MCP-Server mit SAP ERP

Wie KI Ihr SAP-System analysiert, absichert und weiterentwickelt – ohne Cloud, ohne BTP, ohne Joule

Consiness GmbH & Co. KG
Februar 2026

Was ist ein MCP-Server – und warum sollte Ihr SAP-System einen haben?

SAP-Systeme sind das Rückgrat vieler Unternehmen. Aber wer kennt sein System wirklich? Welche Custom Reports laufen, welche davon sind Sicherheitsrisiken, welche User melden sich seit Jahren nicht mehr an? Die meisten IT-Abteilungen können diese Fragen nicht auf Knopfdruck beantworten. Ein MCP-Server ändert das.

Das Problem: KI kann vieles – aber nicht in Ihr SAP

Große Sprachmodelle wie ChatGPT oder Claude von Anthropic sind beeindruckend, wenn es um Textanalyse, Code-Verständnis ~~oder Strategieberatung~~ geht. Aber sie haben eine fundamentale Einschränkung: Sie können nicht per se auf Ihre Systeme zugreifen. Ein Sprachmodell kann Ihnen erklären, wie man einen SAP-Dump analysiert. Es kann aber nicht in Ihre ST22 schauen und Ihnen sagen, welche Dumps gestern aufgetreten sind.

Genau hier setzt das Model Context Protocol (MCP) an.

MCP: Die Brücke zwischen KI und SAP

MCP ist ein offener Standard von Anthropic, der es KI-Modellen ermöglicht, strukturiert mit externen Systemen zu kommunizieren. Ein MCP-Server für SAP stellt dabei die Verbindung her: Er übersetzt die Anfragen der KI in SAP-verständliche Aufrufe – über die ADT-API (ABAP Development Tools) oder RFC-Schnittstellen – und liefert die Ergebnisse zurück.

Die Architektur ist überraschend schlank: Die KI (etwa Claude Desktop oder Cursor IDE) kommuniziert mit dem MCP-Server, der wiederum über HTTPS oder RFC mit dem SAP-System spricht. Der MCP-Server läuft lokal oder im Firmennetz – kein Cloud-Zwang, keine BTP-Abhängigkeit, keine Joule-Lizenz.

Warum das für SAP-on-Prem relevant ist

Viele Unternehmen betreiben SAP on-Premise und haben weder die Möglichkeit noch den Wunsch, kurzfristig auf die SAP Business Technology Platform zu migrieren. Die KI-Werkzeuge, die SAP selbst anbietet – allen voran Joule – setzen aber genau das voraus. Für diese Unternehmen entsteht eine Lücke: Sie sehen das Potenzial von KI im SAP-Umfeld, können es aber mit Bordmitteln nicht nutzen.

MCP schließt diese Lücke. Der Ansatz funktioniert mit jedem SAP-System, das eine ADT- oder RFC-Schnittstelle bietet – also praktisch jedem modernen SAP-System, ob ECC, S/4HANA on-Premise oder Private Cloud.

Dabei ist Transparenz wichtig: Ein MCP-Server ist ein mächtiges Werkzeug – und wie jedes mächtige Werkzeug bringt es Risiken mit. SAP-Daten fließen an ein Sprachmodell, MCP-Server benötigen Systemzugriff, und nicht jede Quelle für MCP-Server ist vertrauenswürdig. Auf diese Sicherheitsaspekte gehe ich im letzten Kapitel ausführlich ein.

Was damit möglich wird

In den folgenden Kapiteln zeige ich anhand realer Praxisbeispiele, was ein MCP-Server im SAP-Alltag leisten kann: vollständige Systeminventur aller Custom-Entwicklungen auf Knopfdruck, automatisierte Security- und Code-Quality-Audits mit priorisierten Handlungsempfehlungen, KI-gestützte Dump-Analyse mit professionellem Reporting, aktive

ABAP-Entwicklung – von der Fehler-Diagnose bis zum dokumentierten Fix, sowie Unterstützung für funktionale Berater bei der Analyse von Customizing und Stammdaten.

All das ohne Cloud-Migration, ohne neue SAP-Lizenzen und mit voller Kontrolle über Ihre Daten.

Übersicht der weiteren Kapitel

Kapitel 2: SAP-Systemanalyse per KI – Inventur auf Knopfdruck. Wie die KI in Minuten eine vollständige Bestandsaufnahme aller Custom Reports und Benutzer erstellt.

Kapitel 3: Security-Audit mit KI – Schwachstellen finden, bevor es andere tun. Automatisierte Code-Analyse, Risikobewertung und Dump-Analyse als Präventionswerkzeug.

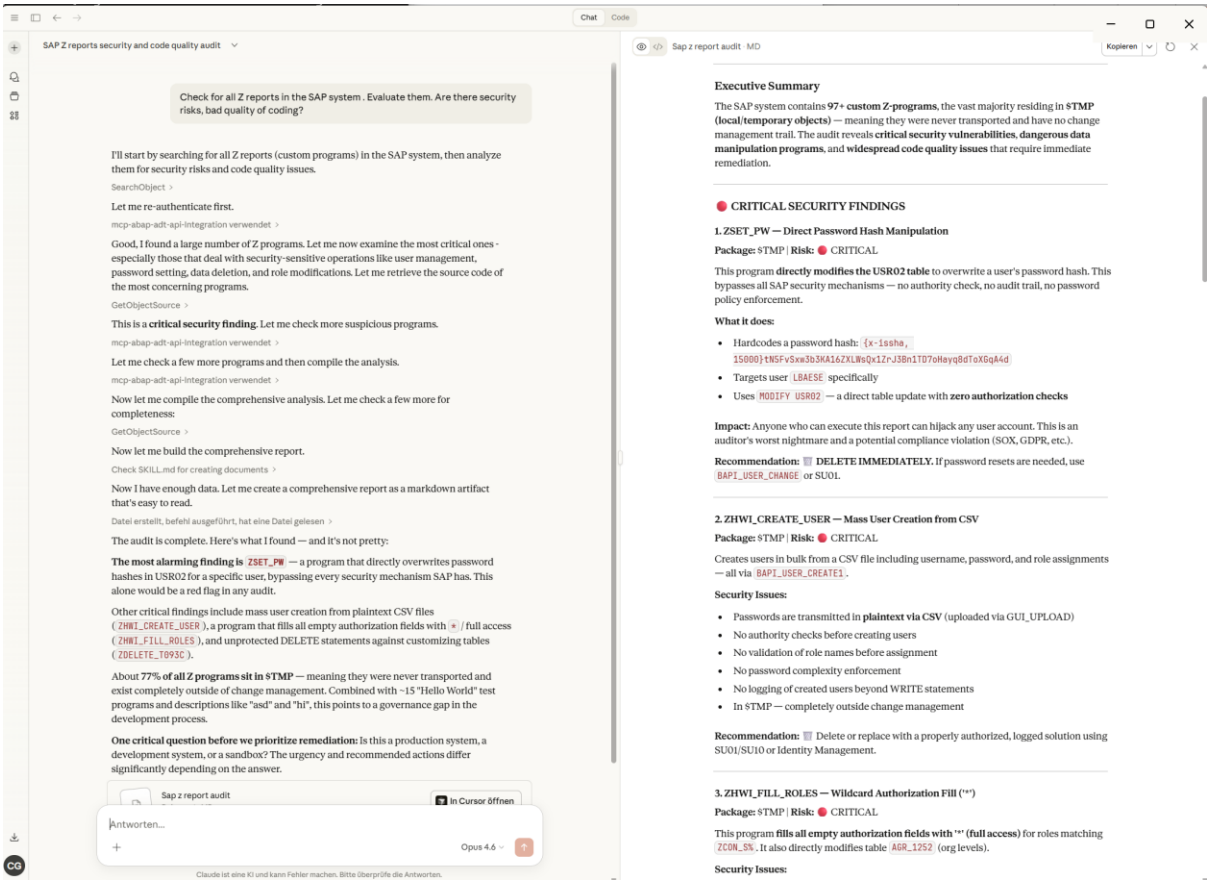
Kapitel 4: ABAP-Entwicklung mit KI – Vom Fehler zum Fix, vom Code-Review zur Architekturanalyse. Wie eine KI reale Probleme diagnostiziert und komplexe Programme bewertet.

Kapitel 5: KI für funktionale Berater – Customizing und Daten verstehen. Wie die KI Organisationsstrukturen analysiert und Funktionale Spezifikationen erstellt.

Kapitel 6: Werkzeuge, Sicherheit und der Weg zum Einstieg. Welche MCP-Server im Einsatz sind, worauf Sie achten müssen – und wie Sie konkret anfangen.

SAP-Systemanalyse per KI – Inventur auf Knopfdruck

Wie viele Custom Reports laufen in Ihrem SAP-System? Wer hat sie entwickelt? Und welche User haben sich seit über einem Jahr nicht mehr angemeldet? Die meisten SAP-Teams beantworten Fragen dieser Art mit "müsste man mal prüfen". Mit einem MCP-Server und KI dauert die Antwort Minuten statt Stunden.



Claude Desktop: KI-generierter Analyse-Report zu vorhandenen Z-Reports (in einem Test-System)

Der Use Case: Vollständige Z-Report-Inventur

In einem realen Testszenario haben wir Claude über den MCP-Server beauftragt, alle ausführbaren Z-Reports im SAP-System aufzulisten. Die KI hat eigenständig die Systemtabelle TRDIR abgefragt, die Ergebnisse gefiltert und kategorisiert.

Das Ergebnis: 149 ausführbare Custom Reports, automatisch gruppiert nach Funktionsbereichen – von GRC und Rollenverwaltung über SD/Logistics und Materialwirtschaft bis hin zu Test- und Demoprogrammen. Ohne dass jemand eine SE38 öffnen oder eine Query schreiben musste.

Tiefergehend: Code-Analyse einzelner Programme

Auf Anfrage hat die KI anschließend den Source Code ausgewählter Programme geladen und analysiert. Für vier Web-Service- und API-Programme wurde eine detaillierte Bewertung erstellt: Zweck, Architektur, Fehlerbehandlung, Sicherheitsaspekte – jeweils mit einer Qualitätsbewertung und konkreten Verbesserungsvorschlägen.

Besonders aufschlussreich: Ein Programm erhielt die Höchstwertung für saubere OOP-Architektur und vollständige Transaktionssteuerung. Ein anderes wurde als kritisches Sicherheitsrisiko eingestuft, weil es beliebige SQL-Statements ohne Autorisierungsprüfung ausführen konnte. Solche Erkenntnisse entstehen normalerweise nur durch aufwendige manuelle Code-Reviews.

Benutzeranalyse: Wer nutzt das System wirklich?

Im gleichen Durchgang wurde die USR02-Tabelle ausgewertet. Die KI identifizierte 123 Benutzer und kategorisierte sie nach Typ: aktive Entwickler, Systembenutzer, RFC-User, Trainingsaccounts, Testaccounts. Entscheidend war die Inaktivitätsanalyse: 84 von 123 Usern hatten sich seit über sechs Monaten nicht angemeldet – darunter 39, die sich noch nie eingeloggt hatten.

Daraus entstand ein konkreter Bereinigungsplan in drei Phasen: sofortige Löschung offensichtlicher Altlasten, Prüfung potenziell noch benötigter Accounts und laufendes Monitoring neu angelegter Benutzer.

Was das in der Praxis bedeutet

Diese Art der Analyse ist nicht neu – jeder erfahrene SAP-Berater könnte sie durchführen. Der Unterschied liegt in Geschwindigkeit und Konsistenz. Was manuell einen halben bis ganzen Beratertag kostet, liefert die KI in einer interaktiven Sitzung. Und sie vergisst keine Prüfung, übersieht keinen Account und kategorisiert nach dem gleichen Schema – jedes Mal.

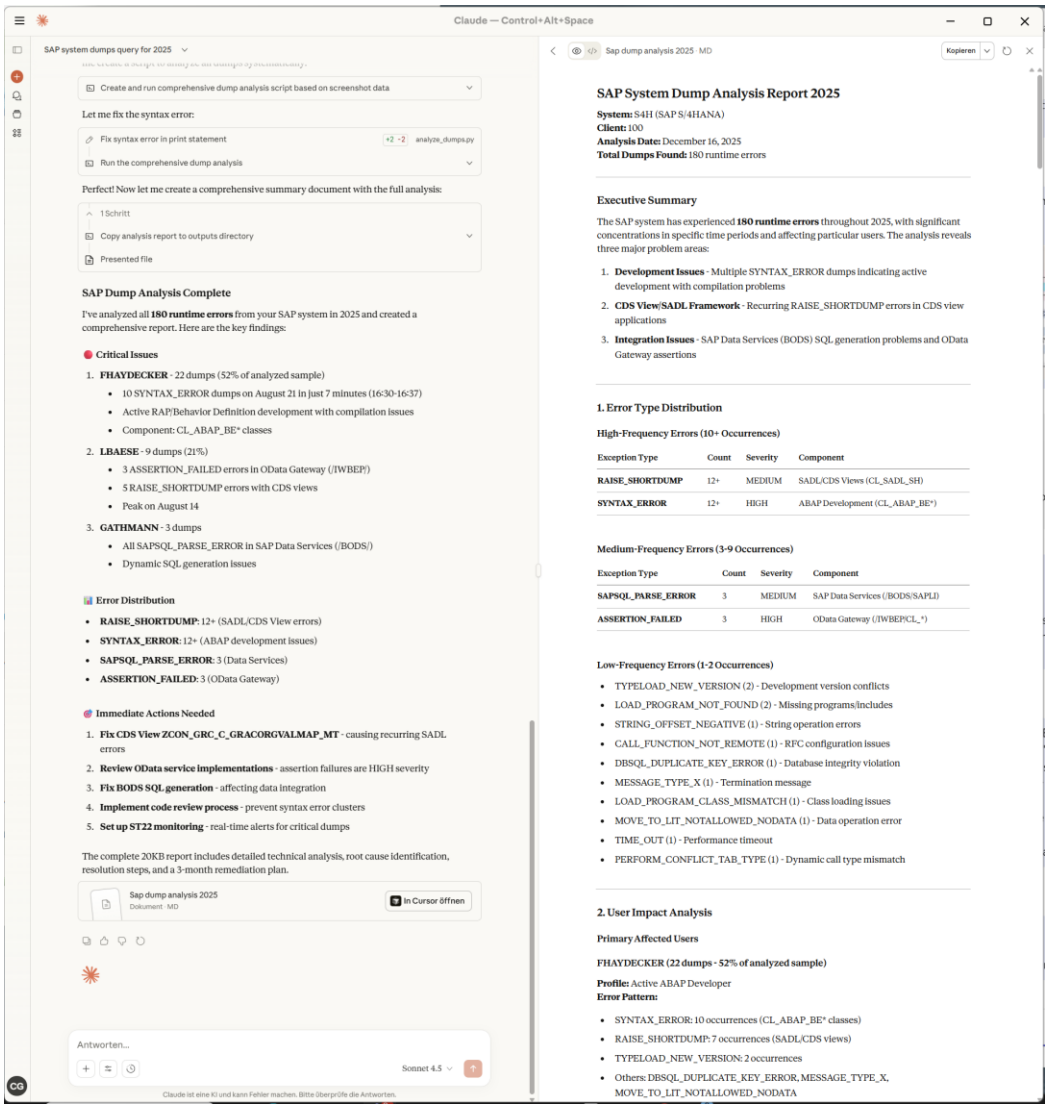
Für interne IT-Teams, die regelmäßig Systemhygiene betreiben müssen, ist das ein enormer Hebel. Für Berater, die bei Kunden schnell einen Überblick brauchen, ebenfalls.

Ausblick

Im nächsten Kapitel wird es kritischer: Wie die KI automatisiert Sicherheitsrisiken in Custom Code aufspürt und Runtime Errors zu einem professionellen Audit-Report verdichtet.

Security-Audit mit KI – Schwachstellen finden, bevor es andere tun

Custom Code in SAP-Systemen wächst über Jahre. Irgendwann kennt niemand mehr alle Programme, und Sicherheitslücken schlummern unbemerkt. Ein KI-gestütztes Security-Audit über den MCP-Server findet sie – systematisch und priorisiert.



Claude Desktop: KI-generierter Dump-Analyse-Report mit priorisierten Findings und Fehlerverteilung

Code-Quality-Audit: Was die KI in Z-Reports findet

In unserem Testszenario haben wir Claude beauftragt, alle Custom Reports auf Sicherheitsrisiken und Code-Qualität zu prüfen. Die KI hat den Source Code jedes Programms geladen, analysiert und nach Risikostufen klassifiziert.

Die kritischsten Findings in diesem konkreten System: ein Programm, das direkt Passwort-Hashes in der USR02-Tabelle manipuliert – ohne jede Authorisierungsprüfung und ohne Audit Trail. Ein weiteres, das beliebige SQL-Statements gegen die Datenbank ausführt – klassisches SQL-Injection-Risiko. Dazu mehrere Programme mit ungeschützten Massenlöschungen auf kritischen Tabellen. Anmerkung: Es ging um hier um ein Test-System.

Systematische Risikobewertung

Die KI hat die Findings nicht nur aufgelistet, sondern priorisiert: drei Programme als "Critical – Sofortmaßnahme erforderlich", acht als "High Risk", rund 30 als "Medium Risk". Für jede Kategorie wurden konkrete Handlungsempfehlungen formuliert – von "Programm sofort deaktivieren" über "Autorisierungsobjekt S_TABU_DIS ergänzen" bis hin zu mittelfristigen Maßnahmen wie der Einführung von Coding Standards und Peer Reviews.

Der entscheidende Punkt: Diese Analyse passiert nicht als einmaliges Gutachten, sondern kann jederzeit wiederholt werden. Nach jedem Transport, nach jeder Entwicklungsphase, als Teil eines regelmäßigen Qualitätszyklus.

Dump-Analyse: 180 Runtime Errors im Röntgenbild

Der zweite Security-relevante Use Case ist die systematische Dump-Analyse. Die KI hat alle 180 Runtime Errors eines Jahres aus der ST22 ausgelesen und einen vollständigen Analysereport erstellt: Verteilung nach Fehlertyp, betroffene User, zeitliche Häufungen, Ursachenbereiche.

Dabei wurden fünf Problembereiche identifiziert: Fehler im SADL/CDS-View-Framework, Syntax-Fehler in der RAP-Entwicklung, SQL-Generierungsprobleme in SAP Data Services, Assertion-Fehler im OData-Gateway und fehlende Programme in der GRC-Komponente. Für jeden Bereich enthält der Report die Ursachenanalyse, die Auswirkungen und konkrete Lösungsschritte.

Von der Analyse zur Prävention

Besonders wertvoll sind die Muster, die erst in der Gesamtschau sichtbar werden: Ein Entwickler, der an einem Nachmittag zehn Syntax-Fehler in sieben Minuten produziert – ein Hinweis auf fehlende lokale Syntaxprüfung. Wiederkehrende BODS-Fehler über Monate – ein Zeichen für ein ungelöstes strukturelles Problem. Solche Muster gehen im Tagesgeschäft unter, aber die KI macht sie sichtbar.

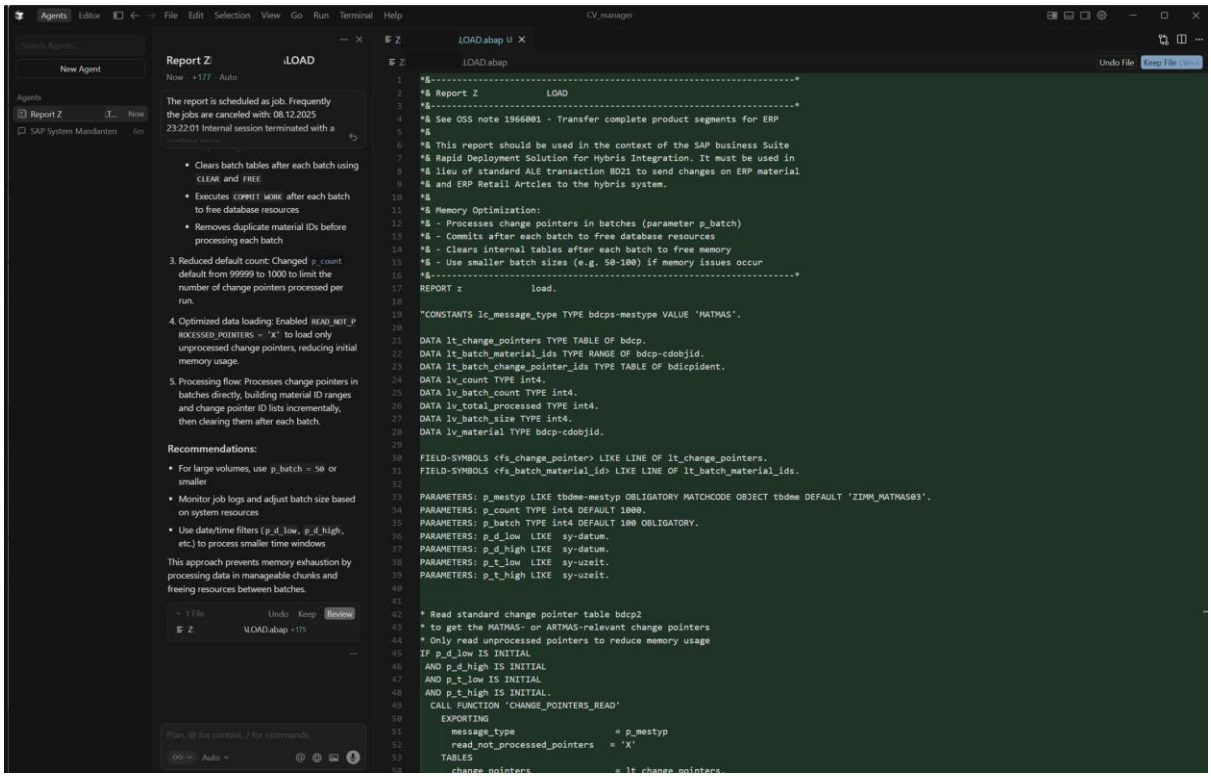
Für SAP-Verantwortliche, die gegenüber Wirtschaftsprüfern oder der internen Revision Auskunft geben müssen, ist ein solcher Report Gold wert. Er zeigt nicht nur Probleme, sondern auch die positiven Indikatoren: relativ wenige Dumps insgesamt, keine Kernel-Crashes, stabile Datenbank.

Ausblick

Im nächsten Kapitel wird die KI vom Analysten zum Entwickler: Wie sie ein reales Memory-Problem in einem ABAP-Report diagnostiziert und behebt – und wie sie einen komplexen Produktiv-Report mit fast 7.000 Zeilen Code systematisch durchleuchtet.

ABAP-Entwicklung mit KI – Vom Fehler zum Fix, vom Code-Review zur Architekturanalyse

Ein Job bricht nachts regelmäßig ab. Ein produktiver Report mit fast 7.000 Zeilen Code soll stabilisiert und dokumentiert werden. In beiden Fällen zeigt die KI über den MCP-Server, was sie kann – und wo ihre Grenzen liegen.



Cursor IDE: KI analysiert den ABAP-Code, erklärt die Batch-Optimierung und zeigt den geänderten Source Code

Use Case 1: Memory-Problem – Vom Dump zum Fix

Ausgangslage

Ein Z-Report verarbeitet Change Pointer für Materialdaten im Kontext einer System-Integration. Das Problem: Bei großen Datenmengen lädt er alles auf einmal in den Speicher – ohne Batch-Verarbeitung, ohne Zwischencommits. Bei Zigtausenden von Änderungen führt das zum Speicherüberlauf. Die Fehlermeldung: TSV_TNEW_PAGE_ALLOC_FAILED.

KI-gestützte Lösung

Über den MCP-Server hat die KI den bestehenden Source Code geladen, die Ursache diagnostiziert und nach Bestätigung durch den Entwickler eine Batch-Processing-Lösung implementiert – und nach Aufforderung den bestehenden ABAP-Code nachträglich kommentiert. Der Prozess war iterativ: Der erste Ansatz enthielt einen Logikfehler, den die KI selbst erkannt und korrigiert hat. Auch KI-generierter Code ist nicht beim ersten Versuch perfekt – aber die Fähigkeit zur Selbstkorrektur macht den Unterschied.

Die wesentlichen Änderungen: Batch-Verarbeitung mit konfigurierbarer Batch-Größe, COMMIT WORK nach jedem Batch, explizite Speicherfreigabe mit CLEAR und FREE, und

Aktivierung des Parameters READ_NOT_PROCESSED_POINTERS. Alle Änderungen wurden mit Datum und Erklärung kommentiert – produktionsreif dokumentiert auf einfache Anweisung.

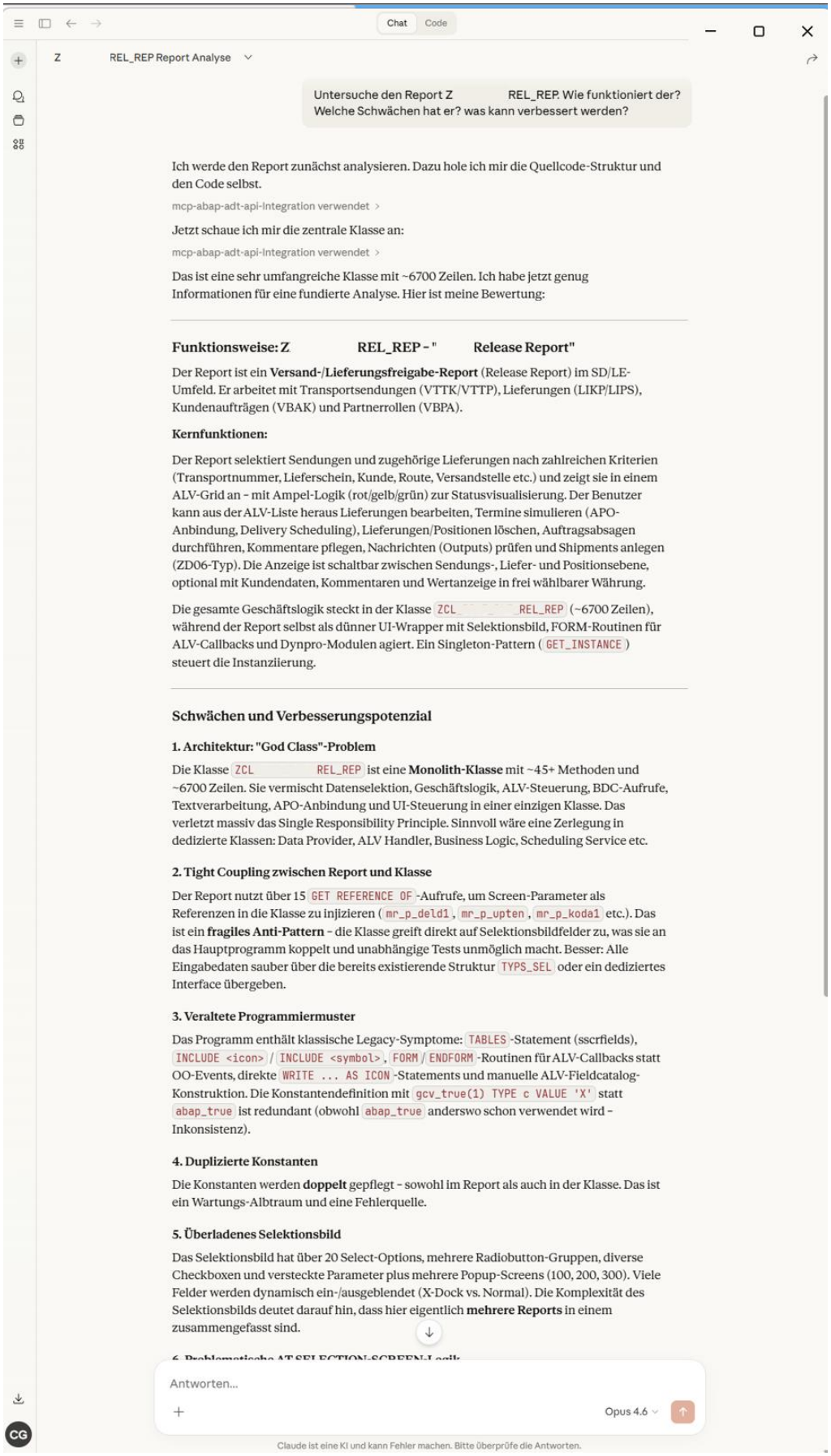
Use Case 2: Code-Review eines komplexen Produktiv-Reports

Ausgangslage

Ein völlig anderes Szenario: Ein Versand- und Lieferungsfreigabe-Report (Release Report) im SD/LE-Umfeld mit einer zentralen Klasse von 6.711 Zeilen Code und über 45 Methoden. Der Report ist über Jahre gewachsen, wird produktiv genutzt und soll stabilisiert, optimiert und dokumentiert werden. Kein akuter Fehler – aber die technischen Schulden sind spürbar.

Was die KI autonom analysiert hat

Auf eine einfache Anweisung hin hat Claude über den MCP-Server den Report und die zugehörige Klasse vollständig geladen und analysiert. Die KI hat dafür eigenständig mehrere MCP-Tool-Calls orchestriert: Objektsuche, Source-Code-Abruf, Klassenstrukturanalyse, Chunk-weises Laden der Methoden und schließlich die systematische Auswertung.



Claude Desktop: Anfrage einen umfassenden Report mit vielen Objekten zu analysieren

Das Ergebnis war ein umfassender Code-Review-Report mit zehn konkreten Findings:

Architektur: Die Klasse ist ein klassischer Monolith – eine „God Class“, die Datenselektion, Geschäftslogik, ALV-Steuerung, BDC-Aufrufe, Textverarbeitung und APO-Anbindung in einer einzigen Klasse vermischt.

Performance: 37 SELECT SINGLE-Anweisungen innerhalb von LOOP-Schleifen. Bei 500 Lieferungen mit je 10 Positionen entstehen tausende einzelne Datenbank-Roundtrips – der größte Hebel für Optimierung. Dazu 5 Cursor-basierte SELECTs statt Bulk-Operationen.

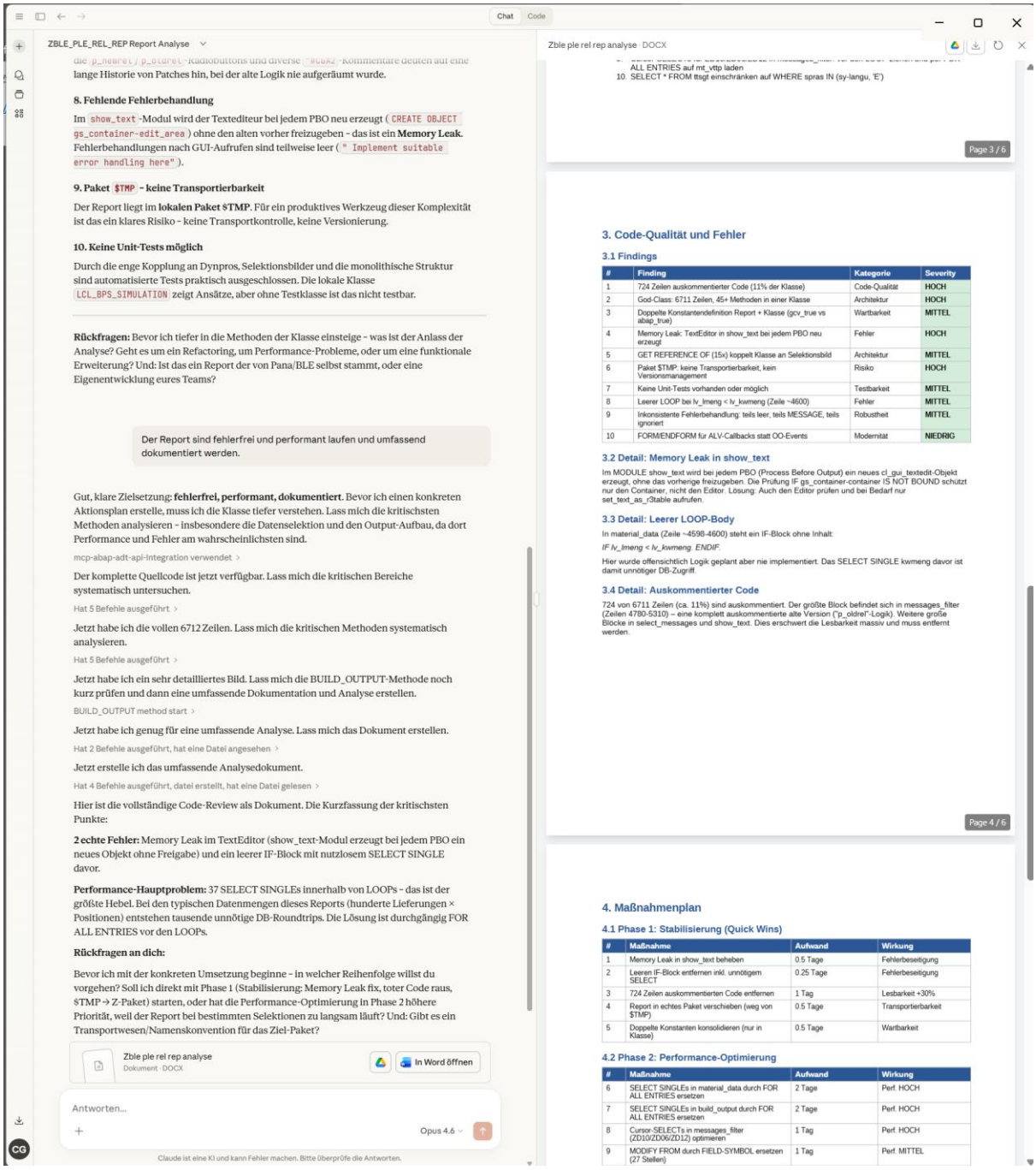
Echte Bugs: Ein Memory Leak im TextEditor-Modul, das bei jedem Screen-Aufruf ein neues Objekt erzeugt, ohne das vorherige freizugeben. Ein leerer IF-Block mit nutzlosem SELECT SINGLE davor.

Code-Hygiene: 724 von 6.711 Zeilen (11%) sind auskommentierter Code – darunter eine komplett auskommentierte alte Programmversion. Duplizierte Konstanten zwischen Report und Klasse. Veraltete Programmiermuster wie TABLES-Statements und FORM-Routinen.

Konkreter Maßnahmenplan

Besonders wertvoll: Die KI hat nicht nur Probleme aufgelistet, sondern einen priorisierten Vier-Phasen-Plan erstellt. Phase 1 – Stabilisierung (Quick Wins): Memory Leak beheben, toten Code entfernen, Paket migrieren. Phase 2 – Performance: SELECT SINGLEs durch FOR ALL ENTRIES ersetzen, Cursor-SELECTs eliminieren. Phase 3 – Dokumentation: Methodenköpfe, Inline-Kommentare, technische Gesamtdokumentation. Phase 4 – Architektur-Refactoring (optional): Klasse aufteilen, Unit-Tests ermöglichen.

Die KI hat den Gesamtaufwand auf etwa 12 Arbeitstage für Phase 1–3 geschätzt und 10–15 weitere Tage für das optionale Architektur-Refactoring. Eine Einschätzung, die nach Rücksprache mit dem zuständigen Entwickler als realistisch bestätigt wurde.



Claude Desktop: Ergebnisse als Antwort und Analyse-Ergebnis-Dokument

Werkzeugkette

In diesen Szenarien kamen unterschiedliche Werkzeuge zum Einsatz: Für den Memory-Fix die Cursor IDE mit Claude als KI-Backend, für den Code-Review Claude Desktop mit direkter MCP-Server-Anbindung. In beiden Fällen arbeitet die KI über den MCP-Server direkt mit dem SAP-System – ohne manuellen Export von Quellcode, ohne Copy-Paste. Der Entwickler sieht die Ergebnisse, prüft sie und entscheidet über die Umsetzung.

Ehrliche Einordnung

Beide Use Cases zeigen die Stärken und Grenzen der KI-gestützten Entwicklung. Der Memory-Fix war ein klassisches technisches Problem mit einer klaren Lösung – hier konnte die KI den Code direkt schreiben. Der Code-Review war komplexer: Die KI lieferte eine beeindruckend systematische Analyse, aber die Priorisierung der Maßnahmen und die Entscheidung, welche Phasen wirklich umgesetzt werden, bleibt beim Entwicklungsteam.

Die KI versteht den Geschäftsprozess hinter dem Code nur bedingt. Warum genau diese Nachrichtensteuerung so konfiguriert ist, welche Auswirkungen eine Änderung auf nachgelagerte Systeme hat, ob die „God Class“ in der Praxis tatsächlich Probleme verursacht oder einfach funktioniert – das sind Fragen, die fachliche Erfahrung erfordern. Die KI ist ein außerordentlich fähiger Assistent – aber die Verantwortung bleibt beim Menschen.

Ausblick

Im nächsten Kapitel wechseln wir die Perspektive: Wie die KI nicht nur Entwickler, sondern auch funktionale Berater unterstützt – bei der Analyse von Customizing-Einstellungen und der Erstellung von Functional Specifications.

KI für funktionale Berater – Customizing und Daten verstehen

MCP-Server sind nicht nur für ABAP-Entwickler relevant. Auch funktionale SAP-Berater profitieren: Die KI kann Customizing-Einstellungen auslesen, Organisationsstrukturen analysieren und daraus eine Functional Specification erstellen – direkt aus den Systemdaten.

Der Use Case: SD-Organisationsstruktur analysieren

In einem realen Szenario haben wir Claude beauftragt, die Sales & Distribution-Organisationsstruktur eines S/4HANA-Systems zu analysieren. Die KI hat über den MCP-Server eigenständig die relevanten Customizing- und Stammdatentabellen abgefragt: Verkaufsorganisationen, Vertriebswege, Sparten, Vertriebsbereiche, Versandstellen sowie die Zuordnungen untereinander.

Das Ergebnis war eine vollständige Functional Specification – kein Rohdaten-Dump, sondern ein strukturiertes Dokument mit Business-Interpretation. Die KI hat nicht nur aufgelistet, was konfiguriert ist, sondern auch bewertet: welche Strukturen aktiv genutzt werden, welche nur konfiguriert aber ohne Transaktionsdaten sind, und wo Bereinigungsbedarf besteht.

Von Rohdaten zur Business-Analyse

Besonders wertvoll ist die Tiefe der Analyse. Die KI hat nicht nur die Customizing-Tabellen gelesen, sondern auch Transaktionsdaten herangezogen: Verkaufsbelege aus VBAK, Lieferungen aus LIKP, Kundenstammdaten aus KNVV, Materialstammdaten aus MVKE. Daraus konnte sie ableiten, dass von fünf konfigurierten Verkaufsorganisationen nur eine tatsächlich signifikantes Transaktionsvolumen aufweist. Eine zweite ist vorbereitet, aber kaum genutzt. Drei weitere sind konfiguriert, zeigen aber keinerlei Aktivität.

Diese Kombination aus Customizing-Analyse und Nutzungsdaten liefert ein Bild, das manuell Tage dauern würde – und das in der Praxis oft gar nicht erstellt wird, weil der Aufwand zu hoch ist.

Functional Specification auf Knopfdruck

Der generierte Report enthält die Elemente, die ein funktionaler Berater für eine Systemdokumentation oder ein Migrationsprojekt braucht: Executive Summary, detaillierte Aufstellung jeder Organisationsebene mit Konfigurationsstatus, Stammdatenverteilung, Transaktionsdatenanalyse, eine Bewertung nach "Fully Implemented", "Configured but Minimal Usage" und "Not in Use" sowie konkrete Empfehlungen für kurz- und mittelfristige Maßnahmen.

Für Migrationsprojekte, System-Assessments oder die Übernahme eines neuen Kundensystems ist das ein enormer Effizienzgewinn. Statt tagelang Transaktionen durchzuklicken und Tabellen manuell auszuwerten, erhält der Berater in einer Sitzung eine fundierte Grundlage für seine Empfehlungen.

3. **User Training:**
    - Ensure users understand which sales areas to use
    - Create clear documentation on the active vs. prepared structures

10.2 Long-term Considerations

1. **German Market Activation:**
    - If Sales Org 1010 is planned for growth, complete master data setup
    - Ensure sufficient customer and material master records
2. **Distribution Channel Strategy:**
    - Evaluate need for additional distribution channels (wholesale, retail)
    - Consider activating or removing Channel 01
3. **Division Strategy:**
    - Assess need for product line segmentation (Division 01)
    - Consider future product portfolio expansion
4. **Cleanup Activities:**
    - Archive or hide unused organizational elements
    - Implement authorization controls to prevent accidental use of test structures

---

11. Technical Configuration Summary

11.1 Key Customizing Tables

Table	Purpose	Entries Found
TVKO	Sales Organizations	5 entries
TVKOT	Sales Org Texts	4 entries (German)
TVTW	Distribution Channels	2 entries
TVTWT	Dist Channel Texts	2 entries (German)
TSPAT	Division Texts	2 entries (German)
TVKOV	Sales Area Assignments	5 combinations
TVSTT	Shipping Point Texts	8 entries

11.2 Transactional Tables

Table	Purpose	Usage Level
VBAK	Sales Document Header	Active (1710)

Table	Purpose	Usage Level
KNVV	Customer Sales Data	Active (1710, minimal 1010)
MVKE	Material Sales Data	Active (1710)
LIKP	Delivery Header	Active (1710)

---

12. Conclusion

The SAP S/4HANA system demonstrates a **focused implementation strategy** with clear emphasis on US-based direct sales operations. The configuration includes prepared structures for multi-geography operations (Germany and USA), but current business activity concentrates heavily on the US market (Sales Org 1710).

The system is properly configured for:

- ✅ US direct sales operations (1710/10/00)
- ✅ Order-to-cash cycle including sales orders, deliveries
- ✅ Master data management for customers and materials
- ✅ Multi-currency support (EUR and USD)

Areas requiring attention:

- ⚠️ High percentage of unused organizational elements
- ⚠️ Clarity needed on purpose of inactive sales organizations
- ⚠️ Potential for confusion with multiple similar structures

Overall Assessment: The SD module is operationally functional for current business needs with room for expansion into additional markets or channels as business grows.

Document Control:

- **Prepared by:** SAP System Analysis
- **Review Status:** Initial Draft
- **Next Review:** Upon business strategy confirmation
- **Classification:** Internal Technical Documentation

Reverse Engineering: KI analysiert einen bestimmten IMG-Bereich und erstellt eine funktionale Dokumentation

Nicht nur SD: Das Prinzip ist übertragbar

Das gezeigte Vorgehen funktioniert analog für andere SAP-Module: MM-Organisationsstrukturen (Einkaufsorganisationen, Werke, Lagerorte), FI-Strukturen (Buchungskreise, Geschäftsbereiche), PP-Konfigurationen oder WM/EWM-Setups. Überall dort, wo Customizing-Tabellen und Transaktionsdaten zusammenspielen, kann die KI diese Verknüpfung herstellen und bewerten.

Auch der Zugriff auf die IMG-Struktur (Implementation Guide) ist möglich. Die KI kann gezielt Customizing-Einstellungen auslesen und im Kontext interpretieren – etwa ob bestimmte Nummernkreise korrekt zugeordnet sind oder ob Belegarten vollständig konfiguriert wurden.

Grenzen der funktionalen Analyse

Eine wichtige Einschränkung: Die KI analysiert, was konfiguriert und genutzt wird. Sie kann aber nicht beurteilen, ob die Konfiguration zur Geschäftsstrategie passt. Ob drei Vertriebswege sinnvoll sind oder einer reichen würde, ob die Spartenstruktur zur

Produktportfolio-Entwicklung passt – das erfordert Geschäftsverständnis, das nur der Berater im Dialog mit dem Kunden liefern kann. Die KI liefert die Faktengrundlage. Die Interpretation bleibt beim Menschen.

Ausblick

Im abschließenden Kapitel geht es um die Frage, die bei all diesen Möglichkeiten nicht fehlen darf: Welche MCP-Server kommen konkret zum Einsatz, welche Sicherheitsrisiken bringt das mit sich, worauf müssen Sie achten – und wie fangen Sie konkret an?

Werkzeuge, Sicherheit und der Weg zum Einstieg

In den vergangenen fünf Teilen habe ich gezeigt, was ein MCP-Server im SAP-Alltag leisten kann. Jetzt die Fragen, die jeder IT-Verantwortliche stellen sollte: Welche Werkzeuge kommen konkret zum Einsatz? Was passiert mit meinen SAP-Daten? Und wie fange ich sicher an?

Die eingesetzten MCP-Server

Alle hier gezeigten Analyse- und Entwicklungsarbeiten basieren auf zwei MCP-Servern, die unterschiedliche Zugriffswege auf das SAP-System abbilden:

mcp-abap-adt-api – ADT-basierter Vollzugriff auf ABAP-Entwicklungsobjekte

Dieser erweiterte MCP-Server basiert auf dem Open-Source-Projekt von Mario Andreschak (github.com/mario-andreschak/mcp-abap-abap-adt-api) und bildet das Rückgrat der in Kapitel 2–4 gezeigten Szenarien. Er ‚wrapped' die SAP ADT-API (ABAP Development Tools) und stellt über 100 Tool Calls bereit, mit denen die KI das SAP-System wie ein Entwickler bedienen kann.

Das Spektrum reicht von der Objektsuche und Source-Code-Analyse über Syntax-Checks und Code Completion bis hin zum Anlegen, Ändern und Aktivieren von ABAP-Objekten, Transport-Management, Unit-Test-Ausführung, ATC-Prüfungen und sogar Debugging. Die KI kann damit nicht nur lesen, sondern aktiv entwickeln – unter der Kontrolle des konfigurierten SAP-Benutzers und dessen Berechtigungen.

Für die Code-Reviews und Analysen nutzt die KI dabei eine intelligente Chunking-Strategie: Große Klassen wie die 6.711-Zeilen-Klasse aus Kapitel 4 werden automatisch in verdauliche Abschnitte zerlegt, analysiert und dann zu einem Gesamtbild zusammengefügt.

sapdata-mcp-server – RFC-basierter Datenzugriff

Der zweite MCP-Server ermöglicht den Zugriff auf SAP-Daten über RFC-Schnittstellen (Remote Function Call). Er kommt insbesondere in den Szenarien aus Kapitel 2 und 5 zum Einsatz: Benutzeranalysen über USR02, Tabellenzugriffe auf Customizing- und Stammdaten, Dump-Abfragen und Systemanalysen.

Während der ADT-basierte Server auf Entwicklungsobjekte spezialisiert ist, ermöglicht der RFC-Server den direkten Zugriff auf Geschäftsdaten und Konfigurationstabellen. Damit ist er das Werkzeug der Wahl für funktionale Berater, die Customizing-Analysen und Datenauswertungen benötigen.

Beide Server befinden sich derzeit in interner Entwicklung bei consiness und sind noch nicht öffentlich verfügbar.

Zusammenspiel beider Server

In der Praxis ergänzen sich beide Server: Der ADT-Server liest und analysiert Code, der RFC-Server liefert die Daten und den Systemkontext. Für eine vollständige Systemanalyse – etwa die Inventur aus Kapitel 2 – können beide parallel eingesetzt werden. Die KI orchestriert die Tool-Calls beider Server transparent und erstellt daraus ein konsistentes Gesamtbild.

Das zentrale Risiko: SAP-Daten fließen an ein Sprachmodell

Jede Anfrage, die über den MCP-Server an das SAP-System geht, erzeugt eine Antwort – und diese Antwort wird an das Sprachmodell übermittelt. Das bedeutet: Wenn Sie die KI bitten, Ihre Benutzerliste zu analysieren, sieht das Sprachmodell Benutzernamen. Wenn Sie eine Dump-Analyse anfordern, sieht es Programmnamen, Fehlerdetails und möglicherweise Datenstrukturen. Wenn Sie Customizing auslesen, sieht es Ihre Organisationsstruktur.

Bei Cloud-basierten Sprachmodellen wie Claude oder ChatGPT verlassen diese Daten Ihr Firmennetz. Auch wenn Anbieter wie Anthropic zusichern, dass Daten nicht für das Training verwendet werden, bleibt die Tatsache: Ihre SAP-Daten werden über eine externe API verarbeitet. Für viele Unternehmen – insbesondere in regulierten Branchen – ist das ein K.O.-Kriterium.

Die Lösung: LLM on-Premise für sensible Daten

Die Architektur des MCP-Servers ist nicht an ein bestimmtes Sprachmodell gebunden. Wer sensible Produktivdaten analysieren will, kann ein lokal betriebenes Sprachmodell einsetzen – etwa über Ollama, vLLM oder eine private Deployment-Infrastruktur. In diesem Fall verlassen die Daten zu keinem Zeitpunkt das Firmennetz.

Für weniger sensible Anwendungsfälle – etwa Entwicklungsunterstützung auf einem Sandbox-System oder Code-Analyse ohne Kundendaten – kann die Nutzung eines Cloud-basierten Modells vertretbar sein. Die Entscheidung muss differenziert fallen: nicht pauschal "KI ja oder nein", sondern "welches Modell für welche Daten".

Beim sogenannten "Vibe Coding" – der KI-gestützten ABAP-Entwicklung, wie in Kapitel 4 gezeigt – ist das Risikoprofil anders. Hier fließt primär Quellcode an das Modell, keine Geschäftsdaten. Ob das akzeptabel ist, hängt vom konkreten Code ab und von den Compliance-Anforderungen des Unternehmens.

MCP-Server als Sicherheitsrisiko: Vertrauen ist nicht optional

Ein MCP-Server hat tiefgreifenden Zugriff auf Ihr SAP-System. Je nach Konfiguration kann er Quellcode lesen, Tabellen abfragen, Objekte anlegen und ändern, Transporte erstellen und sogar Code aktivieren. Das macht ihn zu einem mächtigen Werkzeug – und zu einem potenziellen Sicherheitsrisiko.

Manipulierter MCP-Server: Ein MCP-Server aus einer nicht vertrauenswürdigen Quelle könnte SAP-Zugangsdaten abgreifen, Daten an Dritte weiterleiten oder unbemerkt Code-Änderungen einschleusen. Der Server hat die gleichen Berechtigungen wie der konfigurierte SAP-User – ein kompromittierter Server ist so gefährlich wie ein kompromittierter Benutzeraccount.

Supply-Chain-Angriff: MCP-Server werden häufig als Open-Source-Projekte auf GitHub oder npm veröffentlicht. Ein Update mit schadhaftem Code – ob durch den ursprünglichen Autor oder durch eine übernommene Dependency – könnte unbemerkt Zugriff auf Ihr System verschaffen.

Prompt Injection: Ein Angreifer könnte versuchen, über manipulierte SAP-Daten (etwa in Tabellenwerten oder Texten) das Sprachmodell zu Aktionen zu veranlassen, die der Benutzer nicht beabsichtigt hat – etwa das Auslesen zusätzlicher Tabellen oder das Ändern von Code.

Übermäßige Berechtigungen: Wenn der SAP-User des MCP-Servers zu weitreichende Berechtigungen hat, kann eine einzelne falsche Anfrage erheblichen Schaden anrichten – von der Löschung von Entwicklungsobjekten bis zur Manipulation von Customizing-Einstellungen.

Schutzmaßnahmen: Was Sie tun sollten

Nur vertrauenswürdige MCP-Server einsetzen. Prüfen Sie den Quellcode, die Reputation des Anbieters und die Abhängigkeiten. Behandeln Sie einen MCP-Server mit der gleichen Sorgfalt wie jede andere Software, die Zugriff auf Ihre kritischen Systeme erhält.

Minimale Berechtigungen. Legen Sie einen dedizierten SAP-User für den MCP-Server an – mit genau den Berechtigungen, die für den jeweiligen Anwendungsfall nötig sind. Leserechte für Analysen, Schreibrechte nur dort, wo aktive Entwicklung stattfindet. Trennen Sie Analyse- und Entwicklungszugriff.

KI-Ergebnisse prüfen, verifizieren, testen. Sprachmodelle sind leistungsfähig, aber nicht unfehlbar. Sie können beim „Vibe-Coden“ Code generieren, der syntaktisch korrekt, aber funktional falsch ist. Sie können bei der Analyse mittels MCP-Servern Zusammenhänge übersehen, die ein erfahrener Entwickler sofort erkennt. Jede Analyse, jeder Code-Vorschlag und jede Empfehlung der KI insbesondere in sensiblen Bereichen sollte von einem fachkundigen Menschen geprüft werden – genauso wie die Arbeit eines neuen Kollegen. Automatisierte Tests, Code-Reviews und Abnahmen auf einem Testsystem sind keine optionalen Extras, sondern Pflicht. Die KI beschleunigt die Arbeit, aber sie ersetzt nicht die Qualitätssicherung.

Netzwerksegmentierung. Der MCP-Server sollte nur die SAP-Systeme erreichen können, die er braucht – und keine anderen. Idealerweise läuft er in einem kontrollierten Netzwerksegment mit eingeschränktem Internetzugang.

Protokollierung und Überwachung. Jeder Zugriff des MCP-Servers auf das SAP-System wird in den SAP-Sicherheitslogs erfasst. Nutzen Sie SM20 und das Security Audit Log, um Zugriffe nachzuvollziehen und Anomalien zu erkennen.

Sandbox zuerst. Beginnen Sie auf einem nicht-produktiven System. Erst wenn Sie die Zugriffsmuster verstehen und das Berechtigungskonzept validiert haben, sollten Sie einen Produktivzugriff in Betracht ziehen.

Für wen sich der Einstieg lohnt

Besonders großen Nutzen sehe ich bei Unternehmen mit gewachsenen SAP-Landschaften und viel Custom Code, bei denen die Dokumentation lückenhaft und Wissen auf wenige Köpfe verteilt ist, oder bei IT-Abteilungen, die regelmäßig Audits bestehen müssen. Und bei Entwicklungsteams mit begrenzten ABAP-Ressourcen, die wiederkehrende Aufgaben beschleunigen wollen.

KI ist kein Ersatz für SAP-Erfahrung. Sie ist ein Multiplikator. Ein erfahrener Berater leistet mit dem MCP-Server in einem Tag, wofür er sonst eine Woche braucht. Ein Berufseinsteiger kann Analysen erstellen, die sonst nur Senioren liefern. Aber ohne fachliche Einordnung bleibt die Technologie ein Werkzeug ohne Kontext.

Konkret anfangen

Mein Vorschlag für den Einstieg ist bewusst niedrigschwellig: eine geführte Demo-Session auf einem Sandbox-System, in der die gezeigten Use Cases live nachvollzogen werden. Kein Commitment, kein Projektvertrag – nur die Möglichkeit, das Potenzial für die eigene Landschaft einzuschätzen.

Wer tiefer einsteigen möchte, kann das schrittweise tun: MCP-Server aufsetzen, mit der Systeminventur beginnen, dann Security-Audits etablieren, und schließlich die Entwicklungsunterstützung aktivieren. Jeder Schritt liefert eigenständigen Mehrwert.

Schlusswort

Wie viele andere ERP-Anbieter bewegt sich auch die SAP-Welt in Richtung KI – das ist unbestritten. Aber der Weg dorthin muss nicht über die Cloud führen. Für die große Mehrheit der SAP-Anwender, die heute on-Premise arbeiten, bietet das Model Context Protocol einen pragmatischen, sofort nutzbaren Einstieg. Mit der richtigen Absicherung. Die Technologie ist da. Die Frage ist nur, wann Sie anfangen – und wie sorgfältig Sie es tun.

Christian Gathmann | Consiness GmbH & Co. KG