ARCHITEKTUR-FALLSTUDIE 7B / 01

KI-Orchestrierung und unabhängige Qualitätssicherung

Eine kontrollierte Architektur für die Zusammenarbeit von Mensch, KI-Assistenten, Entwicklungswerkzeugen und Prüfern.

Ausgangslage und Ziel

In KI-gestützten Entwicklungsprozessen ist die eigentliche Herausforderung nicht das Generieren von Programmcode, sondern die eindeutige Verteilung von Verantwortung. Ohne kontrollierte Übergaben drohen doppelte Schreibzugriffe, unbelegte Erfolgsmeldungen und uneinheitliche Prüfstände.

Ziel war eine nachvollziehbare Orchestrierung mit einer fachlichen Freigabeinstanz, genau einem umsetzenden Agenten bei Bedarf und getrenntem Prüfnachweis.

Lösungsarchitektur

Auftraggeber: definiert Ziel und erteilt Freigaben. ChatGPT: Architektur, Arbeitskoordination und eigenständige Abnahme. Desktop Commander: autorisierte Verbindung zum Windows-System. Claude Code: optionaler Implementierungspartner in isoliertem Projekt. Poracode/Crossagents: ausschließlich im Proof of Concept getestete Subagenten-Übergabe.

Die Architektur trennt Anweisung, Ausführung, Bericht und unabhängige Kontrollinstanz. Projektverzeichnisse und Veröffentlichungsgates sind explizit Teil des Designs.

KOMPONENTEN- UND VERANTWORTUNGSFLUSS

Auftrag + Freigabe	Architektur / Delegation	Isolierte Umsetzung	Unabhängige Prüfung

Entwurfsprinzip

Trennung von Entscheidung, Ausführung und Abnahme. Die Architektur ist gezielt begrenzt: Sie bewahrt bestehende Systeme und macht Verantwortlichkeit überprüfbar, statt maximale Autonomie ungeprüft zu behaupten.

ARCHITEKTUR-FALLSTUDIE 7B / 02

Realisierung und Integration

Ausgewählte technische Schritte, Schnittstellen und Entscheidungen – ohne unnötige Implementierungsdetails aus geschützten Systemen.

Technische Umsetzung

Im isolierten Node-Projekt wurde absichtlich ein fehlerhafter geradzahliger Filter geschaffen. Ein einzelner Claude-Thread korrigierte eine eng begrenzte Funktion und führte anschließend vier Tests erfolgreich aus.

Über Crossagents wurde genau ein separater Claude-Reviewagent beauftragt. Der Review ging automatisch an den ursprünglichen Arbeitslauf zurück. Das wurde praktisch beobachtet, ist aber kein Beweis einer zweiten Modellfamilie oder einer technisch erzwungenen Leseisolation.

Qualitätssicherung und Abnahmepunkte

Ein zweiter, von ChatGPT konzipierter deterministischer Prüfablauf überprüfte die reparierte Funktion anhand von 1.000 generierten Arrays, Negativwerten, Nullfällen und einem Mutationstest. Das war eine zusätzliche unabhängige Auswertung und keine weitere Crossagents-Providerintegration.

Die Beweiskette umfasst vor/nach Fehlerzustände, 4/4 ursprüngliche Tests, Rückgabe des Reviews und die nachgeschaltete Testsequenz.

Warum diese Architektur?

Sie unterstützt inkrementelle Umsetzung mit kleinen, beobachtbaren Änderungen. Jede Stufe besitzt ein konkretes Prüfkriterium. Fehler oder Grenzen werden dokumentiert, nicht durch eine positive Statusmeldung überdeckt.

Prüfstrategie

- Baseline sichern und den Umfang einer Änderung begrenzen.
- Den tatsächlich verwendeten Integrationspfad statt nur einer isolierten Komponente kontrollieren.
- Fehlerfälle und unerlaubte Zustände bewusst testen.
- Ergebnis, Restunsicherheiten und Wiederaufnahme reproduzierbar dokumentieren.

Belege, Risiken und Lernwert

Ein belastbarer Referenzbericht unterscheidet echte Beobachtung, isolierten Test und noch nicht erreichte Betriebsziele.

Sicherheitsgrenzen und offene Punkte

Genau ein schreibender Akteur pro Änderung; isolierter Testordner; keine Umkonfiguration produktiver Projekte; klare Kosten- und Berechtigungsgrenzen. Der Crossagents-Subagent war als „full-access“ gemeldet; Read-only war dort nur eine textliche Anweisung.

Weder ein autonomer fremder KI-Provider im Crossagents-Verbund noch die automatische Wiederaufnahme eines bestehenden ChatGPT-Chats wurden nachgewiesen.

Wirkung und eigener Beitrag

Die Fallstudie illustriert ein überprüfbares Orchestrierungsmuster: Rollen und Autorität vor Werkzeugaufrufen definieren, Fehler reproduzieren, klein beheben, unabhängig testen und Grenzen offen dokumentieren.

Mein Beitrag: Architekturkonzept, Freigabe- und Projekttrennung, Koordination der KI-gestützten Implementierung und Qualitätsabnahme; nicht die Behauptung einer vollständig manuell geschriebenen Agentenplattform.

Übertragbarkeit

Das Vorgehen ist auf AI-Integration, interne Automatisierung und Developer-Experience-Plattformen übertragbar: klare Schnittstellen, minimale Rechte, dokumentierte Zustände, Kostenbewusstsein und reproduzierbare Kontrollen. Ein neues Zielsystem erfordert selbstverständlich eine eigene Sicherheits- und Abnahmeprüfung.

Lessons learned

Ein aussagekräftiger Architektur-Nachweis zeigt nicht nur einen erfolgreichen Endzustand. Er macht auch den Weg zur Fehlerentdeckung, die Grenzen des Tests und die Verantwortung für reale Freigaben sichtbar. Die Dokumentation ist deshalb Teil des technischen Ergebnisses und kein nachträgliches Marketingmaterial.

Quellenbasis: interne Projekthaltepunkte vom 22.–25.11.2026, technische Testprotokolle und Portfolio-Register. In dieser Bewerbungsfassung bewusst ohne interne Pfade, Tokens, private Repository-URLs oder personenbezogene Systemdaten.