

Subagents & Skills — der Spickzettel

SPICKZETTEL · STAND: JULI 2026

Die Entscheidungsmatrix

Kriterium	Subagent	Skill	MCP-Server
Was es ist	Eigener Agent mit System-Prompt und Werkzeugrechten	Wiederverwendbare Anleitung (Markdown)	Werkzeug-Anschluss an externe Systeme (APIs, Datenbanken)
Eigener Kontext?	Ja — eigenes Kontextfenster, flutet den Hauptkontext nicht	Nein — wird bei Bedarf in die Session geladen	Nein — liefert Werkzeuge, keinen Kontext
Wiederholbarer Workflow?	Nein — abgegrenzte Teilaufgabe	Ja: «so auditieren wir», «so schreiben wir Posts»	Nein — Fähigkeiten statt Abläufe
Externes System?	Nein	Nein	Ja: Tickets lesen, Datenbanken abfragen, Deploys auslösen
Aufruf	Hauptsession delegiert die Teilaufgabe	Agent lädt selbst — Trigger ist das <code>description</code> -Feld	Gibt dem Agenten Fähigkeiten, die er nicht hat
Typischer Einsatz	Recherche, Review, Suche	Audit, Release-Checkliste, Review-Ritual	Tickets, Deploys, Datenbanken
Merkregel	Arbeitsteilung	Wissen	Anschluss an die Welt

Faustregeln

- **Skills sind Wissen, Subagents sind Arbeitsteilung, MCP ist Anschluss an die Welt.**
- **Eine Rolle pro Subagent.** «Recherchiere und implementiere und teste» ist kein Subagent — das ist die Hauptsession.
- **Werkzeugrechte beschneiden.** Ein Reviewer, der nur lesen darf, kann seine eigene Kritik nicht «schnell fixen».
- `description` **als Trigger schreiben** («Use when ...»), nicht als Zusammenfassung — es entscheidet, ob der Skill im richtigen Moment geladen wird.

Dateiorte

Skills: `.claude/skills/<name>/SKILL.md` · **Subagents:** `.claude/agents/<name>.md` — beides Markdown im Projekt, beides in je rund 20 Minuten gebaut.

Das komplette Tutorial: <https://halvic.ch/de/blog/claude-code-subagents-skills-tutorial>
