



Referenzarchitektur

Die Referenzarchitektur für eine agentische Plattform basiert auf der klaren Trennung von Kontext, Interaktion und Ausführung. Ziel ist es, Wissen strukturiert bereitzustellen, koordinierte Aktionen zwischen Agenten zu ermöglichen und gleichzeitig Governance, Skalierbarkeit und Nachvollziehbarkeit sicherzustellen. Im Kern besteht die Architektur aus drei logischen Schichten.

- Kontextschicht (MCP Layer): Diese Schicht bildet das gemeinsame Gedächtnis des Systems. Sie stellt Ressourcen, Tools und strukturierte Kontextobjekte bereit. Daten stammen aus internen Systemen, Datenbanken, Dokumenten, Sensoren oder externen APIs. MCP abstrahiert diese Quellen und stellt sie in konsistenter Form bereit. Die Kontextschicht ist persistent, versionierbar und durch Berechtigungsmodelle geschützt.
- Interaktionsschicht (A2A Layer): Diese Schicht koordiniert die Kommunikation zwischen Agenten. Nachrichten transportieren Intentionen, Aufgaben, Status und Ergebnisse. Die Interaktionsschicht kann hierarchisch organisiert sein – etwa über einen Orchestrator-Agenten – oder dezentral im Sinne eines Swarm-Modells. Sie ist zuständig für Delegation, Routing, Fehlerbehandlung und Interaktionssteuerung.
- Ausführungsschicht (Execution Layer): Hier befinden sich die Agenten selbst sowie die ausführbaren Tools und angeschlossenen Fachsysteme. Agenten laden Kontext über MCP, interpretieren diesen, treffen Entscheidungen und interagieren über A2A mit anderen Agenten. Die Ausführungsschicht umfasst auch externe

Zielsysteme wie ERP, CRM, ITSM, IoT-Plattformen oder Wissenssysteme.

Ein typischer Ablauf in dieser Referenzarchitektur folgt einem klaren Sequenzmuster: Ein Agent lädt zunächst relevanten Kontext aus der MCP-Schicht. Anschließend bewertet er die Situation, führt ein internes Reasoning durch und sendet über A2A eine Nachricht an einen weiteren Agenten. Dieser übernimmt eine Teilaufgabe, interagiert gegebenenfalls mit Tools oder Fachsystemen und schreibt das Ergebnis anschließend wieder in den MCP-Kontext zurück. Dadurch entsteht ein kontrollierter Kreislauf aus Kontextnutzung, Interaktion und Kontextanreicherung.

Für produktive Plattformen sind mehrere Querschnittsfunktionen erforderlich: Das dreischichtige Modell aus Context, Interaction und Execution ist das zentrale Strukturprinzip agentischer Architekturen. Es dient dazu, Verantwortlichkeiten klar zu trennen und gleichzeitig ein kontrolliertes Zusammenspiel von Wissen, Kommunikation und Handlung zu ermöglichen.

1. Context Layer (MCP – Wissen und Zustand)

Die Kontextschicht ist die Grundlage des gesamten Systems. Sie stellt sicher, dass alle relevanten Informationen strukturiert, konsistent und zugänglich sind. Hier werden Daten nicht nur gespeichert, sondern semantisch angereichert und in einen gemeinsamen Wissensraum überführt.

Der Context Layer übernimmt folgende Aufgaben:

- Aggregation von Daten aus unterschiedlichen Quellen
- Strukturierung und Semantisierung von Informationen
- Bereitstellung von Ressourcen, Tools und Metadaten
- Persistenz und Versionierung von Zustand
- Durchsetzung von Zugriffskontrollen und Policies

Wichtig ist: Der Context Layer ist stateful. Er bildet das „Gedächtnis“ des Systems und ermöglicht es, dass Entscheidungen über mehrere Interaktionen hinweg konsistent bleiben.

2. Interaction Layer (A2A – Kommunikation und Steuerung)

Die Interaktionsschicht definiert, wie Agenten miteinander kommunizieren und Aufgaben koordinieren. Sie transportiert Intentionen, delegiert Aufgaben und synchronisiert Ergebnisse.

Typische Aufgaben dieser Schicht sind:

- Routing und Delegation von Aufgaben zwischen Agenten
- Definition von Kommunikationsmustern (Request/Response, Event-basiert)
- Orchestrierung oder Choreografie von Interaktionen
- Fehlerbehandlung und Retry-Logiken
- Kontrolle von Interaktionsketten (z. B. TTL, Max-Hops)

Der Interaction Layer ist primär stateless. Nachrichten enthalten keine vollständigen Zustände, sondern referenzieren Kontext im MCP. Dadurch bleibt die Kommunikation leichtgewichtig und skalierbar.

3. Execution Layer (Agenten – Entscheidung und Aktion)

Die Ausführungsschicht ist der Ort, an dem Entscheidungen getroffen und Aktionen ausgeführt werden. Hier befinden sich die Agenten als aktive Komponenten des Systems.

Aufgaben der Execution Layer:

- Laden und Interpretieren von Kontext aus MCP
- Durchführung von Reasoning und Entscheidungslogik
- Nutzung von Tools und externen Systemen
- Kommunikation mit anderen Agenten über A2A
- Rückführung von Ergebnissen in den Kontext

Agenten sind die „handelnden Einheiten“. Sie verbinden Kontext (Wissen) mit Interaktion (Koordination) und setzen diese in konkrete Aktionen um.

Zusammenspiel der Schichten

Das Schichtenmodell folgt einem klaren Prinzip:

- Context = Wissen (Was ist bekannt?)
- Interaction = Steuerung (Was soll passieren?)
- Execution = Handlung (Was wird getan?)

Ein typischer Ablauf durchläuft alle drei Schichten:

- Agent lädt Kontext aus dem Context Layer
- Agent interpretiert die Situation (Execution Layer)
- Agent delegiert Aufgaben über Interaction Layer
- Ergebnisse werden wieder im Context Layer gespeichert

Dieses Modell sorgt für:

- klare Entkopplung von Daten, Logik und Kommunikation
- hohe Skalierbarkeit durch stateless Interaktion
- bessere Nachvollziehbarkeit durch zentralen Kontext
- flexible Erweiterbarkeit durch neue Agenten oder Datenquellen

Das Schichtenmodell ist damit nicht nur ein technisches Konzept, sondern ein fundamentales Designprinzip für agentische Systeme. Es schafft die notwendige Struktur, um komplexe, dynamische und verteilte Architekturen beherrschbar zu machen. Die Kernaussage dieser Referenzarchitektur lautet: MCP organisiert Wissen, A2A organisiert Handlung. Erst die Kombination beider Schichten schafft eine belastbare Orchestrierungsplattform für agentische Systeme.

Mehr zum Thema MCP & A2A

Der vollständige Leitfaden „MCP & A2A als Orchestrierungsinfrastruktur“

 [Jetzt bei Amazon bestellen](#)