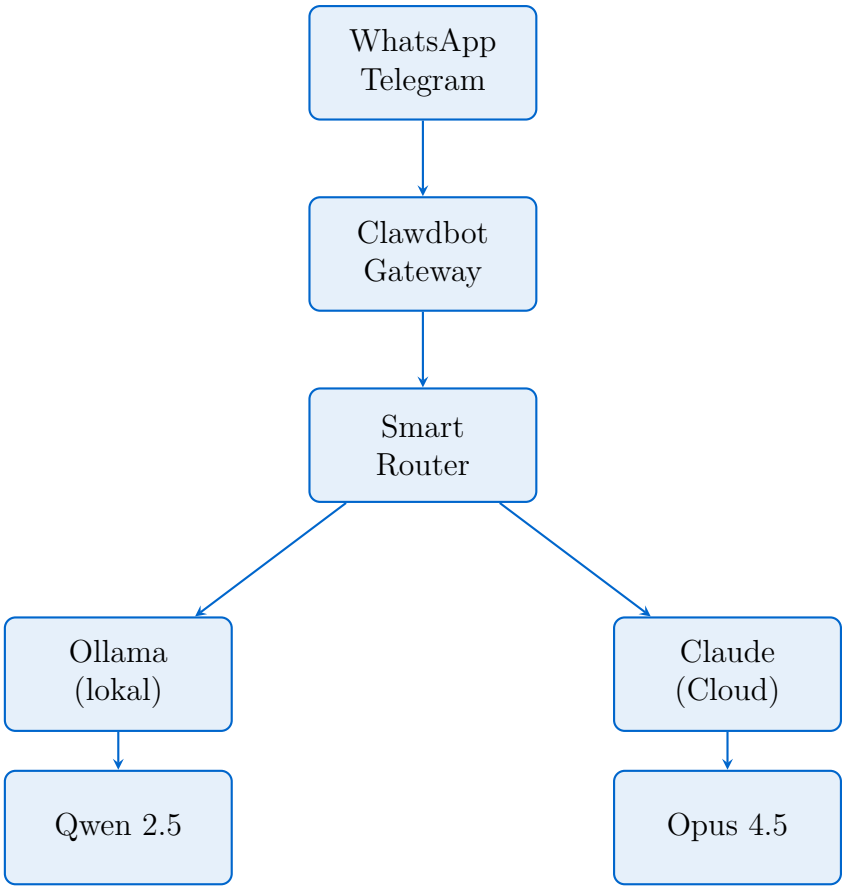
Ollama + Clawdbot

Hybrides LLM-System

Installation, Konfiguration & Best Practices



Version 1.0.0

27. Januar 2026

Dr. Jo

Inhaltsverzeichnis

Kapitel 1

Übersicht

1.1 Was ist Ollama?

Ollama ist eine lokale LLM-Runtime, die große Sprachmodelle wie Qwen, Llama und Mistral direkt auf deinem Mac ausführt – komplett **offline** und **kostenlos**.

Hauptvorteile

- **Offline-Betrieb**: Funktioniert ohne Internetverbindung
- **Kostenlos**: Keine API-Gebühren
- **Privacy**: Daten verlassen niemals deinen Mac
- **Schnell**: Keine Netzwerk-Latenz

1.2 Warum ein Hybrid-System?

Die Kombination von **Ollama** (lokal) und **Claude API** (Cloud) bietet das Beste aus beiden Welten:

Aspekt	Ollama (lokal)	Claude (Cloud)
Kosten	Kostenlos	Pay-per-Use
Geschwindigkeit	Sehr schnell	Latenz durch Netzwerk
Privacy	Maximale Privacy	Daten in der Cloud
Qualität	Gut (14B Modell)	Exzellent (Opus)
Offline	Ja	Nein
Komplexität	Einfache Aufgaben	Komplexe Analysen

Tabelle 1.1: Vergleich Ollama vs. Claude

1.2.1 Anwendungsfälle

Ollama nutzen für:

- WhatsApp-Nachrichten beantworten

- Einfache Übersetzungen
- Textzusammenfassungen
- Schnelle Berechnungen
- Sensible Daten (Privacy)

Claude nutzen für:

- Komplexe Code-Analysen
- Wissenschaftliche Recherchen
- Detaillierte Dokumente erstellen
- Strategische Beratung
- Multi-Step-Reasoning

1.3 Architektur

Das hybride System verwendet einen **Smart Router**, der eingehende Anfragen intelligent zwischen Ollama und Claude verteilt:

Routing-Logik

Einfache Anfragen (kurz, faktisch, schnell) → Ollama
Komplexe Anfragen (lang, analytisch, kreativ) → Claude
Sensible Daten → Immer Ollama

Kapitel 2

Voraussetzungen

2.1 Hardware

Komponente	Anforderung
Betriebssystem	macOS 12.0 oder höher
Prozessor	Apple Silicon (M1/M2/M3) oder Intel
RAM	Mindestens 8 GB (16 GB empfohlen)
Freier Speicher	10–50 GB (je nach Modell)

Tabelle 2.1: Hardware-Anforderungen

Warnung

RAM-Empfehlung: Für optimale Performance wird mindestens 16 GB RAM empfohlen. Mit 8 GB solltest du kleinere Modelle (7B) nutzen.

2.2 Software

Erforderlich:

Homebrew installiert

Node.js v18+ installiert

Clawdbot bereits eingerichtet

2.2.1 Voraussetzungen prüfen

```
# Homebrew installiert?
brew --version

# Node.js installiert?
node --version

# Clawdbot l uft?
```

```
ps aux | grep clawdbot
```

Listing 2.1: Prüf-Befehle

Kapitel 3

Installation

3.1 Ollama installieren

3.1.1 Via Homebrew (empfohlen)

```
brew install ollama
```

Listing 3.1: Ollama Installation

3.1.2 Alternativer Download

1. Besuche: https://ollama.com/download
2. Lade **Ollama für macOS** herunter
3. Öffne die `.dmg`-Datei
4. Ziehe Ollama in den Programme-Ordner

3.2 Ollama Service starten

```
# Im Hintergrund starten
ollama serve &

# Oder als LaunchAgent (automatischer Start)
brew services start ollama
```

Listing 3.2: Service starten

Verifizierung

Prüfe, ob Ollama läuft:

```
curl http://localhost:11434/api/version
```

Erwartete Ausgabe: `{"version:"0.15.2"}`

3.3 Qwen Modell herunterladen

3.3.1 Modell-Auswahl

Modell	Größe	RAM	Speed	Qualität	Anwendung
qwen2.5:3b	2 GB	8 GB			WhatsApp, Einfach
qwen2.5:7b	4.7 GB	16 GB			Allgemein
qwen2.5:14b	8.9 GB	32 GB			Standard
qwen2.5:32b	20 GB	64 GB			Beste Qualität

Tabelle 3.1: Qwen Modell-Varianten

3.3.2 Download-Befehle

```
# Qwen 2.5 14B (empfohlen)
ollama pull qwen2.5:14b

# Kleinere Varianten
ollama pull qwen2.5:7b    # F r 16 GB RAM
ollama pull qwen2.5:3b    # F r 8 GB RAM

# Gr   ere Variante
ollama pull qwen2.5:32b   # F r 64 GB RAM
```

Listing 3.3: Modell herunterladen

3.4 Installation verifizieren

```
ollama list
```

Listing 3.4: Installierte Modelle prüfen

Erwartete Ausgabe:

```
NAME              ID              SIZE      MODIFIED
qwen2.5:14b       abc123def456    8.9 GB    2 minutes ago
```

Kapitel 4

Konfiguration

4.1 Clawdbot konfigurieren

Öffne die Clawdbot-Konfiguration:

```
nano /Users/mac/.clawdbot/clawdbot.json
```

4.2 Hybride LLM-Konfiguration

Füge folgende Konfiguration ein:

```
{
  "llm": {
    "provider": "hybrid",
    "hybrid": {
      "router": "smart",
      "providers": {
        "ollama": {
          "baseURL": "http://localhost:11434",
          "model": "qwen2.5:14b",
          "enabled": true,
          "priority": 1,
          "useFor": ["simple", "translation",
                     "summarize", "whatsapp"]
        },
        "claude": {
          "apiKey": "DEIN_CLAUDE_API_KEY",
          "model": "claude-opus-4-20250514",
          "enabled": true,
          "priority": 2,
          "useFor": ["complex", "code",
                     "analysis", "research"]
        }
      },
      "routing": {
        "simple": "ollama",
        "complex": "claude",
        "translation": "ollama",
        "code": "claude",
        "summarize": "ollama",
        "analysis": "claude",
```

```
        "research": "claude",
        "whatsapp": "ollama",
        "default": "claude"
      }
    }
  }
}
```

Listing 4.1: Hybrid-LLM-Konfiguration

4.3 Gateway neu starten

```
launchctl kickstart -k gui/501/com.clawdbot.gateway
```

Kapitel 5

Qwen Plugin

5.1 Plugin erstellen

```
mkdir -p /Users/mac/.clawdbot/skills/qwen
```

Listing 5.1: Qwen Skill-Verzeichnis erstellen

5.2 SKILL.md erstellen

```
cat > /Users/mac/.clawdbot/skills/qwen/SKILL.md << 'SKILL'
---
name: qwen
description: Lokales Qwen 2.5 LLM via Ollama
license: MIT
---

# Qwen Ollama Integration

Nutzt das lokal laufende Qwen 2.5 Modell.

## Funktionen
- Offline-Betrieb
- Kostenlos
- Schnell
- Privacy-freundlich

## Befehle
/qwen [frage] - Nutze Qwen
/qwen-info - Zeige Status
SKILL
```

5.3 package.json erstellen

```
{
  "name": "@clawdbot-skills/qwen",
  "version": "1.0.0",
  "description": "Qwen 2.5 Ollama Integration",
```

```
  "clawdbot": {
    "type": "skill",
    "category": "llm",
    "tags": ["llm", "ollama", "qwen"],
    "author": "Clawdbot"
  }
}
```

Listing 5.2: Skill Metadaten

5.4 Skill aktivieren

In `clawdbot.json` hinzufügen:

```
{
  "skills": {
    "enabled": true,
    "paths": ["/Users/mac/.clawdbot/skills"],
    "active": ["qwen"]
  }
}
```

Kapitel 6

Testen

6.1 Ollama direkt testen

```
curl http://localhost:11434/api/generate -d '{
  "model": "qwen2.5:14b",
  "prompt": "Was ist 2+2?",
  "stream": false
}'
```

Listing 6.1: Test mit curl

Erwartete Antwort:

```
{
  "response": "2+2 ergibt 4.",
  "done": true
}
```

6.2 Interaktiver Test

```
ollama run qwen2.5:14b "Was ist 5+5?"
```

6.3 Clawdbot-Integration testen

6.3.1 Via WhatsApp

Einfache Anfrage (Ollama)

Du: Hallo, was ist 10-3?
Bot: (via Qwen) 10-3 ergibt 7.

Komplexe Anfrage (Claude)

Du: Schreibe eine detaillierte Analyse der Blockchain-Technologie
Bot: (via Claude) [Ausführliche Antwort...]

Kapitel 7

Troubleshooting

7.1 Connection refused

Problem

Fehler: `Connection refused` beim Zugriff auf Ollama

Ursache: Ollama läuft nicht
Lösung:

```
# Ollama starten
ollama serve &

# Oder als Service
brew services start ollama

# Status prüfen
curl http://localhost:11434/api/version
```

7.2 Model not found

Problem

Fehler: `Model 'qwen2.5:14b' not found`

Ursache: Modell nicht heruntergeladen
Lösung:

```
# Modell herunterladen
ollama pull qwen2.5:14b

# Installierte Modelle prüfen
ollama list
```

7.3 Ollama ist langsam

Performance-Probleme

Ollama antwortet sehr langsam oder stockt

Lösungen:

1. **Kleineres Modell nutzen**

```
ollama pull qwen2.5:7b
```

2. **Andere Apps schließen**

```
# RAM-Nutzung pr fen
top -o mem
```

3. **Intel Mac**: Nutze 3B oder 7B Modell

7.4 Routing funktioniert nicht

Problem

Clawdbot nutzt immer Claude statt Ollama

Lösung: Routing-Konfiguration prüfen

```
{
  "routing": {
    "whatsapp": "ollama",  // Muss auf "ollama" stehen
    "default": "claude"
  }
}
```

Kapitel 8

Erweiterte Nutzung

8.1 Weitere Modelle

8.1.1 Llama 3.1 (Meta)

```
ollama pull llama3.1:8b
```

8.1.2 Mistral (MistralAI)

```
ollama pull mistral:7b
```

8.1.3 CodeLlama (Code-spezialisiert)

```
ollama pull codellama:13b
```

8.2 Multi-Modell-Setup

Verschiedene Modelle für verschiedene Aufgaben:

```
{
  "providers": {
    "ollama-qwen": {
      "model": "qwen2.5:14b",
      "useFor": ["whatsapp", "translation"]
    },
    "ollama-codellama": {
      "model": "codellama:13b",
      "useFor": ["code", "programming"]
    },
    "ollama-llama": {
      "model": "llama3.1:8b",
      "useFor": ["simple", "summarize"]
    },
    "claude": {
      "model": "claude-opus-4-20250514",
      "useFor": ["complex", "research"]
```

```
    }
  }
}
```

Listing 8.1: Multi-Modell-Konfiguration

8.3 Performance-Optimierung

8.3.1 Ollama-Konfiguration

```
mkdir -p ~/.ollama
cat > ~/.ollama/config.json << 'CONFIG'
{
  "num_parallel": 4,
  "num_ctx": 8192,
  "num_gpu": 1,
  "num_thread": 8
}
CONFIG
```

Listing 8.2: Ollama Config erstellen

8.3.2 Parameter-Erklärung

Parameter	Bereich	Beschreibung
num_parallel	1–8	Parallele Anfragen
num_ctx	2048–32768	Kontext-Größe (Tokens)
num_thread	1–16	CPU-Threads (= Anzahl Kerne)

Tabelle 8.1: Ollama Performance-Parameter

Kapitel 9

Sicherheit & Privacy

9.1 Warum Ollama sicherer ist

Privacy-Vorteile

- **Keine Cloud**: Daten verlassen niemals deinen Mac
- **Kein Tracking**: Ollama loggt keine Anfragen
- **Open Source**: Code ist öffentlich einsehbar
- **Offline**: Funktioniert ohne Internet

9.2 Best Practices

Sensible Daten: Immer Ollama statt Claude nutzen

Firmen-Daten: Routing auf `ollama` setzen

Passwörter: Niemals in LLM-Prompts eingeben

Logs: Regelmäßig löschen

9.3 Logs löschen

```
# Ollama-Logs l schen
rm -rf ~/.ollama/logs/*

# Clawdbot-Logs l schen
rm -rf /Users/mac/.clawdbot/logs/*.log
```

Anhang A

Befehls-Referenz

A.1 Ollama-Befehle

Befehl	Beschreibung
`brew services start ollama`	Service starten
`brew services stop ollama`	Service stoppen
`brew services list`	Service-Status
`ollama list`	Modelle auflisten
`ollama pull <model>`	Modell herunterladen
`ollama run <model> "text"`	Modell testen
`ollama rm <model>`	Modell löschen
`curl localhost:11434/api/version`	API-Version

Tabelle A.1: Ollama Befehls-Übersicht

A.2 Clawdbot-Befehle

Befehl	Beschreibung
`launchctl kickstart -k ...`	Gateway neu starten
`launchctl stop ...`	Gateway stoppen
`tail -f logs/gateway.log`	Logs anzeigen
`cat clawdbot.json | grep llm`	Config prüfen
`ls -la skills/`	Skills auflisten

Tabelle A.2: Clawdbot Befehls-Übersicht

Anhang B

Installations-Checklist

B.1 Vor der Installation

macOS 12.0+

Homebrew installiert

Node.js 18+ installiert

Clawdbot läuft

Mindestens 8 GB RAM

10–50 GB freier Speicher

B.2 Installation

Ollama installiert

Ollama Service läuft

Qwen-Modell heruntergeladen

Hybrid-Config eingefügt

Qwen Skill erstellt

Gateway neu gestartet

B.3 Tests

Ollama API erreichbar

Qwen-Modell läuft

Clawdbot nutzt Ollama

Hybrides Routing funktioniert

Logs zeigen keine Fehler