Agentische KI-Frameworks, -Plattformen, Protokolle und Tools auf AWS

AWS Prescriptive Guidance

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Prescriptive Guidance: Agentische KI-Frameworks, -Plattformen, Protokolle und Tools auf AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und die Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Einführung 1
Zielgruppe 2
Ziele 2
Über diese Inhaltsserie 2
Frameworks 3
Strands Agents 4
Hauptmerkmale von Strands Agents 4
Wann sollte es verwendet werden Strands Agents 5
Implementierungsansatz für Strands Agents 6
Real-world Beispiel für Strands Agents 6
LangChain und LangGraph 6
Hauptmerkmale von und LangChain LangGraph 7
Wann sollte man verwenden LangChain und LangGraph 8
Implementierungsansatz für und LangChain LangGraph 8
Real-world Beispiel für und LangChain LangGraph 8
CrewAI 9
Hauptmerkmale von CrewAI 9
Wann sollte es verwendet werden CrewAI 10
Implementierungsansatz für CrewAI 10
Real-world Beispiel für CrewAI 11
AutoGen 11
Hauptmerkmale von AutoGen 12
Wann sollte es verwendet werden AutoGen 12
Implementierungsansatz für AutoGen 13
Real-world Beispiel für AutoGen 13
LlamaIndex 14
Hauptmerkmale von LlamaIndex 14
Wann sollte es verwendet werden LlamaIndex 15
Implementierungsansatz für LlamaIndex 16
Real-world Beispiel für LlamaIndex 16
Vergleich agentischer KI-Frameworks 17
Überlegungen bei der Auswahl eines agentischen KI-Frameworks 18
Plattformen 20
Warum Plattformen wichtig sind 20

Arten von agentischen KI-Plattformen 21
Überlegungen zur Plattformauswahl 21
Agenten für Amazon Bedrock 22
Hauptfunktionen von Amazon Bedrock Agents 22
Wann sollten Sie Amazon Bedrock Agents verwenden 23
Implementierungsansatz für Amazon Bedrock Agents 23
Ein Beispiel aus der Praxis für Amazon Bedrock Agents 24
Amazon Grundgestein AgentCore 24
Die wichtigsten Funktionen von AgentCore 25
Wann sollte es verwendet werden AgentCore 26
Implementierungsansatz für AgentCore 27
Ein Beispiel aus der Praxis für AgentCore 28
Protokolle 29
Warum die Protokollauswahl wichtig ist 29
Vorteile offener Protokolle 30
Agent-to-agent Protokolle 30
Entscheidung zwischen Protokolloptionen 31
Auswahl von Agentenprotokollen 32
Überlegungen zur Auswahl von behördlichen Protokollen 32
Implementierungsstrategie für behördliche Protokolle 33
Erste Schritte mit MCP 34
Erste Schritte mit A2A 35
Tools 37
Kategorien von Tools 37
Auf Protokollen basierende Tools 37
Framework-native Tools 38
Meta-Tools 38
Auf Protokollen basierende Tools 38
Sicherheitsfunktionen von MCP-Tools 39
Erste Schritte mit MCP-Tools 40
Erkunden Sie Gateway AgentCore 40
Framework-native Tools 40
Meta-Tools 41
Workflow-Metatools 41
Metatools für Agentengraphen 42
Speicher-Metatools 42

Strategie zur Integration von Tools 42
Bewährte Sicherheitsverfahren für die Integration von Tools 43
Authentifizierung und Autorisierung 43
Datenschutz 44
Überwachung und Prüfung 44
Schlussfolgerung 45
Ressourcen 46
AWS Blogs 46
AWS Präskriptive Leitlinien 46
AWS Ressourcen 47
Sonstige Ressourcen 47
Dokumentverlauf 48
..... xlix

Agentische KI-Frameworks, -Plattformen, Protokolle und Tools auf AWS

Aaron Sempf, Ansley Verzosa und Joshua Samuel, Amazon Web Services ()AWS

Januar 2026 (Geschichte der Dokumente)

Agentic AI ist ein leistungsstarkes Paradigma an der Schnittstelle von KI, verteilten Systemen und Softwareentwicklung. Es handelt sich um eine Klasse intelligenter Systeme, die aus autonomen, asynchronen Softwareagenten bestehen, die KI-Modelle verwenden und sich in Tools und Ressourcen integrieren lassen. Die Agenten zeigen Entscheidungsfreiheit, können den Kontext wahrnehmen, über Ziele nachdenken, Entscheidungen treffen und zielgerichtete Maßnahmen im Namen von Benutzern oder Systemen ergreifen. Diese Agenten arbeiten unabhängig, oft kollaborativ, in verteilten Umgebungen und sind darauf ausgelegt, delegierte Ziele mit integrierter Intelligenz, Gedächtnis und Absicht zu verfolgen.

Auf diese Weise können Unternehmen die KI von Agenten nutzen AWS, um komplexe Arbeitsabläufe zu automatisieren, Entscheidungsprozesse zu verbessern und reaktionsfähigere Systeme zu entwickeln. Dieser Leitfaden enthält Informationen zu den wichtigsten Komponenten, die für die Entwicklung effektiver KI-Lösungen für Agenturen erforderlich sind:

- Frameworks stellt aktuelle KI-Frameworks für Agenturen vor, einschließlich einer Übersicht über ihre Vorteile und Anwendungsfälle. Erfahren Sie, wie diese Frameworks undifferenzierte Schwerstarbeit bei Mustern, Protokollen und Tools reduzieren. Machen Sie sich mit den wichtigsten Auswahlkriterien vertraut, um das richtige Framework für Ihre Anforderungen auszuwählen.
- Platforms bietet einen Überblick über die wichtigsten KI-Plattformen (Managed Agent, Open-Source-Orchestrierung und Hybrid) sowie Überlegungen zur Auswahl oder zum Design.
- Protocols untersucht wichtige standardisierte Kommunikationsprotokolle für Agenteninteraktionen. Agent-to-agentProtokolle wie das Open-Source-Model Context Protocol (MCP) und Agent2Agent (A2A) sowie andere proprietäre Implementierungen sind im Entstehen begriffen. Erfahren Sie, wie gängige Protokolle die nahtlose Interaktion verschiedener Protokolle ermöglichen.
- Tools bietet Informationen über protokollbasierte Tools (wie das MCP), Framework-native Tools und Meta-Tools. Organizations können ein Toolkit erstellen, das sich in die wichtigsten Systeme ihrer Workflows integrieren lässt und sowohl Endbenutzer- als auch serverbasierte Agentenworkflows ermöglicht.

Zielgruppe

Dieser Leitfaden richtet sich an Architekten, Entwickler und Technologieführer, die das Potenzial KI-gestützter Softwareagenten in modernen Cloud-nativen Anwendungen nutzen möchten.

Ziele

Dieser Leitfaden hilft Ihnen bei folgenden Aufgaben:

- Vergleichen Sie verschiedene agentische KI-Frameworks, um das für Ihren Anwendungsfall am besten geeignete auszuwählen.
- Erfahren Sie mehr über agentische KI-Plattformen, die Funktionen bieten, um einzelne Agenten in koordinierte, anpassungsfähige Systeme umzuwandeln.
- Erfahren Sie mehr über die Vorteile offener Protokolle für den Aufbau nachhaltiger agentischer KI-Architekturen.
- Entwickeln Sie beim Aufbau von Agentensystemen eine geeignete Strategie zur Integration von Tools.

Über diese Inhaltsserie

Dieser Leitfaden ist Teil einer Reihe über agentic AI on. AWS Weitere Informationen und die anderen Leitfäden dieser Reihe finden Sie unter Agentic AI auf der Prescriptive Guidance-Website. AWS

Frameworks

Foundations of Agentic AI on AWS untersucht die Kernmuster und Arbeitsabläufe, die autonomes, zielgerichtetes Verhalten ermöglichen. Im Mittelpunkt der Implementierung dieser Muster steht die Wahl des Frameworks. Ein Framework ist die Softwarebasis aus vordefiniertem Code, die eine strukturierte Umgebung und gemeinsame Funktionen für die Erstellung und Verwaltung von Tools und Orchestrierungsfunktionen bietet, die für die Entwicklung produktionsreifer autonomer KI-Agenten erforderlich sind.

Effektive agentische KI-Frameworks bieten mehrere grundlegende Funktionen, die unbearbeitete Interaktionen mit einem Large Language Model (LLM) in koordinierte, intelligente Systeme umwandeln, die in der Lage sind, zu argumentieren, zusammenzuarbeiten und zu handeln:

- Die Agentenorchestrierung koordiniert den Informationsfluss und die Entscheidungsfindung zwischen einzelnen oder mehreren Agenten, um komplexe Ziele ohne menschliches Eingreifen zu erreichen.
- Die Toolintegration ermöglicht es Agenten, mit externen Systemen, APIs und Datenquellen zu interagieren, um ihre Fähigkeiten über die Sprachverarbeitung hinaus zu erweitern. Weitere Informationen finden Sie in der Strands Agents Dokumentation unter Tool-Übersicht.
- Die Speicherverwaltung bietet einen dauerhaften oder sitzungsbasierten Status, um den Kontext zwischen Interaktionen aufrechtzuerhalten, was für lang andauernde oder adaptive Aufgaben unerlässlich ist. Fortgeschrittenere Frameworks verfügen über ein Langzeitgedächtnis zum Speichern von Zusammenfassungen und Benutzereinstellungen und ermöglichen so personalisierte und kontextsensitive Agentenerlebnisse. Weitere Informationen finden Sie im Blog unter How to think about Agent Frameworks. LangChain
- Die Workflow-Definition unterstützt strukturierte Muster wie Ketten, Routing, Parallelisierung und Reflexionsschleifen, die ein ausgeklügeltes autonomes Denken ermöglichen.
- Einsatz und Überwachung erleichtern den Übergang von der Entwicklung zur Produktion mit Beobachtbarkeit für autonome Systeme. Weitere Informationen finden Sie in der Ankündigung zur AgentCore allgemeinen Verfügbarkeit von Amazon Bedrock.

Diese Funktionen werden in der gesamten Framework-Landschaft mit unterschiedlichen Ansätzen und Schwerpunkten implementiert und bieten jeweils unterschiedliche Vorteile für unterschiedliche Anwendungsfälle autonomer Agenten und Unternehmenskontexte.

In diesem Abschnitt werden die führenden Frameworks für die Entwicklung agentischer KI-Lösungen vorgestellt und verglichen, wobei der Schwerpunkt auf ihren Stärken, Grenzen und idealen Anwendungsfällen für den autonomen Betrieb liegt:

- Strands & Agenten
- LangChain und LangGraph
- Crew I
- AutoGen
- ???
- Vergleich agentischer KI-Frameworks

> **Note**
>
> In diesem Abschnitt werden die Frameworks behandelt, die speziell die Steuerung der KI unterstützen. Frontend-Schnittstellen oder generative KI ohne Agentur werden nicht behandelt.

Strands Agents

Strands Agentsist ein Open-Source-SDK, das ursprünglich von veröffentlicht wurde AWS, wie im AWS Open Source-Blog beschrieben. Strands Agentswurde für die Entwicklung autonomer KI-Agenten mit einem Model-First-Ansatz entwickelt. Es bietet ein flexibles, erweiterbares Framework, das so konzipiert ist, dass es nahtlos funktioniert und AWS-Services gleichzeitig offen für die Integration mit Komponenten von Drittanbietern bleibt. Strands Agents ist ideal für den Aufbau vollständig autonomer Lösungen.

Hauptmerkmale von Strands Agents

Strands Agentsumfasst die folgenden Hauptfunktionen:

- Model-first Design — Es basiert auf dem Konzept, dass das Basismodell den Kern der Agentenintelligenz bildet und ausgeklügeltes autonomes Denken ermöglicht. Weitere Informationen finden Sie in der Strands Agents Dokumentation unter Agent Loop.
- Multi-agent Kooperationsmuster — Built-in Koordinationsmodelle wie Swarm-, Graph- und Workflow-Muster, die eine skalierbare Zusammenarbeit und Steuerung über verteilte

Agentennetzwerke hinweg ermöglichen. Weitere Informationen finden Sie unter Multi-agent Muster in der Dokumentation zu Strands Agents.

- MCP-Integration — Systemeigene Unterstützung für das Model Context Protocol (MCP), wodurch eine standardisierte Kontextbereitstellung für LLMs für einen konsistenten autonomen Betrieb ermöglicht wird.
- AWS-Service Integration — Nahtlose Verbindung zu Amazon Bedrock, AWS Lambda AWS Step Functions, und anderen AWS-Services für umfassende autonome Workflows. Weitere Informationen finden Sie unter AWS Weekly Roundup (Blog)AWS .
- Auswahl des Foundation-Modells — Unterstützt verschiedene Foundation-Modelle, darunter Anthropic Claude, Amazon Nova (Premier, Pro, Lite und Micro) auf Amazon Bedrock und andere, um für verschiedene Funktionen des autonomen Denkens zu optimieren. Weitere Informationen finden Sie in der Strands Agents Dokumentation unter Amazon Bedrock.
- LLM-API-Integration — Flexible Integration mit verschiedenen LLM-Serviceschnittstellen wie Amazon Bedrock, OpenAI und anderen für den Produktionseinsatz. Weitere Informationen finden Sie in der Strands Agents Dokumentation unter Amazon Bedrock Basic Usage.
- Multimodale Funktionen — Support mehrerer Modalitäten, einschließlich Text-, Sprach- und Bildverarbeitung für umfassende autonome Agenteninteraktionen. Weitere Informationen finden Sie in der Dokumentation unter Amazon Bedrock Multimodal Support. Strands Agents
- Tool-Ökosystem — Umfangreiches Angebot an Tools für die AWS-Service Interaktion mit Erweiterbarkeit für benutzerdefinierte Tools, die die autonomen Funktionen erweitern. Weitere Informationen finden Sie in der Strands Agents Dokumentation unter Tool-Übersicht.

Wann sollte es verwendet werden Strands Agents

Strands Agentseignet sich besonders gut für Szenarien mit autonomen Agenten, darunter:

- Organizations, die auf einer AWS Infrastruktur aufbauen und eine native Integration AWS-Services für autonome Workflows wünschen
- Teams, die Sicherheits-, Skalierbarkeits- und Compliance-Funktionen auf Unternehmensebene für autonome Produktionssysteme benötigen
- Projekte, die Flexibilität bei der Modellauswahl verschiedener Anbieter für spezielle autonome Aufgaben benötigen
- Anwendungsfälle, die eine enge Integration mit bestehenden AWS Workflows und Ressourcen für durchgängige autonome Prozesse erfordern

Implementierungsansatz für Strands Agents

Strands Agentsbietet einen unkomplizierten Implementierungsansatz für Geschäftsbeteiligte, wie im Schnellstartleitfaden für Python und im Schnellstartleitfaden für beschrieben. Typescript Das Framework ermöglicht Organisationen:

- Wählen Sie Basismodelle wie Amazon Nova (Premier, Pro, Lite oder Micro) auf Amazon Bedrock auf der Grundlage spezifischer Geschäftsanforderungen aus.
- Definieren Sie benutzerdefinierte Tools, die eine Verbindung zu Unternehmenssystemen und Datenquellen herstellen.
- Verarbeiten Sie mehrere Modalitäten, darunter Text, Bilder und Sprache.
- Stellen Sie Agenten bereit, die selbstständig auf Geschäftsanfragen antworten und Aufgaben ausführen können.

Dieser Implementierungsansatz ermöglicht es Geschäftsteams, autonome Agenten schnell zu entwickeln und einzusetzen, ohne über fundiertes technisches Fachwissen in der Entwicklung von KI-Modellen zu verfügen.

Real-world Beispiel für Strands Agents

AWS Transform for.NET nutzt Strands Agents zur Unterstützung seiner Funktionen zur Anwendungsmodernisierung, wie in AWS Transform for .NET beschrieben, dem ersten agentischen KI-Dienst für die Modernisierung von.NET-Anwendungen im großen Maßstab (AWS Blog). Dieser Produktionsservice setzt mehrere spezialisierte autonome Agenten ein. Die Agenten arbeiten zusammen, um ältere .NET-Anwendungen zu analysieren, Modernisierungsstrategien zu planen und Codetransformationen in cloudnative Architekturen ohne menschliches Eingreifen durchzuführen. AWS Transform for .NET demonstriert die Produktionsreife von autonomen Strands Agents Unternehmenssystemen.

LangChain und LangGraph

LangChainist eines der etabliertesten Frameworks im agentischen KI-Ökosystem. LangGrapherweitert seine Funktionen, um komplexe, zustandsbehaftete Agenten-Workflows zu unterstützen, wie im Blog beschrieben. LangChain Zusammen bieten sie eine umfassende Lösung für den Aufbau ausgeklügelter autonomer KI-Agenten mit umfangreichen Orchestrierungsfunktionen für einen unabhängigen Betrieb.

Hauptmerkmale von und LangChain LangGraph

LangChainund LangGraph beinhalten die folgenden Hauptmerkmale:

- Komponenten-Ökosystem — Umfangreiche Bibliothek mit vorgefertigten Komponenten für verschiedene Funktionen autonomer Agenten, die die schnelle Entwicklung spezialisierter Agenten ermöglicht. Weitere Informationen finden Sie in der Dokumentation unter Schnellstart. LangChain
- Auswahl von Foundation-Modellen — Support für verschiedene Foundation-Modelle, darunter Anthropic Claude, Amazon Nova-Modelle (Premier, Pro, Lite und Micro) auf Amazon Bedrock und andere für unterschiedliche Argumentationsfähigkeiten. Weitere Informationen finden Sie in der Dokumentation unter Eingaben und Ausgaben. LangChain
- LLM-API-Integration — Standardisierte Schnittstellen für mehrere Large Language Model (LLM) - Dienstleister, darunter Amazon Bedrock und andereOpenAI, für eine flexible Bereitstellung. Weitere Informationen finden Sie in der Dokumentation unter LLMs. LangChain
- Multimodale Verarbeitung — Built-in Unterstützung für Text-, Bild- und Audioverarbeitung, um umfangreiche multimodale Interaktionen mit autonomen Agenten zu ermöglichen. Weitere Informationen finden Sie in der Dokumentation unter Multimodalität. LangChain
- Graph-based Workflows — LangGraph ermöglicht die Definition komplexer Verhaltensweisen autonomer Agenten als Zustandsmaschinen und unterstützt so eine ausgeklügelte Entscheidungslogik. Weitere Informationen finden Sie in der Ankündigung von LangGraphPlatform GA.
- Speicherabstraktionen — Mehrere Optionen für die Kurz- und Langzeitspeicherverwaltung, was für autonome Agenten, die den Kontext im Laufe der Zeit aufrechterhalten, unerlässlich ist. Weitere Informationen finden Sie in der Dokumentation unter So fügen Sie Chatbots Speicher hinzu. LangChain
- Tool-Integration — Umfangreiches Ökosystem von Tool-Integrationen für verschiedene Dienste und APIs, wodurch die Funktionen autonomer Agenten erweitert werden. Weitere Informationen finden Sie in der LangChain Dokumentation unter Tools.
- LangGraph Plattform — Verwaltete Bereitstellungs- und Überwachungslösung für Produktionsumgebungen, die autonome Agenten mit langer Laufzeit unterstützt. Weitere Informationen finden Sie in der Ankündigung von LangGraphPlatform GA.

Wann sollte man verwenden LangChain und LangGraph

LangChainund LangGraph eignen sich besonders gut für Szenarien mit autonomen Agenten, darunter:

- Komplexe, mehrstufige Argumentationsabläufe, die eine ausgeklügelte Orchestrierung für autonome Entscheidungen erfordern
- Projekte, die Zugang zu einem großen Ökosystem vorgefertigter Komponenten und Integrationen für vielfältige autonome Funktionen benötigen
- Teams mit vorhandener Infrastruktur und Fachwissen Python auf Basis von maschinellem Lernen (ML), die autonome Systeme aufbauen möchten
- Anwendungsfälle, die eine komplexe Statusverwaltung für lang andauernde autonome Agentensitzungen erfordern

Implementierungsansatz für und LangChain LangGraph

LangChainund LangGraph bieten einen strukturierten Implementierungsansatz für Unternehmensbeteiligte, wie in der LangGraphDokumentation detailliert beschrieben. Das Framework ermöglicht Organisationen:

- Definieren Sie anspruchsvolle Workflow-Diagramme, die Geschäftsprozesse darstellen.
- Erstellen Sie mehrstufige Argumentationsmuster mit Entscheidungspunkten und bedingter Logik.
- Integrieren Sie multimodale Verarbeitungsfunktionen für den Umgang mit unterschiedlichen Datentypen.
- Implementieren Sie die Qualitätskontrolle mithilfe integrierter Überprüfungs- und Validierungsmechanismen.

Dieser grafisch gestützte Ansatz ermöglicht es Geschäftsteams, komplexe Entscheidungsprozesse als autonome Workflows zu modellieren. Die Teams haben einen klaren Überblick über jeden Schritt des Argumentationsprozesses und können Entscheidungswege überprüfen.

Real-world Beispiel für und LangChain LangGraph

Vodafonehat autonome Agenten implementiert, die LangChain (undLangGraph) verwenden, um seine Workflows für Datentechnik und Betrieb zu verbessern, wie in der Fallstudie LangChain Enterprise detailliert beschrieben. Sie haben interne KI-Assistenten entwickelt, die selbstständig

Leistungskennzahlen überwachen, Informationen aus Dokumentationssystemen abrufen und umsetzbare Erkenntnisse liefern — alles durch Interaktionen in natürlicher Sprache.

Die Vodafone Implementierung verwendet LangChain modulare Dokumentenlader, Vektorintegration und Unterstützung für mehrere LLMs (, LLaMA 3 undGemini), um diese Pipelines schnell zu OpenAI prototypisieren und zu vergleichen. Anschließend strukturierten sie die Orchestrierung mehrerer Agenten durch den Einsatz LangGraph modularer Unteragenten. Diese Agenten führen Aufgaben zur Erfassung, Verarbeitung, Zusammenfassung und Argumentation durch. LangGraphhaben diese Agenten über APIs in ihre Cloud-Systeme integriert.

CrewAI

CrewAIist ein Open-Source-Framework, das sich speziell auf die autonome Orchestrierung mehrerer Agenten konzentriert und unter verfügbar ist. GitHub Es bietet einen strukturierten Ansatz zur Bildung von Teams spezialisierter autonomer Agenten, die zusammenarbeiten, um komplexe Aufgaben ohne menschliches Eingreifen zu lösen. CrewAIbetont die rollenbasierte Koordination und Aufgabendelegierung.

Hauptmerkmale von CrewAI

CrewAIbietet die folgenden Hauptfunktionen:

- Role-based Agentendesign — Autonome Agenten werden mit spezifischen Rollen, Zielen und Hintergrundinformationen definiert, um spezialisiertes Fachwissen zu ermöglichen. Weitere Informationen finden Sie in der CrewAIDokumentation unter Crefting Effective Agents.
- Aufgabendelegierung — Built-in Mechanismen zur autonomen Zuweisung von Aufgaben an geeignete Agenten auf der Grundlage ihrer Fähigkeiten. Weitere Informationen finden Sie in der CrewAIDokumentation unter Aufgaben.
- Zusammenarbeit mit Agenten — Framework für autonome Kommunikation und Wissensaustausch zwischen Agenten ohne menschliche Vermittlung. Weitere Informationen finden Sie in der CrewAI Dokumentation unter Zusammenarbeit.
- Prozessmanagement — Strukturierte Workflows für sequentielle und parallel autonome Aufgabenausführung. Weitere Informationen finden Sie in der CrewAI Dokumentation unter Prozesse.
- Auswahl des Foundation-Modells — Support für verschiedene Foundation-Modelle, darunter Anthropic Claude, Amazon Nova-Modelle (Premier, Pro, Lite und Micro) auf Amazon Bedrock

und andere zur Optimierung für verschiedene Aufgaben des autonomen Denkens. Weitere Informationen finden Sie in der Dokumentation unter LLMs. CrewAI

- LLM-API-Integration — Flexible Integration mit mehreren LLM-Serviceschnittstellen, einschließlich Amazon BedrockOpenAI, und lokalen Modellbereitstellungen. Weitere Informationen finden Sie in der Dokumentation unter Konfigurationsbeispielen für Anbieter. CrewAI
- Multimodale Unterstützung — Neue Funktionen für den Umgang mit Text, Bildern und anderen Modalitäten für umfassende Interaktionen mit autonomen Agenten. Weitere Informationen finden Sie in der Dokumentation unter Verwenden multimodaler Agenten. CrewAI

Wann sollte es verwendet werden CrewAI

CrewAIeignet sich besonders gut für Szenarien mit autonomen Agenten, darunter:

- Komplexe Probleme, die von spezialisiertem, rollenbasiertem Fachwissen profitieren, das autonom arbeitet
- Projekte, die eine ausdrückliche Zusammenarbeit zwischen mehreren autonomen Agenten erfordern
- Anwendungsfälle, in denen die teambasierte Problemzerlegung die autonome Problemlösung verbessert
- Szenarien, die eine klare Trennung der Anliegen zwischen den verschiedenen Rollen autonomer Agenten erfordern

Implementierungsansatz für CrewAI

CrewAIbietet eine rollenbasierte Implementierung eines Ansatzes für Teams von KI-Agenten für Geschäftsbeteiligte, wie in der CrewAI Dokumentation unter Erste Schritte beschrieben. Das Framework ermöglicht Organisationen:

- Definieren Sie spezialisierte autonome Agenten mit spezifischen Rollen, Zielen und Fachgebieten.
- Weisen Sie Agenten Aufgaben auf der Grundlage ihrer speziellen Fähigkeiten zu.
- Richten Sie klare Abhängigkeiten zwischen Aufgaben ein, um strukturierte Workflows zu erstellen.
- Orchestrieren Sie die Zusammenarbeit zwischen mehreren Agenten, um komplexe Probleme zu lösen.

Dieser rollenbasierte Ansatz spiegelt die menschlichen Teamstrukturen wider und macht es Unternehmensleitern leicht, ihn zu verstehen und umzusetzen. Organizations können autonome Teams mit speziellen Fachgebieten bilden, die zusammenarbeiten, um Geschäftsziele zu erreichen, ähnlich wie menschliche Teams arbeiten. Das autonome Team kann jedoch kontinuierlich ohne menschliches Eingreifen arbeiten.

Real-world Beispiel für CrewAI

AWS hat autonome Multi-Agenten-Systeme mithilfe von CrewAI implementiert, die in Amazon Bedrock integriert sind, wie in der CrewAIveröffentlichten Fallstudie detailliert beschrieben. AWS und CrewAI entwickelte ein sicheres, herstellerneutrales Framework. Die CrewAI Open-Source-Architektur „Flows-and-Crews“ lässt sich nahtlos in die Grundmodelle, Speichersysteme und Compliance-Richtlinien von Amazon Bedrock integrieren.

Zu den wichtigsten Elementen der Implementierung gehören:

- Blueprints und Open Sourcing — AWS und CrewAI veröffentlichte Referenzdesigns, die CrewAI Agenten Amazon Bedrock-Modellen und Observability-Tools zuordnen. Außerdem veröffentlichten sie beispielhafte Systeme wie ein Team für AWS Sicherheitsüberprüfungen mit mehreren Mitarbeitern, Abläufe zur Code-Modernisierung und Backoffice-Automatisierung für Konsumgüter (CPG).
- Integration des Observability-Stacks — Die Lösung integriert die Überwachung in Amazon CloudWatch und ermöglicht so Rückverfolgbarkeit und LangFuse Debugging vom Machbarkeitsnachweis bis zur Produktion. AgentOps
- Nachgewiesene Investitionsrendite (ROI) — Erste Pilotprojekte zeigen wichtige Verbesserungen: 70 Prozent schnellere Ausführung bei einem großen Code-Modernisierungsprojekt und etwa 90 Prozent Verkürzung der Verarbeitungszeit für CPG-Backoffice-Abläufe.

AutoGen

AutoGenist ein Open-Source-Framework, das ursprünglich von veröffentlicht wurde. Microsoft AutoGenkonzentriert sich darauf, autonome KI-Agenten zur Konversation und Zusammenarbeit zu ermöglichen. Es bietet eine flexible Architektur für den Aufbau von Systemen mit mehreren Agenten, wobei der Schwerpunkt auf asynchronen, ereignisgesteuerten Interaktionen zwischen Agenten für komplexe autonome Workflows liegt.

Hauptmerkmale von AutoGen

AutoGenbietet die folgenden Hauptfunktionen:

- Konversationsagenten — Basiert auf Konversationen zwischen autonomen Agenten in natürlicher Sprache und ermöglicht so anspruchsvolles Denken im Dialog. Weitere Informationen finden Sie in der Dokumentation unter Multi-agent Conversation Framework. AutoGen
- Asynchrone Architektur — Event-driven Design für blockierungsfreie autonome Agenteninteraktionen, unterstützt komplexe parallel Workflows. Weitere Informationen finden Sie in der Dokumentation unter Mehrere Aufgaben in einer Folge von asynchronen Chats lösen. AutoGen
- Human-in-the-loop— Starke Unterstützung für die optionale Beteiligung von Mitarbeitern an ansonsten autonomen Agenten-Workflows, falls erforderlich. Weitere Informationen finden Sie in der AutoGen Dokumentation unter Zulassen von menschlichem Feedback bei Agenten.
- Codegenerierung und -ausführung — Spezialisierte Funktionen für codeorientierte autonome Agenten, die Code schreiben und ausführen können. Weitere Informationen finden Sie in der Dokumentation unter Codeausführung. AutoGen
- Individuell anpassbares Verhalten — Flexible autonome Agentenkonfiguration und Konversationssteuerung für verschiedene Anwendungsfälle. Weitere Informationen finden Sie in der Dokumentation unter agentchat.conversable_agent. AutoGen
- Auswahl des Foundation-Modells — Support für verschiedene Foundation-Modelle, darunter Anthropic Claude, Amazon Nova-Modelle (Premier, Pro, Lite und Micro) auf Amazon Bedrock und andere für verschiedene Funktionen für autonomes Denken. Weitere Informationen finden Sie in der Dokumentation unter LLM-Konfiguration. AutoGen
- LLM-API-Integration — Standardisierte Konfiguration für mehrere LLM-Serviceschnittstellen, darunter Amazon Bedrock, undOpenAI. Azure OpenAI Weitere Informationen finden Sie unter oai.openai_utils in der API-Referenz. AutoGen
- Multimodale Verarbeitung — Support für Text- und Bildverarbeitung, um umfangreiche multimodale Interaktionen mit autonomen Agenten zu ermöglichen. Weitere Informationen finden Sie unter Umgang mit multimodalen Modellen: GPT-4V AutoGen in der Dokumentation. AutoGen

Wann sollte es verwendet werden AutoGen

AutoGeneignet sich besonders gut für Szenarien mit autonomen Agenten, darunter:

- Anwendungen, die für komplexes Denken natürliche Konversationsabläufe zwischen autonomen Agenten erfordern
- Projekte, die sowohl einen vollständig autonomen Betrieb als auch optionale Funktionen zur menschlichen Überwachung benötigen
- Anwendungsfälle, die autonome Codegenerierung, Ausführung und Debugging ohne menschliches Eingreifen beinhalten
- Szenarien, die flexible, asynchrone Kommunikationsmuster für autonome Agenten erfordern

Implementierungsansatz für AutoGen

AutoGenbietet einen dialogorientierten Implementierungsansatz für Geschäftsbeteiligte, wie in der AutoGen Dokumentation unter Erste Schritte beschrieben. Das Framework ermöglicht Organisationen:

- Erstellen Sie autonome Agenten, die über Konversationen in natürlicher Sprache kommunizieren.
- Implementieren Sie asynchrone, ereignisgesteuerte Interaktionen zwischen mehreren Agenten.
- Kombinieren Sie bei Bedarf einen vollständig autonomen Betrieb mit optionaler menschlicher Aufsicht.
- Entwickeln Sie spezialisierte Agenten für verschiedene Geschäftsfunktionen, die im Dialog zusammenarbeiten.

Dieser dialogorientierte Ansatz macht die Argumentation des autonomen Systems transparent und für Geschäftsanwender zugänglich. Decision-makers kann den Dialog zwischen den Akteuren beobachten, um zu verstehen, wie Schlussfolgerungen gezogen werden, und optional an der Konversation teilnehmen, wenn menschliches Urteilsvermögen gefragt ist.

Real-world Beispiel für AutoGen

Magentic-Oneist ein generalistisches Open-Source-Multiagentensystem, das entwickelt wurde, um komplexe, mehrstufige Aufgaben in unterschiedlichen Umgebungen autonom zu lösen, wie im Blog AI Frontiers beschrieben. Microsoft Sein Herzstück ist der Orchestrator-Agent, der übergeordnete Ziele zerlegt und Fortschritte mithilfe strukturierter Ledger verfolgt. Dieser Agent delegiert Unteraufgaben an spezialisierte Agenten (wieWebSurfer,, undComputerTerminal) und passt sich dynamisch an FileSurferCoder, indem er bei Bedarf neu plant.

Das System basiert auf dem AutoGen Framework, ist modellunabhängig und verwendet standardmäßig GPT-4o. Es erreicht bei Benchmarks wie, und eine Leistung auf dem neuesten Stand der Technik — und das alles ohne aufgabenspezifische Feinabstimmung. GAIA AssistantBench WebArena Darüber hinaus unterstützt es modulare Erweiterbarkeit und eine strenge Evaluierung anhand von Vorschlägen. AutoGenBench

LlamaIndex

LlamaIndexist ein Datenframework, das speziell für die Verbindung großer Sprachmodelle (LLMs) mit externen Datenquellen entwickelt wurde, um anspruchsvolle Retrieval Augmented Generation (RAG) und agentische KI-Anwendungen zu ermöglichen. Das Framework bietet Abstraktionen und beschleunigte Entwicklungsworkflows für agentische Systeme, benutzerdefinierte Orchestrierungsmuster und Systemintegrationen, die die Produktionszeit für wissensgestützte KI-Lösungen reduzieren.

Hauptmerkmale von LlamaIndex

LlamaIndexbietet einen umfassenden Funktionsumfang, der sich besonders für KI-Anwendungen in Unternehmen eignet:

- Data-centric Architektur — Hervorragend geeignet für das Erfassen, Indexieren und Abrufen von Informationen aus über 100 Datenformaten, darunter PDF-Dateien, Word-Dokumente, Microsoft Tabellen und mehr. Das Framework wandelt Unternehmensdaten in abfragbare Wissensdatenbanken um, die für KI-Agenten optimiert sind. Weitere Informationen finden Sie in der LlamaIndex-Dokumentation.
- Production-ready Bereitstellung — LlamaIndex bietet sowohl Open-Source-Frameworks als auch Managed Services und bietet Funktionen auf Unternehmensebene wie SicherheitskontrollenLlamaCloud, Skalierbarkeit, Integrationen zur Beobachtbarkeit und Flexibilität bei der Bereitstellung. Weitere Informationen finden Sie in der Framework-Dokumentation. LlamaIndex
- Erweiterte Dokumentenverarbeitung — LlamaCloud bietet Funktionen zum Analysieren, Extrahieren, Indexieren und Abrufen von Dokumenten, die komplexe Layouts, verschachtelte Tabellen, multimodale Inhalte und sogar handschriftliche Notizen verarbeiten. Dieses ausgeklügelte Parsing ermöglicht es Mitarbeitern, effektiv mit realen Unternehmensdokumenten zu arbeiten, die Diagramme, Diagramme und komplexe Formatierungen enthalten. Weitere Informationen finden Sie in der LlamaCloud-Dokumentation.

- Workflow-Orchestrierung — LlamaAgents bietet eine ereignisgesteuerte, asynchrone Orchestrierungs-Engine zum Aufbau mehrstufiger Agentensysteme. Workflows unterstützen komplexe Muster wie Schleifen, parallel Ausführung, bedingte Verzweigung und statusbehaftete Wiederaufnahme, wodurch sie sich ideal für anspruchsvolle Agenteninteraktionen eignen. Weitere Informationen finden Sie in der LlamaIndex Workflow-Dokumentation.
- Agentenabruffunktionen — Erweiterte Abrufmodi wie Hybridsuche, semantische Suche und automatisches Routing, die auf intelligente Weise die beste Abrufstrategie für jede Abfrage bestimmen. Das Framework unterstützt den kombinierten Abruf mehrerer Wissensdatenbanken mit Neueinstufungen zur Erhöhung der Genauigkeit. Weitere Informationen finden Sie in der LlamaIndex RAG-Dokumentation.
- Beobachtbarkeit und Bewertung — LlamaIndex lässt sich in eine Vielzahl von Beobachtungs- und Bewertungsinstrumenten integrieren. Diese Integrationsfunktion hilft Ihnen dabei, Ihre Anwendungen zu verfolgen und zu debuggen, ihre Leistung zu bewerten und die Kosten zu überwachen. Weitere Informationen finden Sie in der Dokumentation Tracing, Debugging and Evaluating. LlamaIndex

Wann sollte es verwendet werden LlamaIndex

LlamaIndexeignet sich besonders gut für agentische KI-Szenarien, in denen datenintensive Workflows und Wissensmanagement im Vordergrund stehen:

- Document-heavy Anwendungen, bei denen Mitarbeiter große Mengen an Unternehmensdokumenten wie Verträgen, Berichten, Handbüchern und behördlichen Unterlagen verarbeiten, analysieren und Erkenntnisse daraus gewinnen müssen
- Schnelles Prototyping bis hin zu Produktionsszenarien, in denen Unternehmen schnell dokumentenorientierte Agenten ohne großen Aufwand für das Infrastrukturmanagement erstellen und einsetzen möchten
- RAG-first Architekturen, bei denen Abrufgenauigkeit und Kontextrelevanz im Vordergrund stehen, insbesondere bei der Arbeit mit komplexen, multimodalen Dokumenten, die Tabellen, Bilder und strukturierte Daten enthalten
- Multi-agent Dokumenten-Workflows, die spezialisierte Agenten für verschiedene Aspekte der Dokumentenverarbeitung erfordern, z. B. Analyse, Zusammenfassung und Konformitätsprüfung

Implementierungsansatz für LlamaIndex

LlamaIndex bietet sowohl Bausteine auf niedriger Ebene als auch Abstraktionen auf hoher Ebene, die unterschiedlichen Implementierungsansätzen Rechnung tragen:

- Schnelle Entwicklung funktionaler RAG-Anwendungen in nur wenigen Codezeilen mithilfe von LlamaIndex High-Level-APIs. Dieser Ansatz ist für Geschäftsteams und Entwickler LlamaIndex zugänglich, die noch keine Erfahrung mit agentischer KI haben.
- Unternehmensintegration LlamaHub für beliebte Unternehmenssysteme wie SharePoint Amazon Simple Storage Service (Amazon S3), Datenbanken und APIs. Dieser Ansatz ermöglicht eine nahtlose Integration in die bestehende Dateninfrastruktur.
- Flexible Bereitstellungsoptionen zwischen selbst gehosteten Open-Source-Bereitstellungen für maximale Kontrolle oder LlamaCloud verwalteten Diensten für weniger Betriebskosten und Unternehmensfunktionen.
- Anwendungen können mit einfachen Abfrage-Engines beginnen und nach und nach um Agentenfunktionen, Orchestrierung mehrerer Agenten und komplexe Workflows erweitern, wenn sich die Anforderungen ändern.

Real-world Beispiel für LlamaIndex

Dieses Beispiel konzentriert sich auf eine Tochtergesellschaft eines Luft- und Raumfahrtunternehmens, das sich auf Navigations- und Betriebslösungen für die Luftfahrt spezialisiert hat. Sie müssen sich einer wachsenden Herausforderung stellen, zu der die Erprobung unkoordinierter KI-Chatbot-Versuche gehört. Die Versuche führten zu wiederholter Arbeit, langen Entwicklungszyklen, Compliance-Hindernissen und isolierten Implementierungen im gesamten Unternehmen.

Sie entwickelten ein einheitliches Agenten-Framework, eine wiederverwendbare, auf Vorlagen basierende Lösung, die auf dem LlamaIndex Open-Source-Framework basiert und die Agentenerstellung erheblich effizienter macht. Sie verglichen mehrere konkurrierende Frameworks, sowohl kettenorientiert als auch grafisch. Letztlich entschieden sie sich LlamaIndex für drei entscheidende Vorteile: das flexible Design, die modularen Komponenten und die produktionsreife Orchestrierungssteuerung.

Die Plattform reduziert die Zeit für die Entwicklung und Bereitstellung von Agenten um 87% von 512 auf 64 Stunden. Diese Reduzierung wurde dadurch erreicht, dass Teams Agenten mit etwa

50 Codezeilen und einer JSON-Konfigurationsdatei erstellen konnten. Die Teams nutzten ein einheitliches Framework mit integrierter Sicherheit, Compliance und privilegiertem Systemzugriff. Weitere Informationen finden Sie in den Fallstudien von LlamaIndex Kunden.

Vergleich agentischer KI-Frameworks

Berücksichtigen Sie bei der Auswahl eines agentischen KI-Frameworks für die Entwicklung autonomer Agenten, wie jede Option Ihren spezifischen Anforderungen entspricht. Berücksichtigen Sie nicht nur die technischen Fähigkeiten, sondern auch die organisatorische Eignung, einschließlich der Expertise des Teams, der vorhandenen Infrastruktur und der langfristigen Wartungsanforderungen. Viele Unternehmen könnten von einem hybriden Ansatz profitieren, bei dem mehrere Frameworks für verschiedene Komponenten ihres autonomen KI-Ökosystems genutzt werden.

In der folgenden Tabelle werden die Reifegrade (am stärksten, am stärksten, angemessen oder schwach) der einzelnen Frameworks anhand der wichtigsten technischen Dimensionen verglichen. Für jedes Framework enthält die Tabelle auch Informationen zu den Optionen für den Einsatz in der Produktion und zur Komplexität der Lernkurve.

Framework	AWS Integration	Autonome Unterstützung für mehrere Agenten	Komplexität des autonomen Workflows	Multimodale Fähigkeiten	Auswahl des Fundamentmodells	LLM-API-Integration	Einsatz in der Produktion	Lernkurve
AutoGen	Schwach	Stark	Stark	Ausreichend	Ausreichend	Stark	Mach es selbst (DIY)	Steil
CrewAI	Schwach	Stark	Ausreichend	Schwach	Ausreichend	Ausreichend	DIY	Mittel
LangChain / LangGraph	Ausreichend	Stark	Am stärksten	Am stärksten	Am stärksten	Am stärksten	Plattform oder DIY	Steil

LlamaIndex	Ausreichend	Ausreichend	Stark	Ausreichend	Stark	Stark	Plattform oder DIY	Mittel
Strands Agents	Am stärksten	Stark	Am stärksten	Stark	Stark	Am stärksten	DIY	Mittel

Überlegungen bei der Auswahl eines agentischen KI-Frameworks

Berücksichtigen Sie bei der Entwicklung autonomer Agenten die folgenden Schlüsselfaktoren:

- AWS Infrastrukturintegration — Organizations, in die viel investiert AWS wird, werden am meisten von den nativen Integrationen von Strands Agents AWS-Services für autonome Workflows profitieren. Weitere Informationen finden Sie unter AWS Weekly Roundup (Blog)AWS .
- Auswahl des Foundation-Modells — Überlegen Sie anhand der Argumentationsanforderungen Ihres autonomen Agenten, welches Framework Ihre bevorzugten Foundation-Modelle am besten unterstützt (z. B. Amazon Nova-Modelle auf Amazon Bedrock oder Anthropic Claude). Weitere Informationen finden Sie auf der Website unter Building Effective Agents. Anthropic
- LLM-API-Integration — Evaluieren Sie Frameworks auf der Grundlage ihrer Integration mit Ihren bevorzugten Large Language Model (LLM) -Serviceschnittstellen (z. B. Amazon Bedrock oderOpenAI) für die Produktionsbereitstellung. Weitere Informationen finden Sie in der Dokumentation unter Model Interfaces. Strands Agents
- Multimodale Anforderungen — Bei autonomen Agenten, die Text, Bilder und Sprache verarbeiten müssen, sollten Sie die multimodalen Fähigkeiten der einzelnen Frameworks berücksichtigen. Weitere Informationen finden Sie in der Dokumentation unter Multimodalität. LangChain
- Komplexität autonomer Workflows — Komplexere autonome Workflows mit ausgefeilter Zustandsverwaltung könnten die fortschrittlichen Funktionen von State Machines begünstigen. von. LangGraph
- Autonome Teamzusammenarbeit — Projekte, die eine explizite, rollenbasierte, autonome Zusammenarbeit zwischen spezialisierten Mitarbeitern erfordern, können von der teamorientierten Architektur von profitieren. CrewAI
- Paradigma der autonomen Entwicklung — Teams, die asynchrone Konversationsmuster für autonome Agenten bevorzugen, bevorzugen möglicherweise die ereignisgesteuerte Architektur von. AutoGen

- Verwalteter oder codebasierter Ansatz — Organizations, die eine vollständig verwaltete Erfahrung mit minimalem Programmieraufwand wünschen, sollten Amazon Bedrock Agents in Betracht ziehen. Organizations, die eine tiefere Anpassung benötigen, bevorzugen Strands Agents möglicherweise andere Frameworks mit speziellen Funktionen, die besser auf die spezifischen Anforderungen autonomer Agenten zugeschnitten sind.
- Produktionsreife autonomer Systeme — Erwägen Sie Bereitstellungsoptionen, Überwachungsmöglichkeiten und Unternehmensfunktionen für autonome Produktionsagenten.

Plattformen

Agentic KI-Plattformen bieten die grundlegenden Laufzeit-, Orchestrierungs- und Integrationsebenen, die für die Bereitstellung, Skalierung und Verwaltung von agentischen Systemen in Produktionsqualität erforderlich sind. Frameworks definieren, wie Agenten aufgebaut sind, und Protokolle regeln, wie sie kommunizieren. Plattformen bieten die Umgebung, in der diese Agenten sicher und skalierbar arbeiten, zusammenarbeiten und sich weiterentwickeln können.

Agentenplattformen kombinieren Modellausführung, Kontextmanagement, Toolintegration, Beobachtbarkeit und Governance-Funktionen in einheitlichen Umgebungen. Diese Plattformen ermöglichen es Unternehmen, vom Experimentieren zur Bereitstellung auf Unternehmensebene überzugehen.

In diesem Abschnitt:

- Warum Plattformen wichtig sind
- Arten von agentischen KI-Plattformen
- Überlegungen zur Plattformauswahl
- Agenten für Amazon Bedrock
- Amazon Grundgestein AgentCore

Warum Plattformen wichtig sind

Agentische KI-Plattformen sind für Unternehmen, die autonome Systeme in der Produktion operationalisieren möchten, von entscheidender Bedeutung. Sie bieten die folgenden Funktionen:

- Stellen Sie Runtime-Orchestrierung für das Hosten, Skalieren und Koordinieren von Agenten bereit.
- Verwalten Sie Status, Kontext und Speicher in Workflows mit mehreren Agenten.
- Bieten Sie Sicherheits-, Identitäts- und Governance-Kontrollen an, die den Unternehmensstandards entsprechen.
- Integrieren Sie mithilfe von Standards APIs oder Protokollen in Tooling-Ökosysteme und externe Systeme.
- Sorgen Sie für Beobachtbarkeit und Überprüfbarkeit aller Agenteninteraktionen und Ereignisabläufe.

- Support modellübergreifende Interoperabilität, sodass Agenten mehrere Basismodelle in einer einzigen Umgebung verwenden können.

Diese Funktionen machen aus einzelnen Agenten koordinierte, anpassungsfähige Systeme, die innerhalb der Unternehmens- und behördlichen Grenzen zuverlässig funktionieren können.

Arten von agentischen KI-Plattformen

Agentische KI-Plattformen lassen sich in der Regel in eine oder mehrere der folgenden Kategorien einteilen:

- Managed Agent — Vollständig verwaltete Plattformen bieten integrierte Infrastruktur-, Speicher- und Orchestrierungsfunktionen. Sie reduzieren den betrieblichen Aufwand und beschleunigen die Produktionszeit.
- Open-Source-Orchestrierung — Agenturbasierte Open-Source-Plattformen bieten Flexibilität und Transparenz für Unternehmen, die anpassbare Umgebungen oder die Bereitstellung vor Ort bevorzugen.
- Hybrides Unternehmen — Hybride Plattformen integrieren verwaltete und selbst gehostete Komponenten und kombinieren so die Skalierbarkeit von Cloud-verwalteten Diensten mit der Steuerung von Unternehmenssystemen.

Überlegungen zur Plattformauswahl

Bei der Auswahl oder Gestaltung einer agentischen KI-Plattform sollten Unternehmen Folgendes berücksichtigen:

- Integrationstiefe — Bewerten Sie, wie gut sich die Plattform in bestehende Datenquellen, Tools und Protokolle integrieren lässt.
- Skalierbarkeit — Stellen Sie sicher, dass die Plattform dynamisch skaliert werden kann, um autonome Workloads und die Zusammenarbeit mehrerer Agenten zu unterstützen.
- Sicherheit und Compliance — Beurteilen Sie die Datenschutz-, Verschlüsselungs- und Governance-Funktionen anhand organisatorischer und regionaler Anforderungen.
- Erweiterbarkeit — Wählen Sie Plattformen mit modularen Architekturen, die es ermöglichen, im Laufe der Zeit neue Tools, Modelle oder Agenten hinzuzufügen.

- Beobachtbarkeit — Bevorzugen Sie Plattformen, die detaillierte Telemetrie-, Rückverfolgbarkeits- und Auditprotokolle für Agenteninteraktionen bieten.
- Kosteneffizienz — Ziehen Sie serverlose oder nutzungsbasierte Modelle in Betracht, um die Kosten für variable Workloads zu optimieren.

Agenten für Amazon Bedrock

Amazon Bedrock Agents ist ein vollständig verwalteter Service, mit dem Sie autonome Agenten in Ihren Anwendungen erstellen und konfigurieren können. Er kann Interaktionen zwischen Basismodellen, Datenquellen, Softwareanwendungen und Benutzerkonversationen orchestrieren. Dank des optimierten Ansatzes zur Erstellung von Agenten müssen Sie keine Kapazität bereitstellen, die Infrastruktur verwalten oder benutzerdefinierten Code schreiben.

Hauptfunktionen von Amazon Bedrock Agents

Amazon Bedrock Agents umfasst die folgenden Hauptfunktionen:

- Vollständig verwalteter Service — Umfassendes Infrastrukturmanagement, ohne dass Kapazitäten bereitgestellt oder die zugrunde liegenden Systeme verwaltet werden müssen. Weitere Informationen finden Sie unter Automatisieren von Aufgaben in Ihrer Anwendung mithilfe von KI-Agenten in der Amazon Bedrock-Dokumentation.
- API-gesteuerte Entwicklung — Definieren Sie Agenten und führen Sie sie durch einfache API-Aufrufe aus, indem Sie Modelle, Anweisungen, Tools und Konfigurationsparameter angeben. Weitere Informationen finden Sie unter Agenten manuell erstellen und konfigurieren in der Amazon Bedrock-Dokumentation.
- Aktionsgruppen — Definieren Sie spezifische Aktionen, die Ihr Agent ausführen kann, indem Sie Aktionsgruppen mit API-Schemas erstellen. Weitere Informationen finden Sie in der Amazon Bedrock-Dokumentation unter Verwenden von Aktionsgruppen zur Definition von Aktionen, die Ihr Agent ausführen soll.
- Wissensdatenbank-Integration — Stellen Sie eine nahtlose Verbindung zu Amazon Bedrock Knowledge Bases her, um die Antworten der Agenten mit den Daten Ihres Unternehmens zu ergänzen. Weitere Informationen finden Sie unter Verbessern Sie die Antwortgenerierung für Ihren Agenten mit der Wissensdatenbank in der Amazon Bedrock-Dokumentation.
- Erweiterte Vorlagen für Eingabeaufforderungen — Passen Sie das Verhalten Ihrer Agenten mithilfe von Vorlagen für die Vorverarbeitung, Orchestrierung, Generierung und Nachbearbeitung von Antworten in der Wissensdatenbank individuell an. Weitere Informationen finden Sie in der Amazon

Bedrock-Dokumentation unter Verbessern Sie die Genauigkeit des Agenten mithilfe erweiterter Eingabeaufforderungsvorlagen in Amazon Bedrock.

- Rückverfolgung und Beobachtbarkeit — Verfolgen Sie den step-by-step Argumentationsprozess des Agenten mithilfe der integrierten Nachverfolgungsfunktionen. Weitere Informationen finden Sie unter Nachverfolgen des step-by-step Argumentationsprozesses des Agenten mithilfe von Trace in der Amazon Bedrock-Dokumentation.
- Versionierung und Aliase — Erstellen Sie mehrere Versionen Ihres Agenten und stellen Sie sie über Aliase für kontrollierte Rollouts bereit. Weitere Informationen finden Sie in der Amazon Bedrock-Dokumentation unter Bereitstellen und Verwenden eines Amazon Bedrock-Agenten in Ihrer Anwendung.

Wann sollten Sie Amazon Bedrock Agents verwenden

Amazon Bedrock Agents eignet sich besonders gut für autonome Agentenszenarien, darunter:

- Organizations, die ein vollständig verwaltetes Erlebnis für die Erstellung und Bereitstellung von Agenten wünschen, ohne die Infrastruktur verwalten zu müssen
- Projekte, die eine schnelle Entwicklung und Bereitstellung von Agenten durch Konfiguration und nicht durch Code erfordern
- Anwendungsfälle, die von einer engen Integration mit anderen Amazon Bedrock-Funktionen wie Knowledge Bases und Guardrails profitieren
- Teams, die nicht über die internen Ressourcen verfügen, um Agenten von Grund auf neu zu erstellen, benötigen aber produktionsreife autonome Funktionen

Implementierungsansatz für Amazon Bedrock Agents

Amazon Bedrock Agents bietet einen konfigurationsbasierten Implementierungsansatz für Geschäftsbeteiligte. Der Service ermöglicht Unternehmen:

- Definieren Sie Agenten über die AWS-Managementkonsole oder API-Aufrufe, ohne komplexen Code zu schreiben.
- Erstellen Sie Aktionsgruppen, die APIs angeben, welche Operationen der Agent ausführen kann.
- Connect Wissensdatenbanken, um dem Agenten domänenspezifische Informationen zur Verfügung zu stellen.
- Testen und wiederholen Sie das Verhalten von Agenten über eine visuelle Oberfläche.

Dieser verwaltete Ansatz ermöglicht es Geschäftsteams, autonome Agenten schnell zu entwickeln und einzusetzen, ohne über fundiertes technisches Fachwissen in der KI-Modellentwicklung oder im Infrastrukturmanagement verfügen zu müssen.

Ein Beispiel aus der Praxis für Amazon Bedrock Agents

Eine in diesem AWS Blogbeitrag beschriebene Financial Operations (FinOps) -Lösung verwendet das Amazon Bedrock Multi-Agent-Framework, um einen KI-gesteuerten Cloud-Kostenmanagement-Assistenten zu erstellen. Das kostengünstige Amazon Nova Foundation-Modell unterstützt die Lösung, bei der ein zentraler FinOps Supervisor-Agent Aufgaben an spezialisierte Agenten delegiert. Diese Agenten rufen Ausgabendaten mithilfe von Daten zu AWS Ausgaben ab AWS Cost Explorer und generieren daraus Empfehlungen zur Kosteneinsparung. AWS Trusted Advisor

Das System umfasst sicheren Benutzerzugriff über Amazon Cognito, ein auf dem gehostetes Frontend AWS Amplify, und AWS Lambda Aktionsgruppen für Analysen und Prognosen in Echtzeit. Finanzteams können Fragen in natürlicher Sprache stellen, z. B. „Wie hoch waren meine Kosten im Februar 2025?“ Das System reagiert mit detaillierten Aufschlüsselungen, Optimierungsvorschlägen und Prognosen — alles innerhalb einer skalierbaren, serverlosen Architektur, die mithilfe von AWS CloudFormation

Amazon Grundgestein AgentCore

Amazon Bedrock AgentCore ist eine Agentenplattform für die sichere und skalierbare Entwicklung, Bereitstellung und den Betrieb hochleistungsfähiger Agenten unter Verwendung eines beliebigen Frameworks, Modells oder Protokolls. Mit AgentCore ihr können Sie Folgendes tun, und das alles ohne Infrastrukturmanagement:

- Erstellen Sie Agenten schneller.
- Ermöglichen Sie es Agenten, tool- und datenübergreifend Maßnahmen zu ergreifen.
- Führen Sie Agenten sicher mit niedriger Latenz und längeren Laufzeiten aus.
- Überwachen Sie Agenten in der Produktion.

AgentCore macht den undifferenzierten Aufwand beim Aufbau einer speziellen Agenteninfrastruktur überflüssig und ermöglicht es Ihnen, Ihre Agenten schneller zur Produktion zu bringen. Die Dienste können zusammen oder unabhängig voneinander genutzt werden und sind mit allen Frameworks kompatibel, einschließlichCrewAI, LangGraphLlamaIndex, undStrands Agents. AgentCoreist auch mit

jedem Fundamentmodell kompatibel, das innerhalb oder außerhalb von Amazon Bedrock erhältlich ist, und bietet so ultimative Flexibilität.

AgentCore besteht aus mehreren wichtigen Diensten:

- Amazon Bedrock AgentCore Runtime — Bietet eine sichere, serverlose, skalierbare Umgebung zum Hosten und Ausführen Ihrer Agenten, ohne dass Sie die Infrastruktur verwalten müssen, die für die Bereitstellung und Ausführung von KI-Agenten oder -Tools erforderlich ist.
- Amazon Bedrock AgentCore Memory — Bietet ein verwaltetes Speichersystem, das es Agenten ermöglicht, den Kontext von Interaktionen für persönlichere und kohärentere Konversationen beizubehalten, indem sowohl sofortiges als auch langfristiges Wissen erhalten bleibt.
- Amazon Bedrock AgentCore Gateway — Vereinfacht den Prozess der Erstellung, Sicherung und Suche nach den richtigen Tools für Agenten. Mit AgentCore Gateway können Entwickler Lambda-Funktionen und bestehende Dienste in Model Context Protocol (MCP) -kompatible Tools konvertieren APIs und sie Agenten zur Verfügung stellen.
- Amazon Bedrock AgentCore Identity — Bietet einen sicheren, skalierbaren Identitäts- und Zugriffsverwaltungsservice für Agenten, der die Entwicklung von KI-Agenten beschleunigt. Mit AgentCore Identity können Sie Agenten eindeutige, überprüfbare Identitäten zuweisen und so eine detaillierte Zugriffskontrolle und sichere agentengestützte Interaktionen mit Unternehmenssystemen ermöglichen.
- AgentCore Integrierte Tools von Amazon Bedrock — Ermöglicht es Ihnen, integrierte Tools zu verwenden, um Ihren Entwicklungs- und Test-Workflow zu verbessern. Verwenden Sie diese Tools, um effektiv mit Ihrer Anwendung zu interagieren, sodass KI-Agenten Code in Sandbox-Umgebungen sicher schreiben und ausführen können. Verwenden Sie das Browser-Tool, damit KI-Agenten in großem Umfang mit Websites interagieren können.
- Amazon Bedrock AgentCore Observability — Bietet Protokollierungs- und Überwachungsfunktionen, mit denen Sie in Echtzeit Einblick in die Leistung und das Verhalten Ihres Agenten erhalten, um Debugging und Optimierung zu erleichtern.

Die wichtigsten Funktionen von AgentCore

AgentCore umfasst die folgenden Hauptfunktionen:

- Vollständig verwaltet und erweiterbar — AgentCore ist ein vollständig verwalteter Service, d. h. er AWS kümmert sich um die zugrunde liegende Infrastruktur und Wartung. Er ist außerdem erweiterbar, sodass Sie die Funktionalität Ihrer Agenten anpassen und erweitern können. Weitere

Informationen finden Sie in der AgentCore Dokumentation unter Erste Schritte mit AgentCore Runtime.

- Langzeit- und Kurzzeitgedächtnis — Sorgen Sie für persönlichere und relevantere Interaktionen, indem Sie Agenten mit einem Speichersystem ausstatten, mit dem sie den Kontext aktueller Konversationen und langfristiges Wissen abrufen können. Weitere Informationen finden Sie in der AgentCore Dokumentation unter Erste Schritte mit dem AgentCore Gedächtnis.
- Vereinfachte Entwicklung und Integration von Tools — Ermöglichen Sie es Ihren Agenten, Tools über einen einzigen, sicheren Endpunkt zu finden und zu verwenden. Verwandeln Sie Ihre vorhandenen Unternehmensressourcen mit nur wenigen Codezeilen schnell in agentenfähige Tools, sodass sich Entwickler auf die Entwicklung einzigartiger Funktionen konzentrieren können. Weitere Informationen finden Sie in der Dokumentation unter Erste Schritte mit AgentCore Gateway. AgentCore
- Sichere und skalierbare Infrastruktur — AgentCore bietet eine sichere und skalierbare Umgebung für die Bereitstellung und den Betrieb von Agenten. Sie umfasst Funktionen für Identitäts- und Zugriffsmanagement, Datenverschlüsselung und Netzwerksicherheit. Weitere Informationen finden Sie in der AgentCore Dokumentation unter Erste Schritte mit AgentCore Identity.
- Integration mit einer Vielzahl von Tools — Ermöglicht es Ihnen, Ihre Agenten mit einer Vielzahl von Tools zu integrieren, darunter einen Codeinterpreter und ein Browser-Tool, das Sie mithilfe der AgentCore integrierten Tools erstellen können. Weitere Informationen finden Sie in der Dokumentation unter Erste Schritte mit AgentCore Code Interpreter und Erste Schritte mit dem AgentCore AgentCore Browser.
- Umfassende Beobachtbarkeit und Überwachung — Verschaffen Sie sich einen umfassenden Einblick in Ihre Agenten mit umfassenden Tools, mit denen Sie deren Leistung in der Produktion verfolgen, debuggen und überwachen können. Visualisieren Sie den gesamten Ausführungspfad des Agenten, um seine Argumentation zu überprüfen und Fehler zu beheben. Verwenden Sie Echtzeit-Dashboards und standardisierte Telemetriedaten, um wichtige Betriebskennzahlen zu verfolgen. Weitere Informationen finden Sie in der Dokumentation unter Hinzufügen von Observability zu Ihren Amazon AgentCore Bedrock-Ressourcen. AgentCore

Wann sollte es verwendet werden AgentCore

AgentCore eignet sich besonders gut für Szenarien mit autonomen Agenten, darunter:

- Organizations, die mit einem vollständig verwalteten Service, der sich um Infrastruktur, Sicherheit, integrierte Tools, Beobachtbarkeit und Skalierung kümmert, die Entwicklung beschleunigen und die Betriebskosten senken möchten
- Projekte, die Flexibilität mit modularen Services benötigen, die zusammen oder unabhängig voneinander funktionieren und mit jedem Framework wie CrewAI Oder LangGraph und jedem Basismodell aus jeder Quelle kompatibel sind
- Anwendungsfälle, die zustandsorientierte, dialogorientierte Agenten erfordern, die den Kontext wahren und aus vergangenen Interaktionen lernen müssen, um personalisierte und relevante Antworten geben zu können
- Agenten sind in der Lage, komplexe Aufgaben durch einfache Integration mit verschiedenen Anwendungen, Datenquellen und APIs

Implementierungsansatz für AgentCore

AgentCore ist für Unternehmen konzipiert, die KI-Agenten von der Machbarkeitsstudie, die mithilfe von Open Source- oder kundenspezifischen Agenten-Frameworks erstellt wurde, in die Produktion überführen möchten. Mit AgentCore können Unternehmen Folgendes tun:

- Stellen Sie Agenten sicher auf einer serverlosen Infrastruktur bereit, die jedes Framework und jedes Modell unterstützt, mit Sitzungsisolierung und integriertem Identitäts- und Zugriffsmanagement für end-to-end Sicherheit und Compliance. Mithilfe des Starter-Toolkits können Sie im Handumdrehen AgentCore Runtime-Agenten für führende Agenten-Frameworks erstellen.
- Verbessern Sie die Agents durch die Integration von persistentem Speicher zur Kontexterhaltung und vereinfachen Sie so die Entwicklung und Integration von Tools über AgentCore Gateway. Nutzen Sie das integrierte Browser-Tool und den Codeinterpreter für erweiterte Workflows.
- Verfolgen, debuggen und überwachen Sie KI-Agenten in der Produktion mithilfe von Observability-Dashboards, die von Amazon CloudWatch Application Insights unterstützt werdenOpenTelemetry, und verfolgen Sie wichtige AgentCore Ressourcenmetriken (Laufzeit, Speicher, Gateway und Tools).
- Beschleunigen Sie die Bereitstellung und Innovation mit vollständig verwalteten, modularen Services, zusammensetzbaren Blöcken zusammen oder unabhängig voneinander, mit beliebigen Agenten-Frameworks und Modellanbietern. Diese Flexibilität hilft Unternehmen dabei, schneller vom Prototyp zur Produktion überzugehen.

Dieser verwaltete Ansatz ermöglicht es Unternehmen, schnell und sicher KI-Agenten und Multi-Agenten-Systeme auf Unternehmensebene in jeder Größenordnung zu erstellen, bereitzustellen und auszuführen.

Ein Beispiel aus der Praxis für AgentCore

AWS hat beobachtet, dass eine der größten Banken Lateinamerikas AI/ML seit Jahren ein hyperpersonalisiertes und sicheres digitales Bankerlebnis anbietet. Die Bank erweitert ihre KI-Dienste, AgentCore um ihren Kunden intuitive Interaktionen, mehr Sicherheit und mehr Automatisierung zu bieten. Nach Angaben des CTO AgentCore wird erwartet, dass es ihre Bemühungen unterstützt, die Verpflichtungen der Kunden in großem Umfang zu erfüllen. AgentCore bietet ihren Entwicklern die Tools und die Flexibilität, um Agenten aufzubauen und zu verwalten, und trägt gleichzeitig zur Einhaltung der Finanzvorschriften bei.

Protokolle

KI-Agenten benötigen standardisierte Kommunikationsprotokolle, um mit anderen Agenten und Diensten interagieren zu können. Organizations, die Agentenarchitekturen implementieren, stehen vor großen Herausforderungen in Bezug auf Interoperabilität, Herstellerunabhängigkeit und Zukunftssicherheit ihrer Investitionen.

Dieser Abschnitt hilft Ihnen, sich in der agent-to-agent Protokolllandschaft zurechtzufinden, wobei der Schwerpunkt auf offenen Standards liegt, die Flexibilität und Interoperabilität maximieren. (Informationen zu agent-to-tool Protokollen finden Sie weiter unten in diesem Handbuch unter Strategie zur Integration von Tools.)

In diesem Abschnitt wird das Model Context Protocol (MCP) vorgestellt, ein offener Standard, der ursprünglich 2024 Anthropic entwickelt wurde. Heute unterstützt er MCP AWS aktiv durch Beiträge zur Entwicklung und Implementierung des Protokolls. AWS arbeitet mit führenden Open-Source-Agenten-Frameworks zusammen, darunter, und LangGraph CrewAILlamaIndex, um die future der Agentenkommunikation auf dem Protokoll zu gestalten. Weitere Informationen finden Sie unter Offene Protokolle für Agenten-Interoperabilität, Teil 1: Agentenübergreifende Kommunikation auf MCP (Blog).AWS

In diesem Abschnitt:

- Warum die Protokollauswahl wichtig ist
- Agent-to-agent Protokolle
- Auswahl von Agentenprotokollen
- Implementierungsstrategie für behördliche Protokolle
- Erste Schritte mit MCP
- ???

Warum die Protokollauswahl wichtig ist

Die Protokollauswahl bestimmt grundlegend, wie Sie Ihre KI-Agentenarchitektur aufbauen und weiterentwickeln können. Durch die Auswahl von Protokollen, die die Portabilität zwischen Agenten-Frameworks unterstützen, erhalten Sie die Flexibilität, verschiedene Agentensysteme und Workflows zu kombinieren, um Ihren spezifischen Anforderungen gerecht zu werden.

Offene Protokolle ermöglichen es Ihnen, Agenten über mehrere Frameworks hinweg zu integrieren. Verwenden Sie es beispielsweise LangChain für die schnelle Prototypenentwicklung und Implementierung von ProduktionssystemenStrands Agents, die über ein gemeinsames Protokoll wie MCP oder das Agent2Agent (A2A) -Protokoll kommunizieren. Diese Flexibilität reduziert die Abhängigkeit von bestimmten KI-Anbietern, vereinfacht die Integration in bestehende Systeme und ermöglicht es Ihnen, die Fähigkeiten der Agenten im Laufe der Zeit zu verbessern.

Gut konzipierte Protokolle sorgen außerdem für einheitliche Sicherheitsmuster für die Authentifizierung und Autorisierung in Ihrem gesamten Agenten-Ökosystem. Am wichtigsten ist jedoch, dass Sie dank der Protokollportabilität die Freiheit haben, neue Agenten-Frameworks und -Funktionen zu übernehmen, sobald diese verfügbar sind. Die Wahl offener Protokolle schützt Ihre Investitionen in die Agentenentwicklung und gewährleistet gleichzeitig die Interoperabilität mit Systemen von Drittanbietern.

Vorteile offener Protokolle

Wenn Sie Ihre eigenen Erweiterungen implementieren oder benutzerdefinierte Agentensysteme erstellen, bieten offene Protokolle überzeugende Vorteile:

- Dokumentation und Transparenz — Sorgen in der Regel für eine umfassende Dokumentation und transparente Implementierungen
- Community-Support — Zugang zu breiteren Entwickler-Communitys für Problemlösungen und Best Practices
- Interoperabilitätsgarantien — Bessere Gewissheit, dass Ihre Erweiterungen in verschiedenen Implementierungen funktionieren
- Künftige Kompatibilität — Geringeres Risiko, dass Änderungen nicht mehr funktionieren oder nicht mehr unterstützt werden
- Einfluss auf die Entwicklung — Gelegenheit, zur Protokollentwicklung beizutragen

Agent-to-agent Protokolle

Die folgende Tabelle bietet einen Überblick über Agentenprotokolle, mit denen mehrere Agenten zusammenarbeiten, Aufgaben delegieren und Informationen austauschen können.

Protocol (Protokoll)	Ideal für	Überlegungen

MCP-Kommunikation zwischen Agenten	Organizations, die nach flexiblen Mustern für die Zusammenarbeit mit Agenten suchen	• Eine von AWS dieser Organisation vorgeschl agene Erweiterung des Model Context Protocol (MCP) baut auf der bestehenden Kommunika tionsgrundlage agent-to-agent auf • Ermöglicht eine nahtlose Zusammenarbeit der Agenten mit OAuth basierter Sicherheit
A2A-Protokoll	Plattformübergreifende Agenten-Ökosysteme	• Unterstützt von Google • Neuerer Standard mit geringerer Akzeptanz als MCP

Entscheidung zwischen Protokolloptionen

Passen Sie bei der Implementierung der agent-to-agent Kommunikation Ihre spezifischen Kommunikationsanforderungen an die entsprechenden Protokollfunktionen an. Verschiedene Interaktionsmuster erfordern unterschiedliche Protokollfunktionen. In der folgenden Tabelle werden gängige Kommunikationsmuster beschrieben und die für jedes Szenario am besten geeigneten Protokolloptionen empfohlen.

Muster	Beschreibung	Ideale Protokollwahl
Einfache Anfrage und Antwort	Einmalige Interaktionen zwischen Agenten	MCP mit zustandslosen Datenströmen
Zustandsreiche Dialoge	Laufende Gespräche mit Kontext	MCP mit Sitzungsverwaltung
Zusammenarbeit mit mehreren Agenten	Komplexe Interaktionen zwischen mehreren Agenten	MCP-Interagent oder AutoGen

Teambasierte Workflows	Hierarchische Agententeams mit definierten Rollen	MCP-Interagent, oder CrewAI AutoGen

Neben Kommunikationsmustern können verschiedene technische und organisatorische Faktoren Ihre Protokollauswahl beeinflussen. In der folgenden Tabelle sind die wichtigsten Überlegungen aufgeführt, anhand derer Sie beurteilen können, welches Protokoll Ihren spezifischen Implementierungsanforderungen am ehesten entspricht.

Überlegung	Beschreibung	Beispiel
Sicherheitsmodell	Anforderungen an Authentifizierung und Autorisierung	OAuth 2.0 im MCP
Bereitstellungsumgebung	Wo Agenten arbeiten und kommunizieren	Verteilter Computer oder einzelner Computer
Kompatibilität mit Ökosystemen	Integration mit bestehenden Agenten-Frameworks	LangChain oder Strands Agents
Anforderungen an Skalierbarkeit	Erwartetes Wachstum der Agenteninteraktionen	Streaming-Funktionen von MCP

Auswahl von Agentenprotokollen

Für die meisten Unternehmen, die Produktionsagentensysteme entwickeln, bietet das Model Context Protocol (MCP) die umfassendste und am besten unterstützte Kommunikationsgrundlage. agent-to-agent MCP profitiert von aktiven Entwicklungsbeiträgen der AWS Open-Source-Community.

Die Auswahl der richtigen behördlichen Protokolle ist wichtig für Unternehmen, die agentische KI effektiv implementieren möchten. Die Überlegungen unterscheiden sich je nach organisatorischem Kontext.

Überlegungen zur Auswahl von behördlichen Protokollen

Organizations sollten bei der Auswahl von Protokollen für agentische KI-Systeme die folgenden bewährten Methoden berücksichtigen:

- Priorisieren Sie offene Standards — Organizations sollten offene Protokolle wie das MCP einführen, um langfristige Interoperabilität und Erweiterbarkeit zu gewährleisten und das Risiko einer Lieferantenbindung zu verringern.
- Balance zwischen Geschwindigkeit und Flexibilität — Startups und Early Adopters können mit gut unterstützten proprietären Protokollen für eine schnelle Entwicklung beginnen, sollten jedoch einen Migrationspfad zu offenen Standards definieren, sobald die Systeme ausgereift sind.
- Implementieren Sie Abstraktionsebenen — Unternehmen sollten Protokollabstraktion implementieren, um die Migration zu vereinfachen, die Einführung von Hybridanwendungen zu ermöglichen und Integrationsstrategien zukunftssicher zu gestalten.
- Betonen Sie Sicherheit und Compliance — Organizations in regulierten Branchen sollten Protokolle mit robusten Authentifizierungs-, Verschlüsselungs- und Auditfunktionen wählen, um die Governance- und Compliance-Anforderungen zu erfüllen.
- Evaluieren Sie den Reifegrad des Ökosystems — Alle Organisationen sollten den Zustand, die Akzeptanz und die Unterstützung jedes einzelnen Protokolls bewerten, um die Nachhaltigkeit sicherzustellen und die technische Verschuldung zu minimieren.
- Sich an der Entwicklung von Standards beteiligen — Organizations sollten sich an Normungsgremien oder Open-Source-Communities beteiligen, um die Entwicklung von Protokollen mitzugestalten und bewährte Verfahren zu beeinflussen.
- Berücksichtigung der Datensouveränität — Regierung und regulierte Sektoren sollten sicherstellen, dass die Protokollauswahl den Anforderungen an die Datenresidenz und Souveränität in den Einsatzregionen entspricht.
- Nutzung verwalteter Dienste — Verwenden Sie nach Möglichkeit verwaltete oder serverlose Implementierungen behördlicher Protokolle, um die betriebliche Komplexität zu reduzieren und die Bereitstellung zu beschleunigen.

Implementierungsstrategie für behördliche Protokolle

Um behördliche Protokolle effektiv in Ihrem Unternehmen zu implementieren, sollten Sie die folgenden strategischen Schritte in Betracht ziehen:

1. Beginnen Sie mit der Angleichung der Standards — Verwenden Sie, wo immer möglich, etablierte offene Protokolle.
2. Erstellen Sie Abstraktionsebenen — Implementieren Sie Adapter zwischen Ihren Systemen und spezifischen Protokollen.

3. Tragen Sie zu offenen Standards bei — Nehmen Sie an Gemeinschaften zur Protokollentwicklung teil.
4. Überwachen Sie die Protokollentwicklung — Bleiben Sie über neue Standards und Updates auf dem Laufenden.
5. Testen Sie die Interoperabilität regelmäßig — Stellen Sie sicher, dass Ihre Implementierungen kompatibel bleiben.

Erste Schritte mit MCP

AWS unterstützt aktiv das Model Context Protocol (MCP) durch Beiträge zur Entwicklung und Implementierung des Protokolls. AWS arbeitet mit führenden Open-Source-Agenten-Frameworks zusammen, darunter, und LangGraph CrewAILlamaIndex, um die future der Agentenkommunikation auf dem Protokoll zu gestalten.

Gehen Sie wie folgt vor, um das MCP in Ihrer Agentenarchitektur zu implementieren:

1. Erkunden Sie MCP-Implementierungen in Frameworks wie dem SDK. Strands Agents
2. Lesen Sie die technische Dokumentation zum Model Context Protocol.
3. Lesen Sie Offene Protokolle für Agenten-Interoperabilität, Teil 1: Kommunikation zwischen Agenten auf MCP (AWS Blog), um mehr über die Interoperabilität von Agenten zu erfahren.
4. Treten Sie der MCP-Community bei, um die Entwicklung des Protokolls zu beeinflussen.

MCP bietet eine Kommunikationsebene, die es Agenten ermöglicht, mit externen Daten und Diensten zu interagieren, und kann auch verwendet werden, um Agenten die Interaktion mit anderen Agenten zu ermöglichen. Die Streamable HTTP-Transportimplementierung des Protokolls bietet Entwicklern umfassende Interaktionsmuster, ohne das Rad neu erfinden zu müssen. Diese Muster unterstützen sowohl statuslose request/response Datenflüsse als auch statusbehaftetes Sitzungsmanagement mit persistenter Funktion. IDs

Durch die Einführung offener Protokolle wie MCP können Sie Ihr Unternehmen in die Lage versetzen, Agentensysteme aufzubauen, die flexibel, interoperabel und anpassungsfähig bleiben, wenn sich die KI-Technologie weiterentwickelt. Informationen zur agent-to-tool Protokollimplementierung finden Sie weiter unten in diesem Handbuch unter Strategie zur Tool-Integration.

Erste Schritte mit A2A

Das Agent2Agent (A2A) -Protokoll ermöglicht die dezentrale Zusammenarbeit zwischen Agenten über eine gemeinsame semantische Ebene. Anstatt die gesamte Arbeit über einen zentralen Orchestrator weiterzuleiten, ermöglicht A2A den Agenten, sich gegenseitig zu entdecken, für ihre Fähigkeiten zu werben, Aufgaben auszuhandeln und den Kontext mithilfe eines einfachen JSON-basierten Protokolls gemeinsam zu nutzen. Jeder Agent veröffentlicht ein Funktionsmanifest.

Das folgende Beispiel zeigt ein vereinfachtes A2A-Fähigkeitsmanifest, das die unterstützten Aktionen, erforderlichen Eingaben und Betriebsmetadaten eines Agenten ankündigt, um die Erkennung und Aushandlung von Aufgaben zu ermöglichen:

```
{
  "can": ["summarize.text", "extract.keywords"],
  "needs": ["document.input"],
  "meta": { "version": "1.0.3", "latencyMs": 120 }
}
```

Dieses Modell ermöglicht einen dynamischen Kapazitätsabgleich, die Delegierung von Aufgaben während der Ausführung und die organisationsübergreifende Zusammenarbeit. Agenten können sich anhand von Aufgaben selbst organisieren, temporäre Arbeitsgruppen bilden und sich anpassen, wenn neue Funktionen in das System eingeführt oder verlassen werden.

A2A unterstützt Interaktionen, die von einfachen Anfragen ohne Status bis hin zu mehrstufigen Verhandlungssitzungen reichen, darunter:

- peer-to-peerDirektnachrichten für Zusammenarbeit mit niedriger Latenz
- Semantische Aufgabenaushandlung, bei der Agenten den am besten geeigneten Peer auswählen
- Auf Fähigkeiten basierende Entdeckung, die eine sich abzeichnende Arbeitsteilung ermöglicht
- Sitzungsverankerung für zustandsbehaftete Interaktionen in mehreren Schritten

Durch die Einführung offener, agenteneigener Protokolle wie A2A schaffen Unternehmen KI-Systeme, die modular und interoperabel sind und eine grenzüberschreitende Zusammenarbeit ermöglichen. A2A stellt sicher, dass die Agenten-Ökosysteme flexibel bleiben und sich weiterentwickeln können, wenn neue Agenten, Teams oder externe Systeme eingeführt werden, ohne dass starre Orchestrierungsebenen oder vorherige Kopplungen erforderlich sind.

Gehen Sie wie folgt vor, um das A2A-Protokoll in Ihrer Agentenarchitektur zu implementieren:

1. Lesen Sie die A2A-Protokollspezifikation — Lesen Sie die neueste Version der Agent2Agent (A2A) -Protokollspezifikation, um zu erfahren, wie Funktionen funktionieren, wie die Verhandlungsabläufe ablaufen und der Agent-Handshake funktionieren.
2. Erkunden Sie A2A-kompatible Laufzeiten — Evaluieren Sie Frameworks wie das Strands Agents SDK oder benutzerdefinierte Runtime-Layer, die Funktionsmanifeste und -aushandlungen im A2A-Stil unterstützen. peer-to-peer
3. Implementieren Sie ein Funktionsmanifest für Ihre Agenten — Definieren Sie die `meta` Felder und Felder der einzelnen Agenten, um die Erkennung `canneeds`, das Matchmaking und die Zusammenarbeit auf Absichtsebene zu ermöglichen.
4. Experimentieren Sie mit A2A-Verhandlungsmustern — nutzen Sie die Anfrage-Angebot-Annahme-Schleife, strukturierte Funktionsabfragen oder eine auf Klatsch und Tratsch basierende Erkennung, um zu verstehen, wie Agenten entscheiden, wer eine Aufgabe bearbeiten soll.
5. Testen Sie A2A in einer gemischten Infrastrukturumgebung — Kombinieren Sie A2A-Peer-Negotiation mit Event-Routing, das standardmäßig AWS über Amazon erfolgt, um hybride Koordinationsmuster EventBridge zu bewerten.
6. Treten Sie der A2A-Community bei — Nehmen Sie an der offenen Arbeitsgruppe teil, um über Erweiterungen, Sicherheitsempfehlungen und herstellerübergreifende Interoperabilitätsverbesserungen auf dem Laufenden zu bleiben und zur Entwicklung des Protokolls beizutragen.

Tools

KI-Agenten bieten Mehrwert, indem sie mit externen Tools und Datenquellen interagieren, um nützliche Aufgaben auszuführen APIs. Die richtige Strategie zur Tool-Integration wirkt sich direkt auf die Fähigkeiten, die Sicherheitslage und die langfristige Flexibilität Ihres Agenten aus.

Dieser Abschnitt hilft Ihnen, sich in der Tool-Integrationslandschaft zurechtzufinden, wobei der Schwerpunkt auf offenen Standards liegt, die Ihre Freiheit und Flexibilität maximieren. In diesem Abschnitt wird das Model Context Protocol (MCP) für die Toolintegration vorgestellt und Framework-spezifische Tools und spezialisierte Meta-Tools zur Verbesserung der Arbeitsabläufe von Agenten besprochen.

In diesem Abschnitt:

- Kategorien von Tools
- Protokollbasierte Tools
- Framework-native Tools
- Meta-Tools
- Strategie zur Integration von Tools
- Bewährte Sicherheitsmethoden für die Toolintegration

Kategorien von Tools

Building Agent Systems umfasst drei Hauptkategorien von Tools.

Auf Protokollen basierende Tools

Protokollbasierte Tools verwenden standardisierte Protokolle für die Kommunikation: agent-to-tool

- MCP-Tools — Offene Standardtools, die in allen Frameworks funktionieren und sowohl lokale als auch Remote-Ausführungsoptionen bieten.
- OpenAIFunktionsaufruf — Proprietäre Tools, die spezifisch für OpenAI Modelle sind.
- AnthropicTools — Eine Variante des OpenAI Funktionsaufrufs für proprietäre Tools, die spezifisch für Anthropic Claude-Modelle sind.

Framework-native Tools

Framework-native Tools sind direkt in spezifische Agenten-Frameworks integriert:

- Strands Agents Tools — Leichte quick-to-implement Tools, die spezifisch für das Strands Agents Framework sind.
- LangChainPythonauf Tools basierende Tools, die eng in das LangChain Ökosystem integriert sind.
- LlamaIndexTools — Tools, die für den Datenabruf und die interne LlamaIndex Verarbeitung optimiert sind.

Meta-Tools

Meta-Tools verbessern die Arbeitsabläufe der Agenten, ohne direkt externe Maßnahmen ergreifen zu müssen:

- Workflow-Tools — Verwalten Sie den Ausführungsablauf der Agenten, die Verzweigungslogik und die Statusverwaltung.
- Tools für Agentendiagramme — Koordinieren Sie mehrere Agenten in komplexen Workflows.
- Speichertools — Sorgen für die persistente Speicherung und den dauerhaften Abruf von Informationen über Agentensitzungen hinweg.
- Reflexionstools — Ermöglichen es Agenten, ihre eigene Leistung zu analysieren und zu verbessern.

Auf Protokollen basierende Tools

Wenn es um protokollbasierte Tools geht, bietet das Model Context Protocol (MCP) die umfassendste und flexibelste Grundlage für die Toolintegration. Wie im AWS Open-Source-Blogbeitrag zur Interoperabilität von Agenten dargelegt, AWS hat sich das Unternehmen MCP als strategisches Protokoll zu eigen gemacht und aktiv zu seiner Entwicklung beigetragen.

In der folgenden Tabelle werden die Optionen für die Bereitstellung des MCP-Tools beschrieben.

Bereitstellungsmodell	Beschreibung	Ideal für	Implementierung

Lokales Studio	Tools werden im gleichen Prozess wie der Agent ausgeführt	Entwicklung, Testen und einfache Tools	Schnelle Implementierung ohne Netzwerk-Overhead
Auf lokalen serverges tützten Ereignissen (SSE)	Tools werden lokal ausgeführt, kommunizieren aber über HTTP	Komplexere lokale Tools mit getrennten Aufgabenbereichen	Bessere Isolierung, aber immer noch geringe Latenz
Remote-HTTP-Streamingfähig	Tools werden auf Remote-Servern ausgeführt	Produktionsumgebungen und gemeinsam genutzte Tools	Skalierbar und zentral verwaltet

Für die Erstellung von SDKs MCP-Tools stehen die offiziellen MCP zur Verfügung:

- PythonSDK — Umfassende Implementierung mit vollständiger Protokollunterstützung
- TypeScriptSDK —JavaScript//TypeScriptImplementierung für Webanwendungen
- JavaSDK — Java-Implementierung für Unternehmensanwendungen

Diese SDKs bieten die Bausteine für die Erstellung MCP-kompatibler Tools in Ihrer bevorzugten Sprache mit konsistenten Implementierungen der Protokollspezifikation.

Darüber hinaus AWS hat MCP im SDK implementiert. Strands Agents Das Strands Agents SDK bietet eine einfache Möglichkeit, MCP-kompatible Tools zu erstellen und zu verwenden. Eine umfassende Dokumentation ist im Repository verfügbar. Strands Agents GitHub Für einfachere Anwendungsfälle oder wenn Sie außerhalb des Strands Agents Frameworks arbeiten, SDKs bietet das offizielle MCP direkte Implementierungen des Protokolls in mehreren Sprachen an.

Sicherheitsfunktionen von MCP-Tools

Zu den Sicherheitsfunktionen der MCP-Tools gehören:

- OAuth 2.0/2.1-Authentifizierung — Authentifizierung nach Industriestandard
- Umfang der Berechtigungen — Präzise Zugriffskontrolle für Tools
- Erkennung der Funktionen von Tools — Dynamische Erkennung verfügbarer Tools
- Strukturierte Fehlerbehandlung — Konsistente Fehlermuster

Erste Schritte mit MCP-Tools

Gehen Sie wie folgt vor, um MCP für die Toolintegration zu implementieren:

1. Erkunden Sie das Strands AgentsSDK für eine produktionsreife MCP-Implementierung.
2. Lesen Sie die technische Dokumentation zu MCP, um die Kernkonzepte zu verstehen.
3. Verwenden Sie die in diesem AWS Open-Source-Blogbeitrag beschriebenen praktischen Beispiele.
4. Beginnen Sie mit einfachen lokalen Tools, bevor Sie zu Remote-Tools übergehen.
5. Treten Sie der MCP-Community bei, um die Entwicklung des Protokolls zu beeinflussen.

Erkunden Sie Gateway AgentCore

Amazon Bedrock AgentCore Gateway bietet Entwicklern eine einfache und sichere Möglichkeit, MCP-Tools und andere Zielendpunkte in großem Umfang zu entwickeln, bereitzustellen, zu entdecken und eine Verbindung zu ihnen herzustellen. Mit AgentCore Gateway können Entwickler AWS Lambda Funktionen und bestehende Services in MCP-kompatible Tools umwandeln APIs. Anschließend können sie diese Tools mit nur wenigen Codezeilen Agenten über AgentCore Gateway-Endpunkte zur Verfügung stellen. AgentCore Gateway unterstützt OpenAPISmithy, und Lambda als Eingabetypen und ist die einzige Lösung, die sowohl eine umfassende Eingangs- als auch Ausgangsauthentifizierung in einem vollständig verwalteten Service bietet.

Framework-native Tools

Obwohl das Model Context Protocol (MCP) die flexibelste Grundlage bietet, bieten Framework-native Tools Vorteile für bestimmte Anwendungsfälle.

Das Strands AgentsSDK bietet Python basierte Tools, die sich durch ihr leichtes Design auszeichnen und nur minimalen Aufwand für einfache Operationen erfordern. Sie ermöglichen eine schnelle Implementierung und ermöglichen es Entwicklern, Tools mit nur wenigen Codezeilen zu erstellen. Darüber hinaus sind sie eng integriert, sodass sie nahtlos in das Strands Agents Framework integriert werden können.

Das folgende Beispiel zeigt, wie Sie mit Hilfe von ein einfaches Wetter-Tool erstellenStrands Agents. Entwickler können Python Funktionen schnell und mit minimalem Codeaufwand in Tools umwandeln, auf die Agenten zugreifen können, und automatisch die entsprechende Dokumentation aus dem Docstring der Funktion generieren.

```
#Example of a simple Strands native tool

@tool

def weather(location: str) -> str:

"""Get the current weather for a location""" #

Implementation here

return f"The weather in {location} is sunny."
```

Für schnelles Prototyping oder einfache Anwendungsfälle können Framework-native Tools die Entwicklung beschleunigen. Für Produktionssysteme bieten MCP-Tools jedoch eine bessere Interoperabilität und future Flexibilität als Framework-native Tools.

Die folgende Tabelle bietet einen Überblick über andere Framework-spezifische Tools.

Framework	Art des Werkzeugs	Vorteile	Überlegungen
AutoGen	Funktionsdefinitionen	Starke Unterstützung für mehrere Agenten	MicrosoftÖkosystem
LangChain	PythonKlassen	Großes Ökosystem vorgefertigter Tools	Bindung an ein Framework
LlamaIndex	Python-Funktionen	Optimiert für Datenoperationen	Beschränkt auf LlamaIndex

Meta-Tools

Meta-Tools interagieren nicht direkt mit externen Systemen. Stattdessen verbessern sie die Fähigkeiten der Agenten, indem sie Agentenmuster implementieren. In diesem Abschnitt werden Workflows, Agentendiagramme und Speicher-Metatools behandelt.

Workflow-Metatools

Workflow-Metatools verwalten den Ablauf der Agentenausführung:

- Statusverwaltung — Behalten Sie den Kontext über mehrere Agenteninteraktionen hinweg bei

- Verzweigungslogik — Aktivieren Sie bedingte Ausführungspfade
- Wiederholungsmechanismen — Behandeln Sie Fehler mit ausgeklügelten Wiederholungsstrategien

Zu den Beispiel-Frameworks mit Workflow-Meta-Tools gehören auch LangGraphWorkflow-Funktionen. Strands Agents

Metatools für Agentengraphen

Die Metatools für Agentengraphen koordinieren die Zusammenarbeit mehrerer Agenten:

- Delegierung von Aufgaben — Weisen Sie spezialisierten Agenten Unteraufgaben zu
- Ergebnisaggregation — Kombinieren Sie die Ergebnisse mehrerer Agenten
- Konfliktlösung — Beilegung von Meinungsverschiedenheiten zwischen Agenten

Frameworks sind auf die Graphkoordination von Agenten CrewAIspezialisiert AutoGenund haben sich darauf spezialisiert.

Speicher-Metatools

Speicher-Metatools ermöglichen persistentes Speichern und Abrufen:

- Konversationsverlauf — Behalten Sie den Kontext zwischen den Sitzungen bei
- Wissensdatenbanken — Speichern und Abrufen domänenspezifischer Informationen
- Vektorspeicher — Ermöglichen semantische Suchfunktionen

Das Ressourcensystem von MCP bietet eine standardisierte Methode zur Implementierung von Speicher-Metatools, die über verschiedene Agenten-Frameworks hinweg funktionieren.

Strategie zur Integration von Tools

Ihre Wahl der Tool-Integrationsstrategie wirkt sich direkt darauf aus, was Ihre Agenten erreichen können und wie einfach sich Ihr System weiterentwickeln kann. Priorisieren Sie offene Protokolle wie das Model Context Protocol (MCP) und setzen Sie gleichzeitig Framework-native Tools und Meta-Tools strategisch ein. Auf diese Weise können Sie ein Tool-Ökosystem aufbauen, das auch im Zuge der Weiterentwicklung der KI-Technologie flexibel und leistungsstark bleibt.

Der folgende strategische Ansatz zur Toolintegration maximiert die Flexibilität und erfüllt gleichzeitig die unmittelbaren Bedürfnisse Ihres Unternehmens:

1. Nutzen Sie MCP als Grundlage — MCP bietet eine standardisierte Möglichkeit, Agenten mit Tools mit starken Sicherheitsfunktionen zu verbinden. Beginnen Sie mit MCP als primärem Tool-Protokoll für:
   - Strategische Tools, die in mehreren Agentenimplementierungen eingesetzt werden.
   - Sicherheitsempfindliche Tools, die eine zuverlässige Authentifizierung und Autorisierung erfordern.
   - Tools, die in Produktionsumgebungen remote ausgeführt werden müssen.
2. Verwenden Sie gegebenenfalls Framework-native Tools — Ziehen Sie Framework-native Tools in Betracht für:
   - Schnelles Prototyping während der ersten Entwicklung.
   - Einfache, unkritische Tools mit minimalen Sicherheitsanforderungen.
   - Framework-spezifische Funktionen, die einzigartige Funktionen nutzen.
3. Implementieren Sie Meta-Tools für komplexe Workflows — Fügen Sie Meta-Tools hinzu, um Ihre Agentenarchitektur zu verbessern:
   - Beginnen Sie einfach mit grundlegenden Workflow-Mustern.
   - Erhöhen Sie die Komplexität, wenn Ihre Anwendungsfälle immer ausgereifter werden.
   - Standardisieren Sie die Schnittstellen zwischen Agenten und Meta-Tools.
4. Für die Entwicklung planen — mit Blick auf future Flexibilität bauen:
   - Dokumentieren Sie die Benutzeroberflächen der Tools unabhängig von den Implementierungen.
   - Erstellen Sie Abstraktionsebenen zwischen Agenten und Tools.
   - Richten Sie Migrationspfade von proprietären zu offenen Protokollen ein.

Bewährte Sicherheitsverfahren für die Integration von Tools

Die Tool-Integration wirkt sich direkt auf Ihre Sicherheitslage aus. In diesem Abschnitt werden bewährte Methoden beschrieben, die Sie für Ihr Unternehmen berücksichtigen sollten.

Authentifizierung und Autorisierung

Nutzen Sie die folgenden robusten Zugriffskontrollen:

- Verwenden Sie OAuth 2.0/2.1 — Implementieren Sie die branchenübliche Authentifizierung für Remote-Tools.
- Implementierung der geringsten Rechte — Gewähren Sie Tools nur die Berechtigungen, die sie benötigen.
- Zugangsdaten wechseln — Aktualisieren Sie regelmäßig API-Schlüssel und Zugriffstoken.

Datenschutz

Ergreifen Sie zum Schutz Ihrer Daten die folgenden Maßnahmen:

- Eingaben und Ausgaben validieren — Implementieren Sie die Schemavalidierung für alle Werkzeuginteraktionen.
- Vertrauliche Daten verschlüsseln — Verwenden Sie TLS für die gesamte Remote-Tool-Kommunikation.
- Implementieren Sie Datenminimierung — Geben Sie nur die erforderlichen Informationen an die Tools weiter.

Überwachung und Prüfung

Sorgen Sie mit den folgenden Mechanismen für Transparenz und Kontrolle:

- Protokollieren Sie alle Tool-Aufrufe — Pflegen Sie umfassende Prüfprotokolle.
- Auf Anomalien achten — Erkennen Sie ungewöhnliche Nutzungsmuster von Tools.
- Implementieren Sie eine Ratenbegrenzung — Beugen Sie Missbrauch durch übermäßige Tool-Aufrufe vor.

Das Sicherheitsmodell Model Context Protocol (MCP) geht umfassend auf diese Bedenken ein. Weitere Informationen finden Sie in der MCP-Dokumentation unter Überlegungen zur Sicherheit.

Schlussfolgerung

Die Landschaft der agentischen KI entwickelt sich weiterhin rasant und bietet Unternehmen leistungsstarke neue Möglichkeiten zum Aufbau intelligenter, autonomer Systeme. In diesem Leitfaden wurden drei wesentliche Komponenten für eine erfolgreiche Implementierung untersucht: Frameworks, die die Grundlage bilden, Plattformen, die die Umgebung bereitstellen, Protokolle, die die Kommunikation ermöglichen, und Tools, die die Funktionen erweitern.

Mit zunehmender Reife der Frameworks können Sie mit einer erhöhten Interoperabilität, einer Standardisierung von Protokollen wie dem Model Context Protocol (MCP) und ausgefeilteren Orchestrierungsfunktionen für autonome Agenten rechnen. Organizations, die sich heute mit diesen Frameworks auskennen, sind gut positioniert, um zunehmend autonome, intelligente Agenten zu entwickeln, die einen erheblichen Geschäftswert bieten.

Plattformen bieten die Ausführungs-, Verwaltungs- und Lebenszyklusumgebung, in der agentische Systeme betrieben werden. Sie kümmern sich um Themen wie Identität, Sicherheitsgrenzen, Beobachtbarkeit, Speicherverwaltung, Sitzungsbindung und sichere Interaktion mit Tools und Daten. In AWS Umgebungen ermöglichen Plattformen wie Managed Agent Runtimes und Orchestration Services Unternehmen, autonome Agenten und Agentensysteme in großem Umfang einzusetzen, zu überwachen, weiterzuentwickeln und zu steuern. Plattformen verbinden grundlegende Frameworks mit realen betrieblichen Anforderungen.

Die Wahl der Agentenprotokolle stellt eine strategische Entscheidung dar, bei der unmittelbare Entwicklungsanforderungen mit langfristiger Flexibilität und Interoperabilität in Einklang gebracht werden. Durch die Priorisierung offener Protokolle und die Schaffung geeigneter Abstraktionsebenen können Unternehmen Agentensysteme aufbauen, die an sich entwickelnde Technologien anpassbar bleiben und gleichzeitig den aktuellen Geschäftsanforderungen entsprechen.

Für die meisten Unternehmen stellt MCP aufgrund seines offenen Standards, seines wachsenden Ökosystems, der Unterstützung von agent-to-agent Kommunikationsmustern und der Fähigkeit zur Toolintegration eine solide Grundlage dar. AWS hat MCP und Agent2Agent (A2A) als strategische Protokolle eingeführt und aktiv zu deren Entwicklung und Implementierung in Diensten wie dem SDK beigetragen. Strands Agents Durch die Verwendung von MCP oder A2A zusammen mit geeigneten Framework-nativen Tools und Meta-Tools können Sie Agentensysteme erstellen, die sofort einen Mehrwert bieten und gleichzeitig an future Innovationen anpassbar sind.

Ressourcen

Nutzen Sie die folgenden AWS und andere Ressourcen im Zusammenhang mit der Entwicklung autonomer Agenten.

AWS Blogs

- Amazon Bedrock AgentCore Memory: Aufbau kontextsensitiver Agenten
- Bewährte Methoden für die Erstellung robuster generativer KI-Anwendungen mit Amazon Bedrock Agents — Teil 1
- Bewährte Methoden für die Erstellung robuster generativer KI-Anwendungen mit Amazon Bedrock Agents — Teil 2
- Erstellen Sie leistungsstarke RAG-Pipelines mit Amazon LlamaIndex Bedrock
- Bauen Sie mit Amazon Bedrock AgentCore Observability vertrauenswürdige KI-Agenten auf
- Evaluieren Sie die Antworten von RAG mit Amazon Bedrock LlamaIndex und RAGAS
- Wir stellen vor: den Amazon Bedrock AgentCore Code Interpreter
- Wir stellen vor: Amazon Bedrock AgentCore Gateway: Transformation der Entwicklung von KI-Agententools für Unternehmen
- Wir stellen vor: Amazon Bedrock AgentCore Identity: Sicherung agentischer KI im großen Maßstab
- Wir stellen vor: Strands Agents ein Open-Source-SDK für KI-Agenten
- Offene Protokolle für Agenten-Interoperabilität Teil 1: Agentenübergreifende Kommunikation auf MCP
- Starten und skalieren Sie Ihre Agenten und Tools sicher auf Amazon Bedrock Runtime AgentCore
- AWS Transform für.NET, den ersten agentischen KI-Service für die Modernisierung von.NET-Anwendungen im großen Maßstab
- AWS Wöchentliche Zusammenfassung: Strands Agents

AWS Präskriptive Leitlinien

- Operationalisierung der Agenten-KI am AWS
- Grundlagen der Agenten-KI auf AWS
- Agentengestützte KI-Muster und Workflows auf AWS

- Aufbau serverloser Architekturen für agentische KI auf AWS
- Aufbau von Mehrmandantenarchitekturen für agentische KI auf AWS
- Sicherheit für Agenten-KI auf AWS
- Optionen und Architekturen für Augmented Generation abrufen auf AWS

AWS Ressourcen

- Dokumentation zu Amazon Bedrock
- Dokumentation zu Amazon Bedrock AgentCore
- Amazon Bedrock AgentCore Starter Toolkit (Repository) GitHub
- Amazon Nova-Dokumentation
- AWS MCP-Server (GitHubRepository)

Sonstige Ressourcen

- AutoGenDokumentation () Microsoft
- Wirksame Agenten aufbauen (Anthropic)
- CrewAI GitHubEndlager
- LangChain-Dokumentation
- LangGraphPlattform
- LlamaIndex-Dokumentation
- Dokumentation zum Model Context Protocol
- Strands Agents-Dokumentation
- Strands AgentsÜberblick über die Tools
- Strands AgentsSchnellstart-Anleitung

Dokumentverlauf

In der folgenden Tabelle werden wichtige Änderungen in diesem Leitfaden beschrieben. Um Benachrichtigungen über zukünftige Aktualisierungen zu erhalten, können Sie einen RSS-Feed abonnieren.

Änderung	Beschreibung	Datum
Neuer Abschnitt	Abschnitt „Plattformen“ hinzugefügt	16. Januar 2026
Erste Veröffentlichung	—	14. Juli 2025

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.