

Entscheidungsmatrix für die Wahl der Softwarearchitektur (für KI-Agenten/Entscheidungsträger)

Die Matrix dient dazu, die Anforderungen der identifizierten Domänen (oder Teil-Domänen/Bounded Contexts) mit dem Architekturmuster abzugleichen, das die primären Qualitätsziele dieser Domäne am besten erfüllt.

Domänenanforderung / Primäres Ziel (abgeleitet aus DDD-Domäne)	Domänenmerkmale & Komplexität	Geeignetes Architekturmuster	Begründung (Stärken der Architektur)
Hohe Skalierbarkeit & Elastizität	Komplexe Projekte, große Unternehmenssysteme, Plattformen für Millionen von Benutzern oder unvorhersehbare Verkehrsspitzen (z. B. E-Commerce).	Microservices-Architektur	Zeichnet sich durch Skalierbarkeit aus, da sie die unabhängige Skalierung einzelner Komponenten ermöglicht. Bietet Flexibilität zur Aufrechterhaltung der Leistung bei unvorhersehbaren Spitzen.
Echtzeit-Verarbeitung & Reaktionsfähigkeit	Systeme, die darauf ausgelegt sind, auf Ereignisse in Echtzeit zu reagieren um diese asynchron zu verarbeiten.	Ereignisgesteuerte Architektur (Event-Driven Architecture, EDA)	Unterstützt lose gekoppelte Komponenten, was hohe Flexibilität und Reaktionsfähigkeit ermöglicht. Gewährleistet hohe Reaktionsfähigkeit und Skalierbarkeit für Echtzeit-Datenverarbeitung.
Hohe Flexibilität & Modularität	Systeme mit individuellen Workflow-Anforderungen, die häufige Aktualisierungen oder den Austausch einzelner Komponenten erfordern (z. B. Workflow-Automatisierung, Finanzdienstleistungen).	Mikrokern-Architektur (auch Plug-in-Architektur genannt)	Ermöglicht Flexibilität und Modularität. Erleichtert die Aktualisierung oder den Austausch einzelner Komponenten, ohne das gesamte System zu stören .
Wartbarkeit und Testbarkeit der Geschäftslogik	Core Domain oder Domänen, in denen die Geschäftslogik (Implementierung) von Infrastrukturaspekten (Datenbank, UI, externe	Hexagonale Architektur (Ports & Adapters)	Bietet Flexibilität und Skalierbarkeit. Organisiert Code so, dass Ressourcen (Datenbank, Protokollkommunikation) leicht geändert werden

Domänenanforderung / Primäres Ziel (abgeleitet aus DDD- Domäne)	Domänenmerkmale & Komplexität	Geeignetes Architekturmuster	Begründung (Stärken der Architektur)
	Dienste) strikt entkoppelt werden muss.		können und die Kernlogik einfach zu testen ist.
Einfache Entwicklung & Wartung, Stabile Basis	Kleine Anwendungen mit einfachen Arbeitsabläufen oder Anwendungen mit stabilen Benutzerbasen (z. B. interne Tools, Standard- Webanwendungen).	Mehrschichtiges Architekturmuster (Layered) oder Monolith	Bietet Einfachheit. Ist oft kostengünstiger in der Entwicklung und Wartung. Bei geringer Skalierbarkeitsanforderung geeignet.
Maximale Leistung bei ressourcenintensiven Prozessen	Hochgradig verteilte Systeme, die massive Datenverkehrsspitzen bewältigen müssen und unabhängige Komponenten erfordern (wie z.B. In- Memory- Datenverarbeitung).	Space-Based Architecture (Weltraumgestützte Architekturen)	Eignet sich gut für Systeme, die massive Datenverkehrsspitzen bewältigen müssen und eine unabhängige Skalierung erfordern.