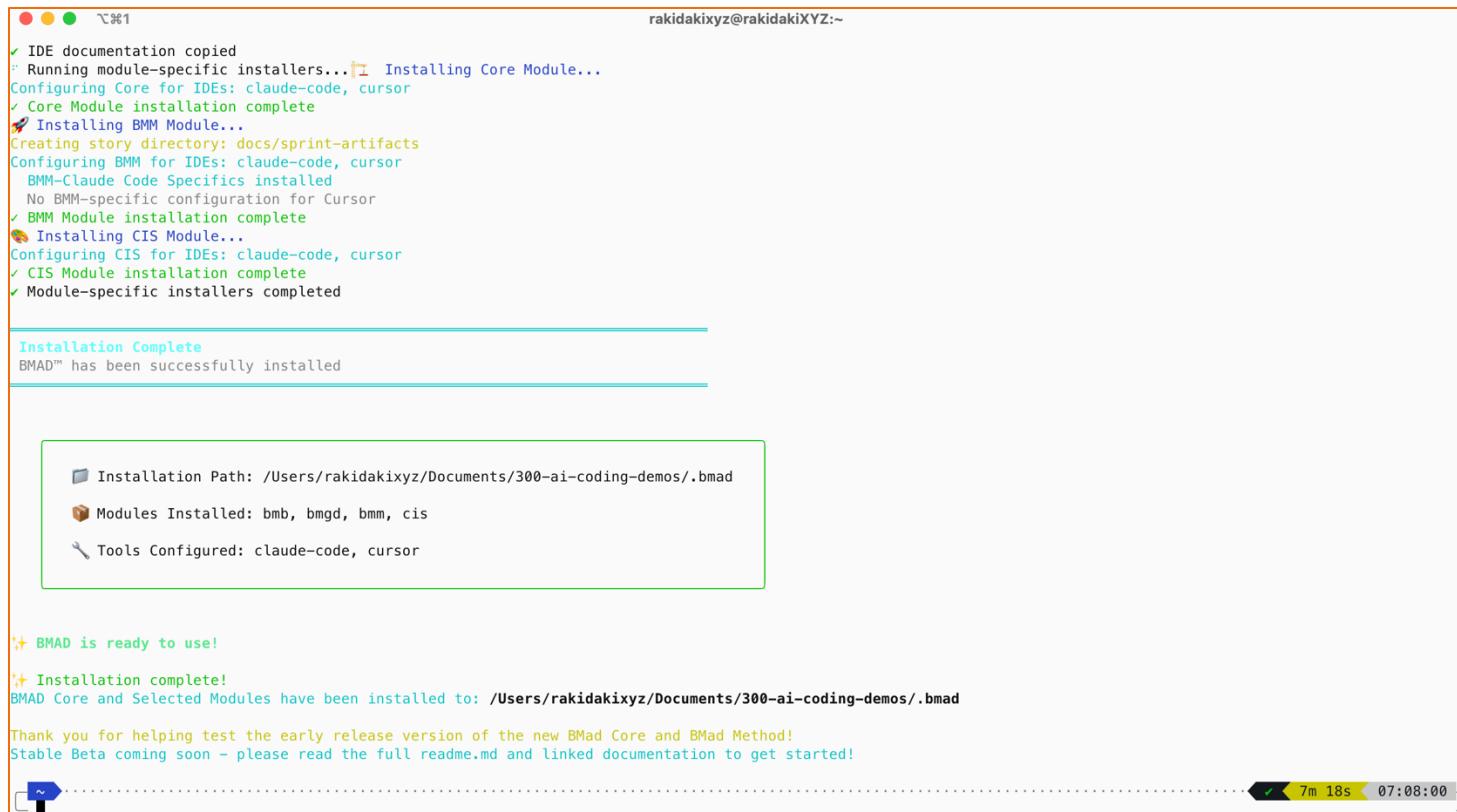
Tool Integration
Select AI coding assistants and IDEs to configure

? Select tools to configure: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
— Recommended Tools —
o Claude Code ★★
o Codex ★★
o Cursor ★★
o GitHub Copilot ★★
o OpenCode ★★
o Windsurf ★★
— Additional Tools —
o Augie CLI
o Cline
o Crush
o Gemini CLI
o Google Antigravity
o iFlow CLI
o Kilo Code
(Move up and down to reveal more choices)

Terminal: npx bmad-method@alpha install

Claude Code + Senior Entwickler Know-How

🟡 BMAD Github Repo



```
rakidakixyz@rakidakiXYZ:~
✓ IDE documentation copied
✖ Running module-specific installers... [+] Installing Core Module...
Configuring Core for IDEs: claude-code, cursor
✓ Core Module installation complete
✖ Installing BMM Module...
Creating story directory: docs/sprint-artifacts
Configuring BMM for IDEs: claude-code, cursor
  BMM-Claude Code Specifics installed
  No BMM-specific configuration for Cursor
✓ BMM Module installation complete
✖ Installing CIS Module...
Configuring CIS for IDEs: claude-code, cursor
✓ CIS Module installation complete
✓ Module-specific installers completed

Installation Complete
BMAD™ has been successfully installed

  Installation Path: /Users/rakidakixyz/Documents/300-ai-coding-demos/.bmad
  Modules Installed: bmb, bmfd, bmm, cis
  Tools Configured: claude-code, cursor

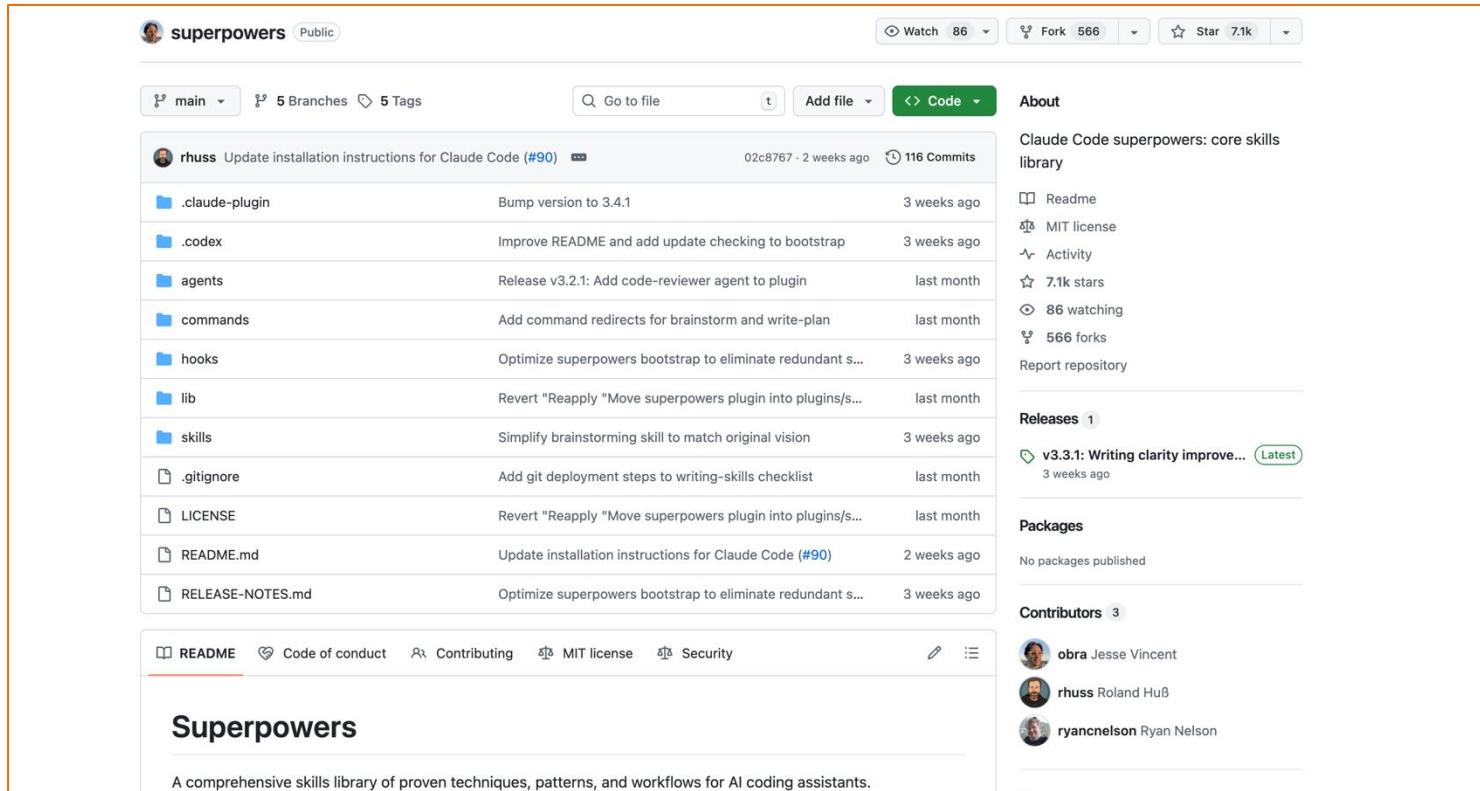
✖ BMAD is ready to use!
✖ Installation complete!
BMAD Core and Selected Modules have been installed to: /Users/rakidakixyz/Documents/300-ai-coding-demos/.bmad

Thank you for helping test the early release version of the new BMAD Core and BMAD Method!
Stable Beta coming soon - please read the full readme.md and linked documentation to get started!
```

Installation fertig, jetzt könnte man mit claude oder cursor starten

Claude Code

Claude Superpowers



About

Claude Code superpowers: core skills library

- Readme
- MIT license
- Activity
- 7.1k stars
- 86 watching
- 566 forks

Report repository

Releases 1

v3.3.1: Writing clarity improve... Latest
3 weeks ago

Packages

No packages published

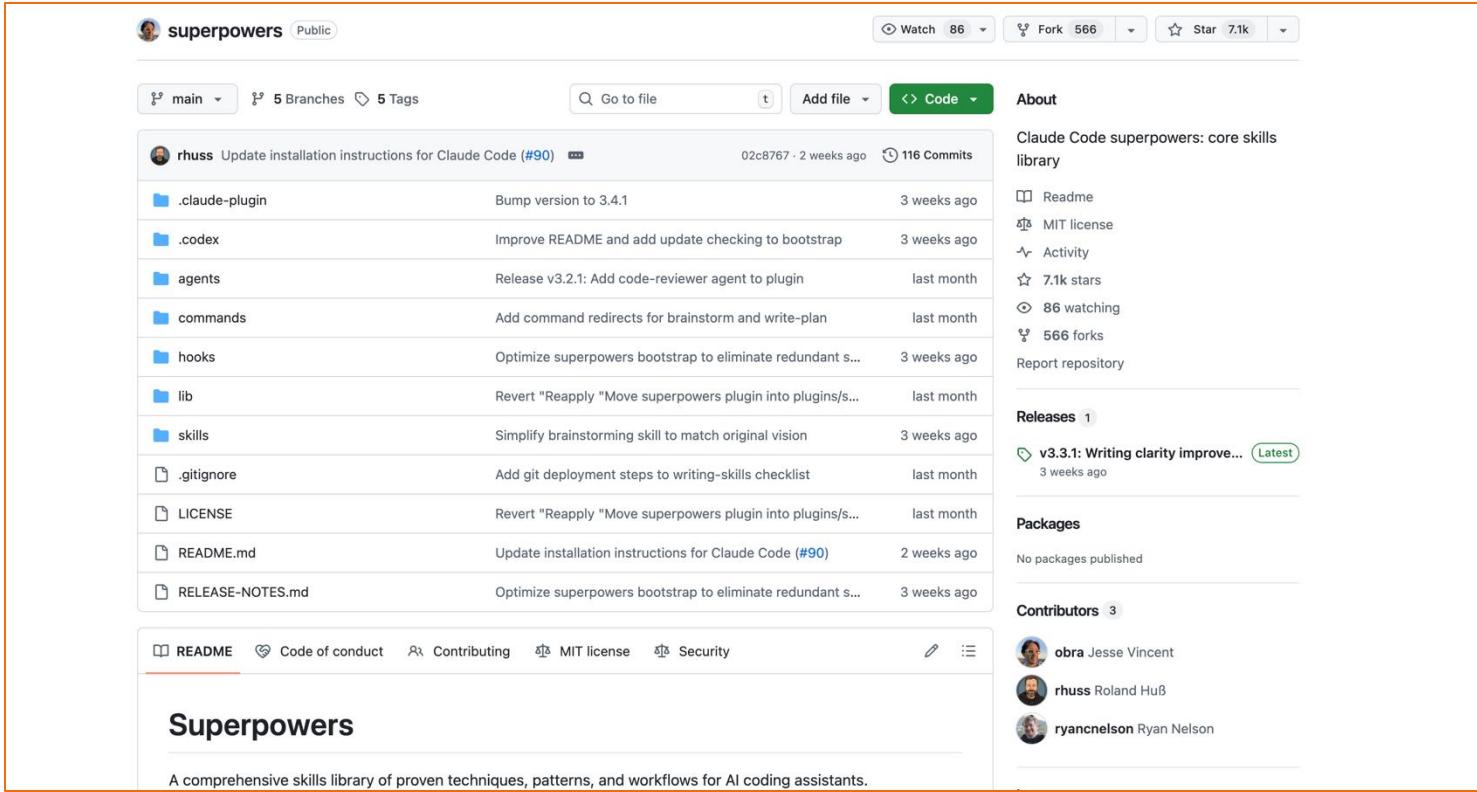
Contributors 3

- obra Jesse Vincent
- rhuss Roland Huß
- ryancnelson Ryan Nelson

<https://github.com/obra/superpowers>

Claude Code

Claude Superpowers



The screenshot shows the GitHub repository page for 'superpowers' (Public). The repository has 86 watchers, 566 forks, and 7.1k stars. It contains 5 branches and 5 tags. The main branch has 116 commits from user 'rhuss'. Recent commits include bumping the version to 3.4.1 and releasing v3.2.1. The repository is described as a 'core skills library' for Claude Code. It includes a README, MIT license, and security information. There is one release, v3.3.1, which improves writing clarity. No packages are published, and three contributors are listed: obra, rhuss, and ryancnelson.

superpowers Public

Watch 86 Fork 566 Star 7.1k

main 5 Branches 5 Tags Go to file Add file Code

rhuss Update installation instructions for Claude Code (#90) 02c8767 - 2 weeks ago 116 Commits

.claude-plugin Bump version to 3.4.1 3 weeks ago

.codex Improve README and add update checking to bootstrap 3 weeks ago

agents Release v3.2.1: Add code-reviewer agent to plugin last month

commands Add command redirects for brainstorm and write-plan last month

hooks Optimize superpowers bootstrap to eliminate redundant s... 3 weeks ago

lib Revert "Reapply "Move superpowers plugin into plugins/s..." last month

skills Simplify brainstorming skill to match original vision 3 weeks ago

.gitignore Add git deployment steps to writing-skills checklist last month

LICENSE Revert "Reapply "Move superpowers plugin into plugins/s..." last month

README.md Update installation instructions for Claude Code (#90) 2 weeks ago

RELEASE-NOTES.md Optimize superpowers bootstrap to eliminate redundant s... 3 weeks ago

README Code of conduct Contributing MIT license Security

Superpowers

A comprehensive skills library of proven techniques, patterns, and workflows for AI coding assistants.

About

Claude Code superpowers: core skills library

Readme MIT license Activity 7.1k stars 86 watching 566 forks Report repository

Releases 1

v3.3.1: Writing clarity improve... Latest 3 weeks ago

Packages

No packages published

Contributors 3

obra Jesse Vincent

rhuss Roland Huß

ryancnelson Ryan Nelson

<https://github.com/obra/superpowers>

Claude Code



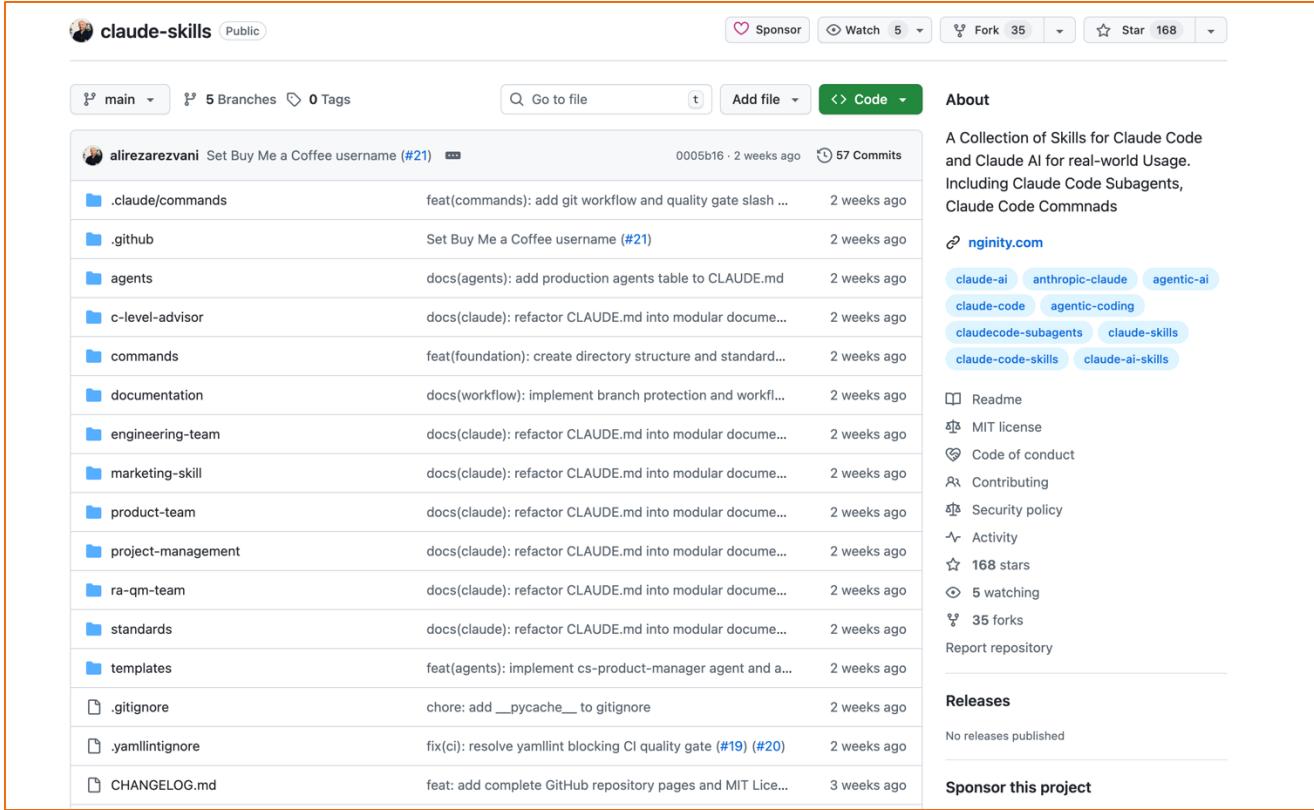
Claude Code Factory Architecture

The screenshot shows the GitHub repository page for `claude-code-tresor`. The repository is public and has 149 stars, 5 forks, and 26 commits. The repository structure includes branches like `main`, `4 Branches`, and `2 Tags`. Key commits include adding Meta prompting Slash commands and improvements to Subagents. The repository is associated with `alirezarezvani.com` and uses labels such as `agents`, `anthropic-claude`, `agentic-workflow`, `agentic-ai`, `claude-code`, `agent-development-kit`, and `agentic-coding`. It also includes a `Readme`, `MIT license`, and `Contributing` files. The repository has 149 stars, 5 watchers, and 56 forks.

<https://github.com/alirezarezvani/claude-code-tresor>

Claude Code

Claude Code Factory Architecture



The screenshot shows a GitHub repository page for 'claude-skills' (Public). The repository has 5 branches and 0 tags. The main branch is selected. The repository was created by alirezarezvani on 0005b16 · 2 weeks ago with 57 commits. The commit history includes various contributions from different teams like engineering-team, product-team, and marketing-skill. The repository includes sub-agents for Claude Code and Claude AI, and features like git workflow and quality gate. It also includes documentation for agents and commands. The repository is associated with 'nginity.com' and has labels for 'claude-ai', 'anthropic-claude', 'agentic-ai', 'claude-code', 'agentic-coding', 'claudicode-subagents', 'claude-skills', 'claude-code-skills', and 'claude-ai-skills'. The repository has 168 stars, 5 watchers, 35 forks, and a report repository. There are no releases published. A 'Sponsor this project' button is at the bottom.

A Collection of Skills for Claude Code and Claude AI for real-world Usage.
Including Claude Code Subagents,
Claude Code Commands

claude-ai anthropic-claude agentic-ai
claude-code agentic-coding
claudicode-subagents claude-skills
claude-code-skills claude-ai-skills

Readme
MIT license
Code of conduct
Contributing
Security policy
Activity
168 stars
5 watching
35 forks
Report repository

No releases published

Sponsor this project

<https://github.com/alirezarezvani/claude-skills>

Claude Code



Claude Code Factory Architecture

The screenshot shows the GitHub repository page for `claude-code-skill-factory`. The repository is a public template with 95 commits, 2 branches, and 0 tags. The repository has 117 stars, 4 forks, and 4 watchers. The repository URL is <https://github.com/alirezarezvani/claude-code-skill-factory>.

About

Claude Code Skill Factory — A powerful open-source toolkit for building and deploying production-ready Claude Skills, Code Agents, custom Slash Commands, and LLM Prompts at scale. Easily generate structured skill templates, automate workflow integration, and accelerate AI agent development with a clean, developer-friendly setup.

medium.com/@alirezarezvani/the-clau...

Tags

- ai-agents
- ai-tools
- claude-ai
- claude-code
- claude-skills
- claude-skills-creator

Files

- `archive`: security: Add comprehensive security audit and policy (3 weeks ago)
- `claudie`: feat: Add App Store Optimization skill + Hook Factory v2.... (2 weeks ago)
- `github`: feat(workflows): add auto-branch creation from labeled is... (last week)
- `vscode`: chore: Update configuration and reorganize README (3 weeks ago)
- `claude-skills-examples`: feat: Implement modular CLAUDE.md architecture (Anthr... (3 weeks ago)
- `documentation`: Feat add git governance workflows (#1014) (2 weeks ago)
- `generated-agents`: Fix/psychology advisor link (#1023) (last week)
- `generated-commands`: Fix/psychology advisor link (#1023) (last week)
- `generated-hooks`: feat: Add App Store Optimization skill + Hook Factory v2.... (2 weeks ago)
- `generated-prompts`: fix: disable yamllint check (false positives) (#1018) (2 weeks ago)
- `generated-skills`: Fix/psychology advisor link (#1023) (last week)
- `.commitlintrc.cjs`: feat(git): add comprehensive Git governance and CI workf... (2 weeks ago)
- `.editorconfig`: chore: Professional repository cleanup and standards (v2.... (3 weeks ago)
- `.gitignore`: feat: Add App Store Optimization skill + Hook Factory v2.... (2 weeks ago)
- `CHANGELOG.md`: docs: add v1.5.0 changelog entry (#1019) (2 weeks ago)
- `CLAUDE.md`: Fix/psychology advisor link (#1023) (last week)

Repository Details

- Readme
- MIT license
- Contributing
- Security policy
- Activity
- 117 stars
- 4 watching
- 40 forks

[Report repository](#)

Releases

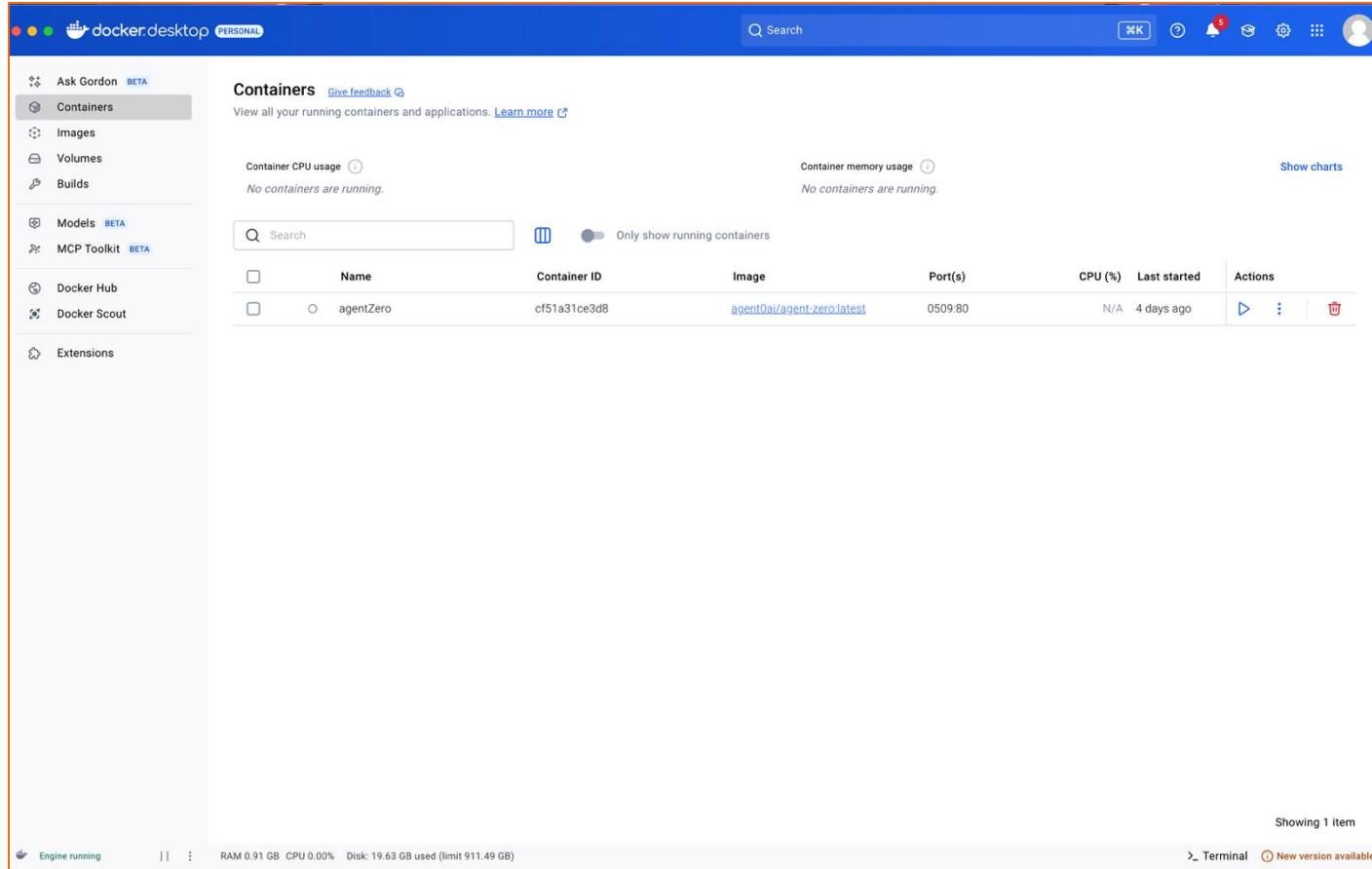
<https://github.com/alirezarezvani/claude-code-skill-factory>

Lokale Ollama + OpenWebUI KI Umgebung

Docker, Ollama, OpenWebUI, Cloudflare

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

 Lokale Umgebung + online erreichbar?



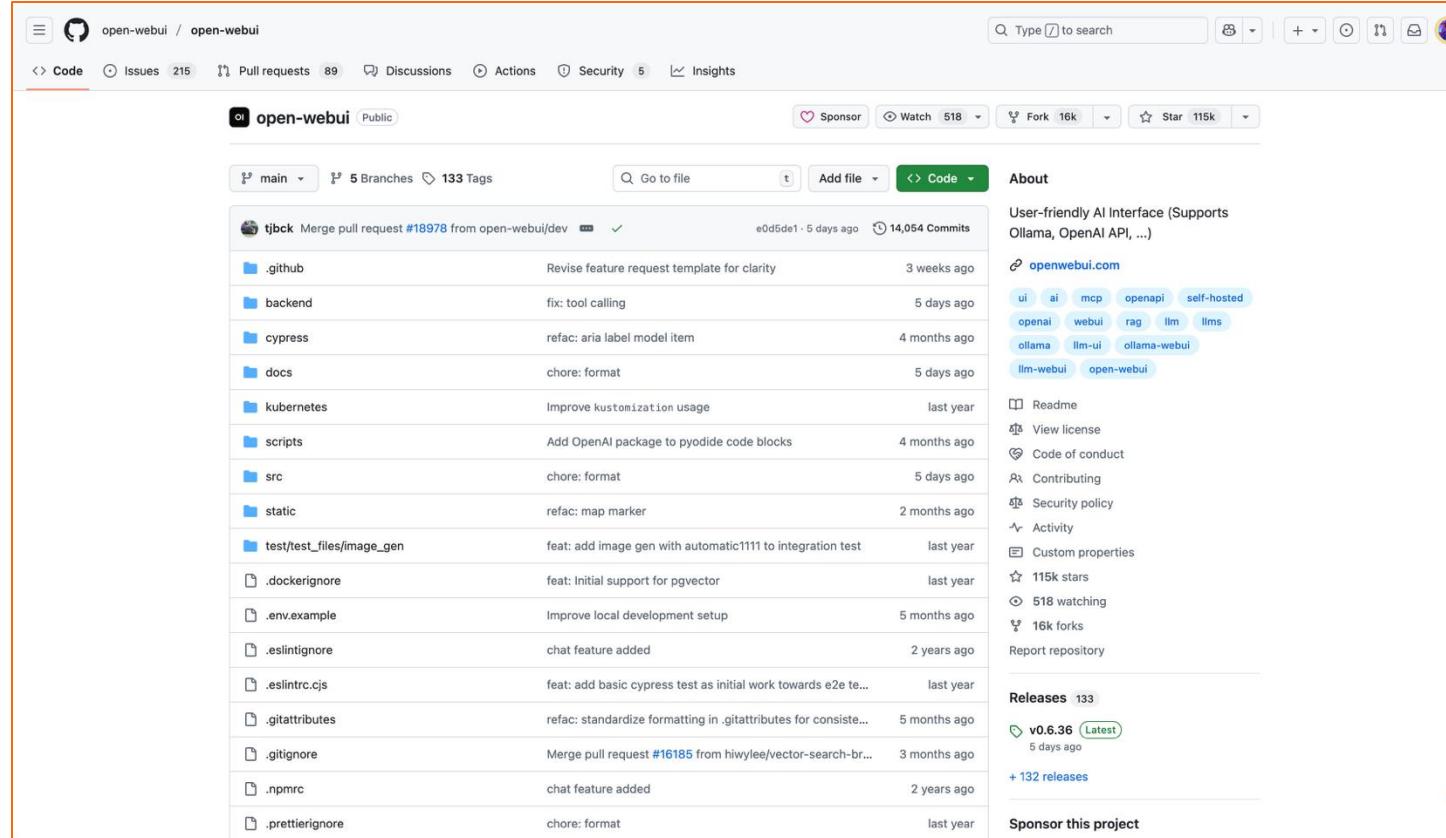
<https://www.docker.com/>

Docker – Docker Desktop (Mac, Windows):

Container Lösung, um „innerhalb“ des Betriebssystems eines PC's oder Notebooks eine sichere virtuelle Umgebung einzurichten für eine komplette KI-Lösung

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

 Lokale Umgebung + online erreichbar?



The screenshot shows the GitHub repository page for 'open-webui / open-webui'. The repository has 215 issues, 89 pull requests, and 5 discussions. It has 5 branches and 133 tags. The main branch has 14,054 commits. The repository description is 'User-friendly AI Interface (Supports Ollama, OpenAI API, ...)'. It features a 'About' section with links to 'openwebui.com' and various tags like 'ui', 'ai', 'mcp', 'openapi', 'self-hosted', 'openai', 'webui', 'rag', 'ilm', 'llms', 'ollama', 'ilm-ui', 'ollama-webui', 'ilm-webui', and 'open-webui'. The repository has 115k stars, 518 watching, and 16k forks. It includes sections for 'Readme', 'View license', 'Code of conduct', 'Contributing', 'Security policy', 'Activity', 'Custom properties', 'Report repository', and 'Releases' (133). The latest release is v0.6.36, which was released 5 days ago. There is also a 'Sponsor this project' button.

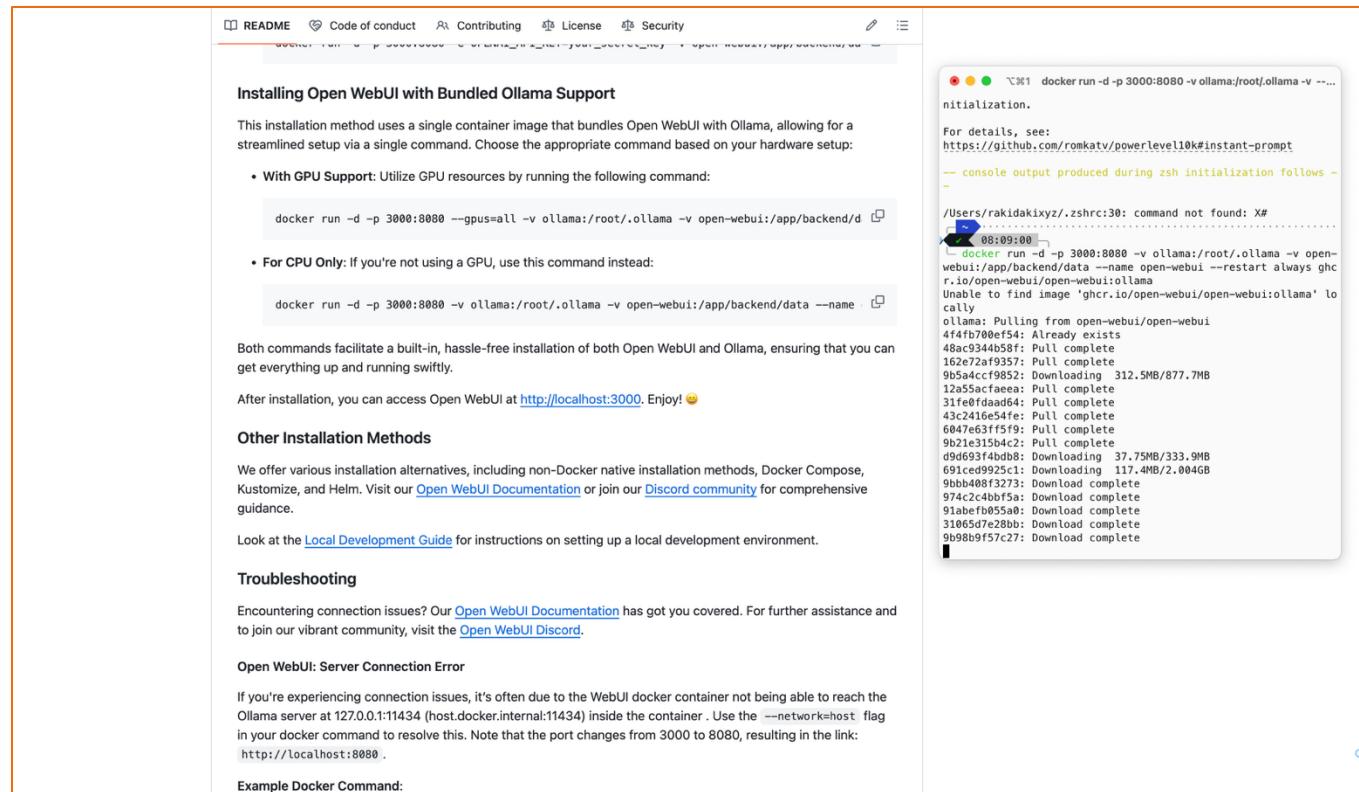
<https://github.com/open-webui/open-webui>

OpenWebUI + Ollama

Chatbot Oberfläche wie bei OpenAI ChatGPT und lokale Sprachmodelle und RAG Lösung.

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

 Lokale Umgebung + online erreichbar?



The screenshot shows the GitHub repository page for "open-webui/open-webui". It features a sidebar with links like README, Code of conduct, Contributing, License, and Security. The main content area is titled "Installing Open WebUI with Bundled Ollama Support". It provides two command examples: one for GPU support (using --gpus=all) and one for CPU only. Both commands involve running a Docker container with Ollama and Open WebUI. Below the commands, it says both facilitate a built-in, hassle-free installation. After installation, you can access it at <http://localhost:3000>. The "Other Installation Methods" section lists Docker Compose, Kustomize, and Helm as alternatives. The "Troubleshooting" section links to the documentation and the Discord community. The "Open WebUI: Server Connection Error" section explains how to resolve connection issues. At the bottom, there's an "Example Docker Command:" section.

<https://github.com/open-webui/open-webui>

Terminal für die Installation des „Docker-Pakets mit Ollama und Open WebUI als Bundle“:

Windows Rechner mit Grafikkarte - also GPU's dann die Zeile für das Terminal mit GPU Support verwenden

Apple Rechner (MacBooks) die Zeile mit CPU Only Support verwenden (klappt trotzdem sehr gut ...)

```
docker run -d -p 3000:8080 -v ollama:/root/.ollama -v open-webui:/app/backend/d...  
initialization.  
For details, see:  
https://github.com/romkatv/powerlevel10k#instant-prompt  
-- console output produced during zsh initialization follows --  
  
/Users/rakdakinyz/.zshrc:30: command not found: X#  
08:09:08 docker run -d -p 3000:8080 -v ollama:/root/.ollama -v open-webui:/app/backend/data --name open-webui/open-webui:ollama  
Unable to find image 'ghcr.io/open-webui/open-webui:ollama' locally  
ollama: Pulling from open-webui/open-webui  
4f4fb700ef54: Already exists  
48ac934ab58f: Pull complete  
162e72af9357: Pull complete  
9b5a4ccf9852: Downloading 312.5MB/877.7MB  
12a55acf9eaa: Pull complete  
31fe0fdaad64: Pull complete  
43c2416e54fe: Pull complete  
6047e6354b9: Pull complete  
9b21e315b4c2: Pull complete  
d946934b4db8: Pull complete  
691ced9925c1: Downloading 37.75MB/333.9MB  
0bbba408f3273: Download complete  
974c24bbf5fa: Download complete  
91abefb055a0: Download complete  
31065d7e28bb: Download complete  
9b9889f57c27: Download complete  
  
s ghcr.io/open-webui/open-webui:ollama  
Unable to find image 'ghcr.io/open-webui/open-webui:ollama' locally  
ollama: Pulling from open-webui/open-webui  
4f4fb700ef54: Pull complete  
48ac934ab58f: Pull complete  
162e72af9357: Pull complete  
9b5a4ccf9852: Pull complete  
12a55acf9eaa: Pull complete  
31fe0fdaad64: Pull complete  
43c2416e54fe: Pull complete  
6047e63ff5f9: Pull complete  
9b21e315b4c2: Pull complete  
d946934b4db8: Pull complete  
691ced9925c1: Pull complete  
9bbba408f3273: Pull complete  
974c24bbf5fa: Pull complete  
91abefb055a0: Pull complete  
31065d7e28bb: Pull complete  
9b9889f57c27: Pull complete  
Digest: sha256:d6e7e2dd06800229defdfde55021d157a1689a6ee2ef16c42b083936ea388a  
Status: Downloaded newer image for ghcr.io/open-webui/open-webui:ollama  
a5b9beb417d885403281alf867292de92cf5050293a138de720703aca3f61880
```

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?

| Name | Created | Size | Actions |
|------------|----------------|---------|---------|
| ollama | 56 seconds ago | 0 Bytes | |
| open-webui | 56 seconds ago | 230 MB | |

Showing 2 items

Walkthroughs

- Multi-container applications 8 mins
- Persist your data between containers 3 mins

[View more in the Learning center](#)

Engine running | RAM 7.29 GB CPU 1.86% Disk: 30.72 GB used (limit 911.49 GB) Terminal New version available

Installation im Terminal erfolgreich:

Sobald im Terminal ohne Fehlermeldung die Installation erfolgreich abgeschlossen ist (Sie sehen keine Erfolgsmeldung im Terminal nur einen blinkenden Cursor für den nächsten Befehl), können Sie in Docker Desktop unter den Volumes die beiden installierten Pakete ollama und open-webui aufgelistet sehen.

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?

The screenshot shows the Docker Desktop interface with the 'Containers' tab selected. There are two containers listed:

| Name | Container ID | Image | Port(s) | CPU (%) | Last started | Actions |
|------------|--------------|------------------------------|-----------|---------|---------------|---------|
| agentZero | cf51a31ce3d8 | agent0ai/agent-zero:latest | 0509:80 | 0% | 4 days ago | |
| open-webui | a5b9beb417d8 | open-webui/open-webui:ollama | 3000:8080 | 6.18% | 4 minutes ago | |

At the bottom of the interface, there is a status bar showing system resources: RAM 7.16 GB, CPU 0.36%, Disk: 30.72 GB used (limit 911.49 GB). A 'Terminal' link and a 'New version available' notification are also present.

Docker Paket starten:

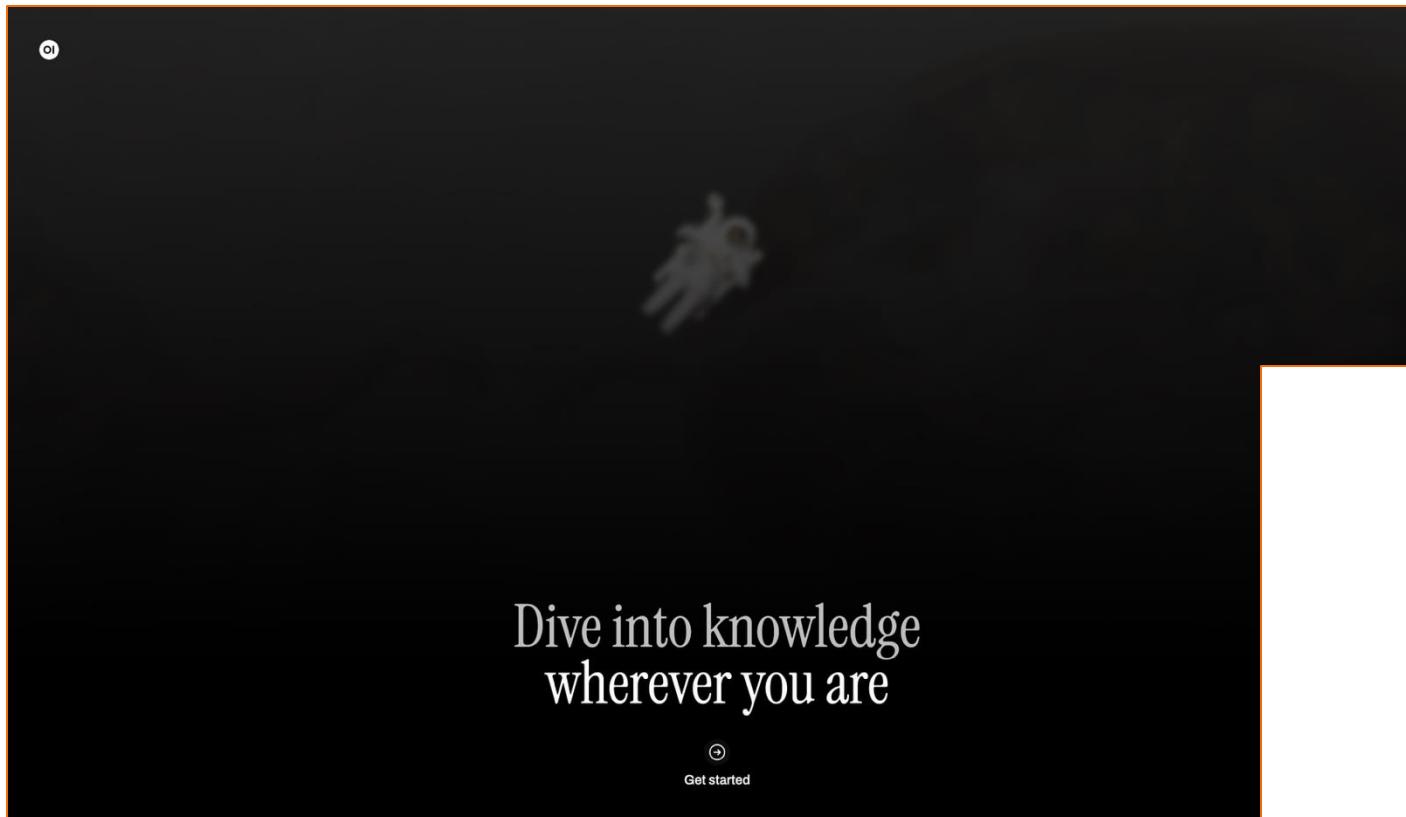
Wenn man in Docker Desktop dann auf „Containers“ geht, findet man alle virtuellen Container mit den jeweiligen Anwendungen, die Sie installiert haben.

Hier sieht man das Paket open-webui mit dem integrierten Ollama, welches schon gestartet ist und über einen Link in der Spalte Ports aufgerufen werden kann.

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?



Lokale Web-Anwendung Ollama + OpenWebUI

Sobald man den Link angeklickt hat, startet der lokale Webserver localhost mit der Anwendung.

Man muss jetzt ein Account anlegen

Get started with Open WebUI

ⓘ Open WebUI does not make any external connections, and your data stays securely on your locally hosted server.

Name
rakidakiXYZ

Email
ralf@from-scratch.ai

Password
.....

Create Admin Account

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?

The screenshot shows the Open WebUI application interface. On the left is a sidebar with icons for 'Open WebUI', 'Select a model', 'New Chat', 'Search', 'Notes', 'Workspace', 'Folders', and 'Chats'. The main area displays a modal titled 'What's New in Open WebUI' for version v0.6.36 (2025-11-07). The modal lists changes under 'ADDED' and 'FIXED' sections, along with a 'v0.6.35 - 2025-11-06' section. At the bottom of the modal is a 'Okay, Let's Go!' button. In the bottom right corner of the main window, there is a callout box with the text: 'ADDED Image generation system received a comprehensive overhaul with major new capabilities including full image editing support allowing users to modify existing images using text prompts with OpenAI, Gemini, or ComfyUI engines, adding Gemini 2.5 Flash Image (Nano Banana) support, Qwen Image Edit integration, resolution of base64-encoded image display issues, streamlined AUTOMATIC1111 configuration by consolidating parameters into a flexible JSON parameters field, and enhanced UI with a code editor modal for ComfyUI workflow management. #17434, #16976, Commit, Commit, Commit, Commit'.

Erfolgreicher Login:

Nach der ersten Anmeldung wird man dann erfolgreich in der Anwendung begrüßt und die letzten Änderungen werden im Changelog aufgelistet.

Hier sieht man auch, dass es eine aktive Community gibt, die dieses Produkt weiterentwickelt (Menge und Inhalt der Changes).

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?

The screenshot shows the 'Connections' section of the Ollama AI Settings page. On the left, there's a sidebar with various options like 'Open WebUI', 'New Chat', 'Search', 'Notes', 'Workspace', 'Folders', and 'Chats'. The main area has a header 'General' and sections for 'OpenAI API' and 'Ollama API'. Under 'OpenAI API', it says 'Manage OpenAI API Connections' with the URL 'https://api.openai.com/v1'. Under 'Ollama API', it says 'Manage Ollama API Connections' with the URL 'http://localhost:11434'. There are also sections for 'Direct Connections' and 'Cache Base Model List'. On the right side, there are several toggle switches and buttons, with some being turned on (green) and others off (grey). At the bottom right is a 'Save' button with a circular arrow icon.

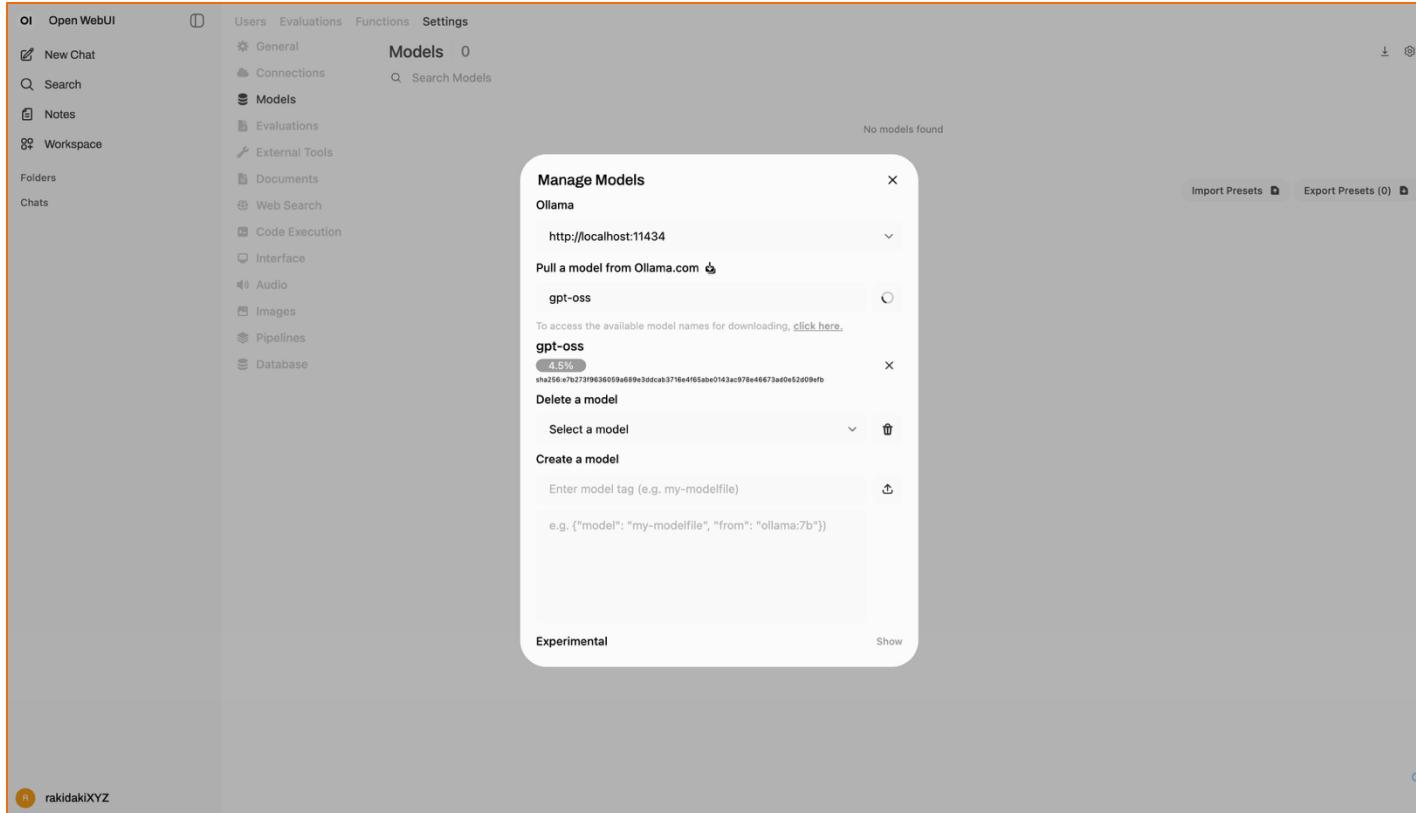
Einstellungen

Über das Profil-Icon rechts oben kann man auf die Administration und die Einstellungen zugreifen.

Hier sieht man u.a. die Connections – also die Einstellungen zu OpenAI (per Default schon da) und die Verbindung zu Ollama (hier im gleichen Container – mit einer anderen Port Adresse).

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

 Lokale Umgebung + online erreichbar?



Modelle (Sprachmodelle auswählen über Ollama)

Über den Eintrag Models kann man jetzt mit Ollama Sprachmodelle für die Nutzung in unserer lokalen Chat-KI Lösung auswählen und herunterladen.

Idealerweise über die Webseite von Ollama (Ollama.com) die Modelle suchen, den Namen kopieren und hier dann eingeben, um das jeweilige Modell herunterzuladen.

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)



Lokale Umgebung + online erreichbar?

The screenshot shows the Ollama web interface. On the left, there's a sidebar with options like 'Open WebUI', 'New Chat', 'Search', 'Notes', 'Workspace', 'Folders', and 'Chats'. The main area has a title 'gpt-4o-mini-search-preview' and a search bar with placeholder text 'How can I help you today?'. Below the search bar, there's a microphone icon and a 'Suggested' section with three items: 'Give me ideas for what to do with my kids' art', 'Overcome procrastination give me tips', and 'Show me a code snippet of a website's sticky header'. At the bottom left, there's a user icon with the name 'rakidakixyz'.

So, los geht's:

Wenn man die Verbindung zu den „großen“ LLMs über deren API Keys hergestellt hat und/ oder mit lokalen Sprachmodellen arbeitsfähig ist über Ollama (Download der Modelle), kann man wie bei OpenAI loslegen

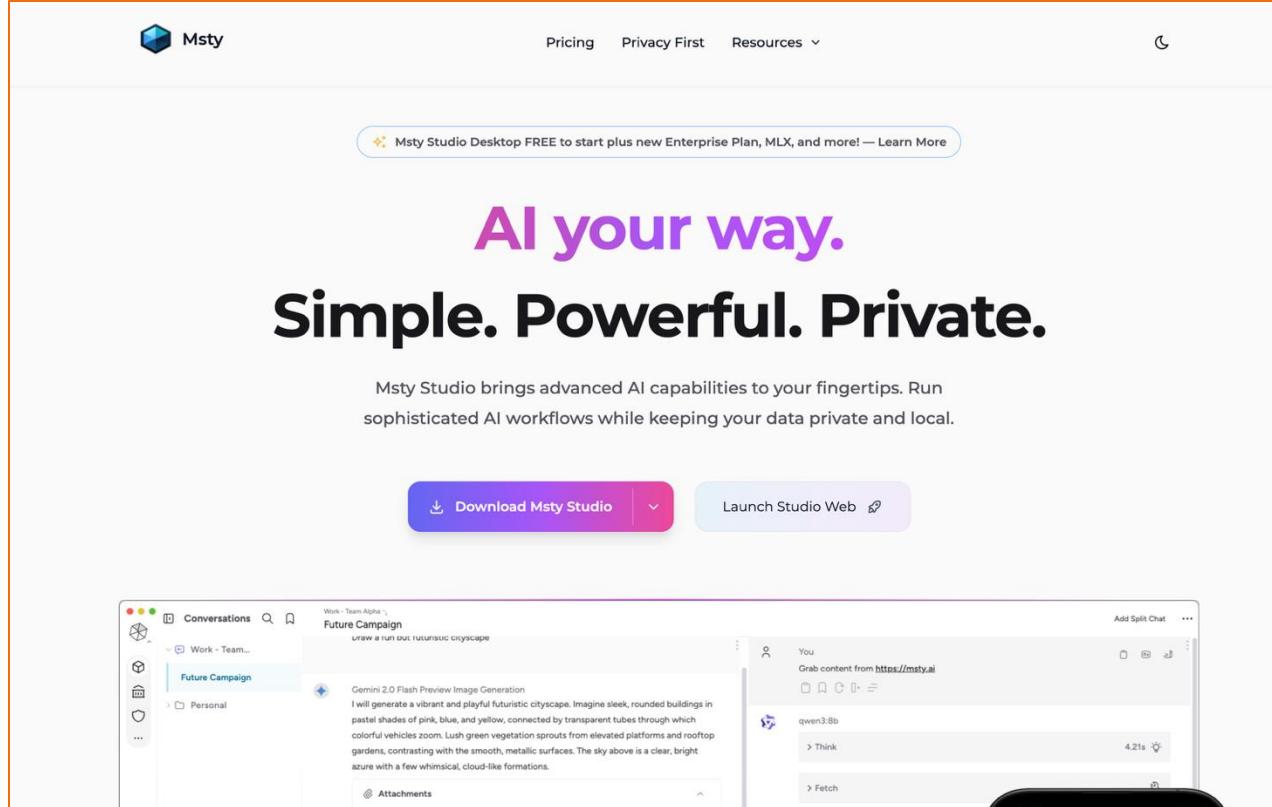
Chat

Lokale KI Umgebung mit Msty AI App, Jan AI, ...

Installationspaket für Windows, Mac, ..

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

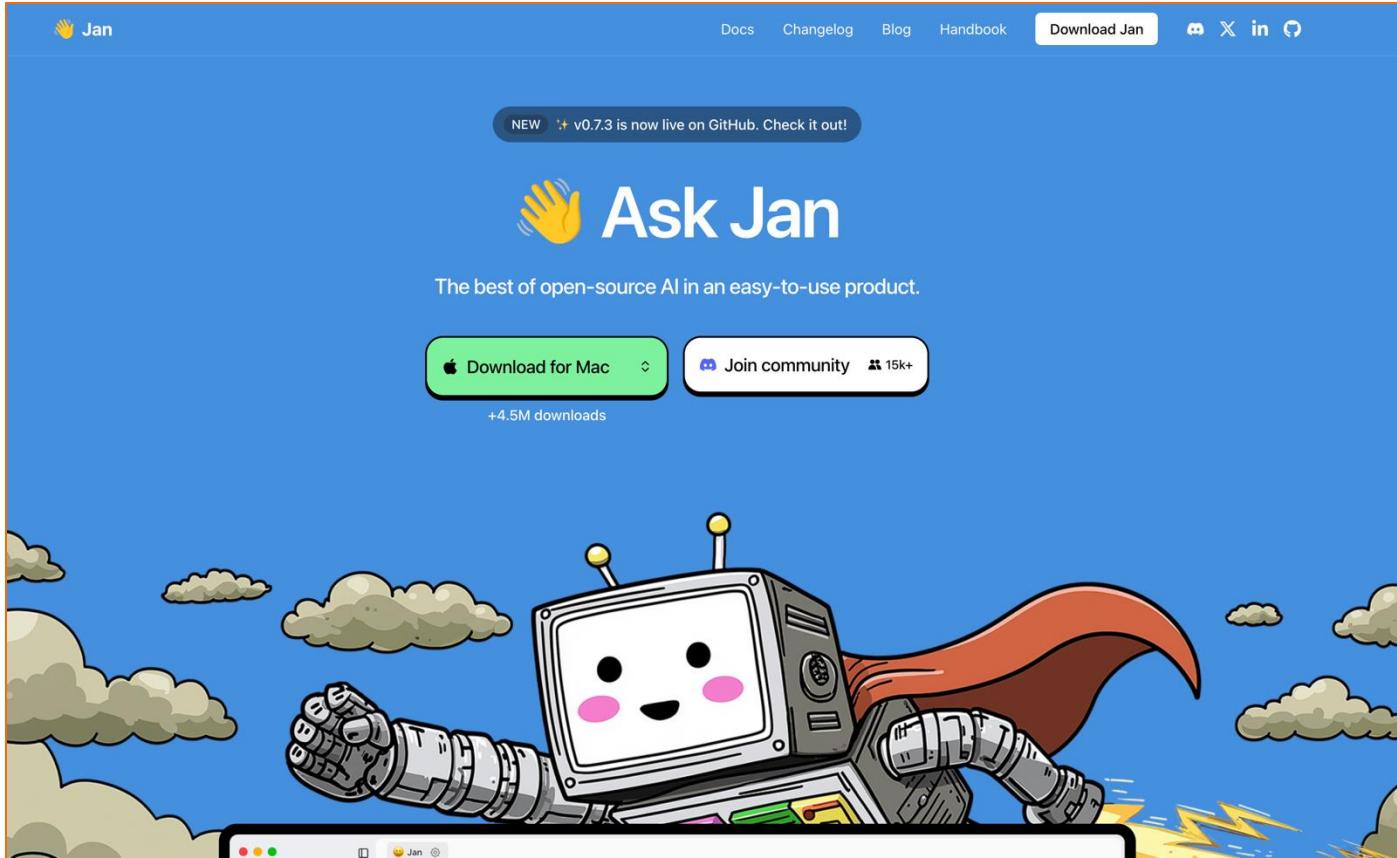
 Msty AI App



<https://msty.ai/>

Lokale KI-Umgebung (RAG, Chatbot, KI-Entwicklung)

🤔 Jan AI



<https://www.jan.ai/>



66

Ralf Kluth

@ ralf@from-scratch.ai

X: fromscratchai

LinkedIn: <https://www.linkedin.com/in/rakidakixyz>

99