Architektur- und Funktionsvergleich zwischen dem OpenCaseLaw KI-Agenten und Claude Desktop mit Connectors zu den OpenCaseLaw-MCP-Tools

Die Frage „Wo finde ich die System-Prompts?“ gibt Anlass, die grundlegende Architektur des OpenCaseLaw KI-Agenten im Vergleich zu Claude Desktop mit MCP-Connectors kurz darzustellen.

Open Case Law KI-Agent

Anwender wählen in der Benutzeroberfläche einen Betriebsmodus. Jeder Modus ist einem eigenen System-Prompt zugeordnet, den das Backend in den Konversationskontext einfügt.

Dieser System-Prompt fungiert als persistente Richtlinie für die gesamte agentische Reasoning-Schleife. Er bleibt über jede Iteration hinweg präsent und beeinflusst kontinuierlich:

- wie das Modell den aktuellen Zustand analysiert,
- ob und welche Tools (OpenCaseLaw- und Zefix-MCP-Tools) aufgerufen werden,
- wie die Tool-Ergebnisse interpretiert werden und
- wann das Modell entscheidet, dass ausreichend Informationen vorliegen, um eine abschliessende Antwort zu generieren.

Die System-Prompts sind als interner Backend-Mechanismus implementiert und für den Nutzer nicht direkt zugänglich.

Claude Desktop mit MCP-Connectors

Claude Desktop unterstützt MCP-Tools über zwei Hauptwege:

- Lokale Server (stdio), konfiguriert über `claude_desktop_config.json` oder Desktop Extensions.
- Remote-Server über Custom Connectors (Streamable HTTP), die über die Infrastruktur von Anthropic laufen.

Sobald die Tools verbunden sind, stehen sie in der Konversation zur Verfügung. Claude entscheidet (je nach Konfiguration nur nach expliziter Nutzerfreigabe), ob und wann sie aufgerufen werden. Die Architektur ist primär tool-zentriert: MCP-Server ermöglichen die Verwendung der OpenCaseLaw-MCP-Tools, während die zentralen Verhaltensrichtlinien weitgehend unter der Kontrolle von Anthropic bleiben.

Claude Desktop bietet begrenzte Instruktionsmechanismen:

- Globale Anweisungen sowie Ordner-/Projekt-Anweisungen (besonders in Cowork),
- projektbezogene Custom Instructions,
- Skills und Plugins, die zusätzliche Hinweise mit Tools bündeln können.

Diese sind jedoch eher leichte Kontext-Einschübe und keine vollständigen, modus-spezifischen System-Prompts, die die Entscheidungspolitik des Agenten für die Dauer einer Sitzung neu definieren.

Kann der Anwender auf dem Claude Desktop die Betriebsmodi auf die gleiche Art und Weise auswählen wie beim OpenCaseLaw KI-Agenten?

Nein. Claude Desktop bietet keinen UI-Selektor für verschiedene Betriebsmodi, mit dem ein entsprechender, vollständiger System-Prompt in den Backend-Kontext der jeweiligen agentischen Reasoning-Schleife eingefügt würde.

Claude Desktop bietet kein Äquivalent zum Mechanismus des OpenCaseLaw KI-Agenten *„Modus X auswählen → Backend fügt System-Prompt X ein → die gesamte Reasoning-Schleife (Tool-Auswahl, Analyse, Abbruchkriterien) wird durch diesen Prompt gesteuert.“*

Die nächstliegenden Annäherungen im Claude-Ökosystem sind:

- Wechsel zwischen verschiedenen Projects (jedes mit eigenen Instructions),
- Nutzung der Permission-Modes in Cowork (die vor allem das Freigabeverhalten von Tools steuern und nicht die vollständige Entscheidungspolitik),
- Aufbau eines eigenen Agenten über die Anthropic API, bei dem der System-Prompt vollständig kontrolliert werden kann (dies entspricht jedoch nicht mehr der Standard-Erfahrung von Claude Desktop).