Framework, piattaforme, protocolli e strumenti di intelligenza artificiale agentica su AWS

AWS Linee guida prescrittive

AWS Linee guida prescrittive: Framework, piattaforme, protocolli e strumenti di intelligenza artificiale agentica su AWS

Table of Contents

Framework, piattaforme, protocolli e strumenti di intelligenza artificiale agentica su AWS

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

Gennaio 2026 (storia del documento)

L'intelligenza artificiale agentica è un potente paradigma all'intersezione tra intelligenza artificiale, sistemi distribuiti e ingegneria del software. È una classe di sistemi intelligenti composti da agenti software autonomi e asincroni che utilizzano modelli di intelligenza artificiale e si integrano con strumenti e risorse. Gli agenti sono agenti, possono percepire il contesto, ragionare sugli obiettivi, prendere decisioni e intraprendere azioni mirate per conto degli utenti o dei sistemi. Questi agenti operano in modo indipendente, spesso collaborativo, all'interno di ambienti distribuiti e sono progettati per perseguire obiettivi delegati con intelligenza, memoria e intenti integrati.

Inoltre AWS, le organizzazioni possono sfruttare l'intelligenza artificiale agentica per automatizzare flussi di lavoro complessi, migliorare i processi decisionali e creare sistemi più reattivi. Questa guida fornisce informazioni sui componenti chiave necessari per creare soluzioni di intelligenza artificiale agentica efficaci:

- Frameworks descrive gli attuali framework di intelligenza artificiale agentica, comprese le revisioni dei relativi vantaggi e casi d'uso. Scopri come questi framework riducono il carico di lavoro indifferenziato tra modelli, protocolli e strumenti. Comprendi i criteri di selezione chiave per scegliere il framework giusto per le tue esigenze.
- Platforms offre una panoramica delle piattaforme di intelligenza artificiale agentica (agente gestito, orchestrazione open source e ibrida) e considerazioni per la selezione o la progettazione.
- Protocols esplora i protocolli di comunicazione standardizzati essenziali per le interazioni tra agenti. Agent-to-agentstanno emergendo protocolli, come gli open source Model Context Protocol (MCP) e Agent2Agent (A2A), insieme ad altre implementazioni proprietarie. Scopri come i protocolli comuni consentono a protocolli diversi di interagire senza problemi.
- Tools fornisce informazioni su strumenti basati su protocolli (come MCP), strumenti nativi del framework e meta-strumenti. Le organizzazioni possono creare un toolkit che si integra con i sistemi chiave dei loro flussi di lavoro, abilitando flussi di lavoro agentici sia per utenti finali che basati su server.

Destinatari principali

Questa guida è rivolta ad architetti, sviluppatori e leader tecnologici che desiderano sfruttare la potenza degli agenti software basati sull'intelligenza artificiale all'interno delle moderne applicazioni native del cloud.

Obiettivi

Questa guida ti consente di:

- Confronta diversi framework di intelligenza artificiale agentica per selezionare quello più appropriato per il tuo caso d'uso.
- Scopri le piattaforme di intelligenza artificiale agentica che forniscono funzionalità per trasformare i singoli agenti in sistemi coordinati e adattivi.
- Comprendi i vantaggi dei protocolli aperti per la creazione di architetture di intelligenza artificiale agentica sostenibili.
- Crea una strategia di integrazione degli strumenti appropriata durante la creazione di sistemi con agenti.

Informazioni su questa serie di contenuti

Questa guida fa parte di una serie sull'intelligenza artificiale agentica su. AWS Per ulteriori informazioni e per visualizzare le altre guide di questa serie, consulta Agentic AI sul sito Web Prescriptive Guidance. AWS

Framework

Foundations of agentic AI on AWS esamina i modelli e i flussi di lavoro fondamentali che consentono un comportamento autonomo e mirato. Alla base dell'implementazione di questi modelli c'è la scelta del framework. Un framework è la base software del codice già scritto che fornisce un ambiente strutturato e funzionalità comuni per la creazione e la gestione degli strumenti e delle funzionalità di orchestrazione necessari per creare agenti di intelligenza artificiale autonomi pronti per la produzione.

I framework di intelligenza artificiale agentica offrono diverse funzionalità essenziali che trasformano le interazioni grezze con modelli di linguaggio di grandi dimensioni (LLM) in sistemi coordinati e intelligenti in grado di ragionare, collaborare e agire:

- L'orchestrazione degli agenti coordina il flusso di informazioni e il processo decisionale tra uno o più agenti per raggiungere obiettivi complessi senza l'intervento umano.
- L'integrazione degli strumenti consente agli agenti di interagire con sistemi esterni, API e fonti di dati per estendere le proprie funzionalità oltre l'elaborazione del linguaggio. Per ulteriori informazioni, vedere Panoramica degli strumenti nella Strands Agents documentazione.
- La gestione della memoria fornisce uno stato persistente o basato sulla sessione per mantenere il contesto tra le interazioni, essenziale per attività di lunga durata o adattive. I framework più avanzati incorporano memoria a lungo termine per archiviare riepiloghi e preferenze degli utenti, consentendo esperienze agentiche personalizzate e contestualmente consapevoli. Per ulteriori informazioni, consulta How to think about agent framework sul blog. LangChain
- La definizione del flusso di lavoro supporta modelli strutturati come catene, routing, parallelizzazione e cicli di riflessione che consentono un ragionamento autonomo sofisticato.
- L'implementazione e il monitoraggio facilitano la transizione dallo sviluppo alla produzione con l'osservabilità per i sistemi autonomi. Per ulteriori informazioni, consulta l'annuncio sulla disponibilità AgentCore generale di Amazon Bedrock.

Queste funzionalità sono implementate con approcci ed enfasi diversi in tutto il panorama del framework, ognuna delle quali offre vantaggi distinti per diversi casi d'uso di agenti autonomi e contesti organizzativi.

Questa sezione descrive e confronta i principali framework per la creazione di soluzioni di intelligenza artificiale agentiche, con particolare attenzione ai loro punti di forza, limiti e casi d'uso ideali per il funzionamento autonomo:

- Agenti Strands
- LangChain e LangGraph
- Crew AI
- AutoGen
- ???
- Confronto tra framework di intelligenza artificiale agentica

> **Note**
>
> Questa sezione tratta i framework che supportano specificamente l'agenzia dell'intelligenza artificiale e non copre le interfacce frontend o l'IA generativa senza agenzia.

Strands Agents

Strands Agentsè un SDK open source inizialmente rilasciato da AWS, come descritto nel blog Open Source.AWS Strands Agentsè progettato per creare agenti di intelligenza artificiale autonomi con un approccio basato sul modello e fornisce un framework flessibile ed estensibile progettato per funzionare senza interruzioni Servizi AWS pur rimanendo aperto all'integrazione con componenti di terze parti. Strands Agents è ideale per creare soluzioni completamente autonome.

Caratteristiche principali di Strands Agents

Strands Agentsinclude le seguenti funzionalità chiave:

- Model-first design: costruito attorno al concetto che il modello di base è il fulcro dell'intelligenza degli agenti, che consente un ragionamento autonomo sofisticato. Per ulteriori informazioni, consulta Agent Loop nella Strands Agents documentazione.
- Multi-agent modelli di collaborazione: modelli di Built-in coordinamento come Swarm, Graph e Workflow che consentono la collaborazione e la governance scalabili su reti di agenti distribuite. Per ulteriori informazioni, consulta Multi-agent Patterns nella documentazione di Strands Agents.
- Integrazione MCP: supporto nativo per il Model Context Protocol (MCP), che consente la fornitura di un contesto standardizzato ai LLM per un funzionamento autonomo e coerente.

- Servizio AWS integrazione: connessione perfetta ad Amazon Bedrock e altro Servizi AWS per AWS Step Functions flussi di lavoro autonomi completi. AWS Lambda Per ulteriori informazioni, consulta AWS Weekly Roundup (blog).AWS
- Selezione del modello Foundation: supporta vari modelli di base tra cui Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) su Amazon Bedrock e altri per ottimizzare diverse capacità di ragionamento autonomo. Per ulteriori informazioni, consulta Amazon Bedrock nella Strands Agents documentazione.
- Integrazione dell'API LLM: integrazione flessibile con diverse interfacce di servizio LLM tra cui Amazon Bedrock, OpenAI e altre per l'implementazione in produzione. Per ulteriori informazioni, consulta Amazon Bedrock Basic Usage nella Strands Agents documentazione.
- Funzionalità multimodali: supporto per più modalità tra cui l'elaborazione di testo, voce e immagini per interazioni complete con agenti autonomi. Per ulteriori informazioni, consulta Amazon Bedrock Multimodal Support nella Strands Agents documentazione.
- Ecosistema di strumenti: ricco set di strumenti per Servizio AWS l'interazione, con estensibilità per strumenti personalizzati che ampliano le capacità autonome. Per ulteriori informazioni, vedere Panoramica degli strumenti nella Strands Agents documentazione.

Quando usare Strands Agents

Strands Agentsè particolarmente adatto per scenari con agenti autonomi, tra cui:

- Organizzazioni che si basano su AWS un'infrastruttura che desiderano un'integrazione nativa Servizi AWS per flussi di lavoro autonomi
- Team che richiedono funzionalità di sicurezza, scalabilità e conformità di livello aziendale per sistemi autonomi di produzione
- Progetti che richiedono flessibilità nella selezione dei modelli tra diversi fornitori per attività autonome specializzate
- Casi d'uso che richiedono una stretta integrazione con i AWS flussi di lavoro e le risorse esistenti per processi autonomi end-to-end

Approccio di implementazione per Strands Agents

Strands Agentsfornisce un approccio di implementazione semplice per gli stakeholder aziendali, come indicato nella Guida rapida per Python e nella Guida rapida per. Typescript Il framework consente alle organizzazioni di:

- Seleziona modelli base come Amazon Nova (Premier, Pro, Lite o Micro) su Amazon Bedrock in base a requisiti aziendali specifici.
- Definisci strumenti personalizzati che si connettono ai sistemi e alle fonti di dati aziendali.
- Elabora più modalità tra cui testo, immagini e voce.
- Implementa agenti in grado di rispondere autonomamente alle domande aziendali ed eseguire attività.

Questo approccio di implementazione consente ai team aziendali di sviluppare e implementare rapidamente agenti autonomi senza una profonda esperienza tecnica nello sviluppo di modelli di intelligenza artificiale.

Real-world esempio di Strands Agents

AWS Transform for .NET lo utilizza per Strands Agents potenziare le proprie funzionalità di modernizzazione delle applicazioni, come descritto in AWS Transform per.NET, il primo servizio di intelligenza artificiale agentica per modernizzare le applicazioni.NET su larga scala (Blog).AWS Questo servizio di produzione impiega più agenti autonomi specializzati. Gli agenti collaborano per analizzare le applicazioni.NET legacy, pianificare strategie di modernizzazione ed eseguire trasformazioni del codice verso architetture native del cloud senza l'intervento umano. AWS Transform for .NET dimostra la disponibilità alla produzione dei sistemi autonomi aziendali. Strands Agents

LangChain e LangGraph

LangChainè uno dei framework più affermati nell'ecosistema di intelligenza artificiale agentica. LangGraphestende le sue funzionalità per supportare flussi di lavoro complessi e basati sullo stato degli agenti, come descritto nel blog. LangChain Insieme, forniscono una soluzione completa per la creazione di sofisticati agenti di intelligenza artificiale autonomi con ricche capacità di orchestrazione per operazioni indipendenti.

Caratteristiche principali di e LangChain LangGraph

LangChaine LangGraph includono le seguenti funzionalità chiave:

- Ecosistema di componenti: ampia libreria di componenti predefiniti per varie funzionalità di agenti autonomi, che consente lo sviluppo rapido di agenti specializzati. Per ulteriori informazioni, consulta Quickstart nella LangChain documentazione.

- Selezione del modello Foundation: supporto per diversi modelli di base, tra cui Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) su Amazon Bedrock e altri per diverse funzionalità di ragionamento. Per ulteriori informazioni, consulta Ingressi e uscite nella documentazione. LangChain
- Integrazione dell'API LLM: interfacce standardizzate per più fornitori di servizi LLM (Large Language Model) tra cui Amazon Bedrock e altri per una OpenAI distribuzione flessibile. Per ulteriori informazioni, consulta gli LLM nella documentazione. LangChain
- Elaborazione multimodale: Built-in supporto per l'elaborazione di testo, immagini e audio per consentire ricche interazioni multimodali tra agenti autonomi. Per ulteriori informazioni, vedete Multimodalità nella documentazione. LangChain
- Graph-based flussi di lavoro: LangGraph consente di definire comportamenti complessi di agenti autonomi come macchine a stati, supportando una logica decisionale sofisticata. Per ulteriori informazioni, consulta l'annuncio di LangGraphPlatform GA.
- Astrazioni di memoria: opzioni multiple per la gestione della memoria a breve e lungo termine, essenziali per gli agenti autonomi che mantengono il contesto nel tempo. Per ulteriori informazioni, consulta Come aggiungere memoria ai chatbot nella documentazione. LangChain
- Integrazione con strumenti: ricco ecosistema di integrazioni di strumenti tra vari servizi e API, che estende le capacità degli agenti autonomi. Per ulteriori informazioni, consulta Tools nella documentazione. LangChain
- LangGraph piattaforma: soluzione gestita di implementazione e monitoraggio per ambienti di produzione, che supporta agenti autonomi a lunga durata. Per ulteriori informazioni, consulta l'annuncio di LangGraphPlatform GA.

Quando usare LangChain e LangGraph

LangChaine LangGraph sono particolarmente adatti per scenari con agenti autonomi, tra cui:

- Flussi di lavoro complessi di ragionamento in più fasi che richiedono un'orchestrazione sofisticata per un processo decisionale autonomo
- Progetti che richiedono l'accesso a un ampio ecosistema di componenti e integrazioni predefiniti per diverse funzionalità autonome
- Team con infrastruttura ed esperienza di machine Python learning (ML) esistenti che desiderano creare sistemi autonomi

- Casi d'uso che richiedono una gestione dello stato complessa in sessioni di agenti autonomi di lunga durata

Approccio di implementazione per LangChain e LangGraph

LangChaine LangGraph forniscono un approccio di implementazione strutturato per gli stakeholder aziendali, come dettagliato nella LangGraphdocumentazione. Il framework consente alle organizzazioni di:

- Definisci grafici sofisticati del flusso di lavoro che rappresentano i processi aziendali.
- Crea modelli di ragionamento in più fasi con punti decisionali e logica condizionale.
- Integra funzionalità di elaborazione multimodali per gestire diversi tipi di dati.
- Implementa il controllo di qualità attraverso meccanismi di revisione e convalida integrati.

Questo approccio basato su grafici consente ai team aziendali di modellare processi decisionali complessi come flussi di lavoro autonomi. I team hanno una chiara visibilità su ogni fase del processo di ragionamento e la capacità di verificare i percorsi decisionali.

Real-world esempio di e LangChain LangGraph

Vodafoneha implementato agenti autonomi utilizzando LangChain (andLangGraph) per migliorare i flussi di lavoro di ingegneria dei dati e operativi, come dettagliato nel case study LangChain Enterprise. Hanno creato assistenti di intelligenza artificiale interni che monitorano autonomamente le metriche delle prestazioni, recuperano informazioni dai sistemi di documentazione e presentano informazioni utili, il tutto attraverso interazioni in linguaggio naturale.

L'Vodafoneimplementazione utilizza caricatori di documenti LangChain modulari, integrazione vettoriale e supporto per più LLM (3, e) per prototipare e confrontare rapidamente queste pipeline. OpenAI LLaMA Gemini Sono stati quindi utilizzati per LangGraph strutturare l'orchestrazione multiagente implementando agenti secondari modulari. Questi agenti eseguono attività di raccolta, elaborazione, riepilogo e ragionamento. LangGraphhanno integrato questi agenti tramite API nei loro sistemi cloud.

CrewAI

CrewAIè un framework open source incentrato specificamente sull'orchestrazione autonoma multiagente, disponibile su. GitHub Fornisce un approccio strutturato alla creazione di team di agenti

autonomi specializzati che collaborano per risolvere compiti complessi senza l'intervento umano. CrewAIenfatizza il coordinamento basato sui ruoli e la delega dei compiti.

Caratteristiche principali di CrewAI

CrewAIoffre le seguenti funzionalità chiave:

- Role-based progettazione degli agenti: gli agenti autonomi sono definiti con ruoli, obiettivi e precedenti specifici per consentire competenze specializzate. Per ulteriori informazioni, consulta Crafting Effective Agents nella CrewAI documentazione.
- Delega delle attività: Built-in meccanismi per l'assegnazione autonoma delle attività agli agenti appropriati in base alle loro capacità. Per ulteriori informazioni, consulta Tasks nella CrewAI documentazione.
- Collaborazione tra agenti: framework per la comunicazione autonoma tra agenti e la condivisione delle conoscenze senza la mediazione umana. Per ulteriori informazioni, consulta Collaborazione nella documentazione. CrewAI
- Gestione dei processi: flussi di lavoro strutturati per l'esecuzione di attività autonome sequenziali e parallele. Per ulteriori informazioni, consulta Processi nella CrewAI documentazione.
- Selezione del modello di base: supporto per vari modelli di base, tra cui Anthropic Claude, i modelli Amazon Nova (Premier, Pro, Lite e Micro) su Amazon Bedrock e altri per l'ottimizzazione per diverse attività di ragionamento autonomo. Per ulteriori informazioni, consulta gli LLM nella documentazione. CrewAI
- Integrazione dell'API LLM: integrazione flessibile con più interfacce di servizio LLM, tra cui Amazon BedrockOpenAI, e implementazioni di modelli locali. Per ulteriori informazioni, consulta gli esempi di configurazione del provider nella documentazione. CrewAI
- Supporto multimodale: funzionalità emergenti per la gestione di testo, immagini e altre modalità per interazioni complete con agenti autonomi. Per ulteriori informazioni, consulta Using Multimodal Agents nella documentazione. CrewAI

Quando usare CrewAI

CrewAIè particolarmente adatto per scenari con agenti autonomi, tra cui:

- Problemi complessi che traggono vantaggio da competenze specializzate e basate sui ruoli che operano in modo autonomo

- Progetti che richiedono una collaborazione esplicita tra più agenti autonomi
- Casi d'uso in cui la scomposizione dei problemi basata sul team migliora la risoluzione autonoma dei problemi
- Scenari che richiedono una chiara separazione delle preoccupazioni tra i diversi ruoli degli agenti autonomi

Approccio di implementazione per CrewAI

CrewAIfornisce un'implementazione basata sui ruoli dell'approccio dei team di agenti di intelligenza artificiale agli stakeholder aziendali, come descritto in Getting Started nella CrewAI documentazione. Il framework consente alle organizzazioni di:

- Definisci agenti autonomi specializzati con ruoli, obiettivi e aree di competenza specifici.
- Assegna attività agli agenti in base alle loro capacità specializzate.
- Stabilisci dipendenze chiare tra le attività per creare flussi di lavoro strutturati.
- Orchestra la collaborazione tra più agenti per risolvere problemi complessi.

Questo approccio basato sui ruoli rispecchia le strutture dei team umani, il che lo rende intuitivo da comprendere e implementare per i leader aziendali. Le organizzazioni possono creare team autonomi con aree di competenza specializzate che collaborano per raggiungere gli obiettivi aziendali, in modo simile a come operano i team umani. Tuttavia, il team autonomo può lavorare ininterrottamente senza l'intervento umano.

Real-world esempio di CrewAI

AWS ha implementato sistemi multiagente autonomi utilizzando CrewAI integrato con Amazon Bedrock, come dettagliato nel CrewAI case study pubblicato. AWS e CrewAI ha sviluppato un framework sicuro e indipendente dal fornitore. L'architettura CrewAI open source «flows-and-crews» si integra perfettamente con i modelli di base, i sistemi di memoria e le barriere di conformità di Amazon Bedrock.

Gli elementi chiave dell'implementazione includono:

- Progetti e open source, oltre a progetti di riferimento CrewAI rilasciati che associano CrewAI gli agenti ai modelli AWS e agli strumenti di osservabilità di Amazon Bedrock. Hanno inoltre rilasciato sistemi esemplari come un team di controllo della AWS sicurezza composto da più agenti, flussi

di modernizzazione del codice e automazione del back-office per i beni di consumo confezionati (CPG).

- Integrazione dello stack di osservabilità: la soluzione integra il monitoraggio con Amazon CloudWatch e consente la tracciabilità e LangFuse il debug dal proof of concept alla produzione. AgentOps
- Dimostrato ritorno sull'investimento (ROI): i primi progetti pilota mostrano importanti miglioramenti: un'esecuzione più rapida del 70% per un grande progetto di modernizzazione del codice e una riduzione di circa il 90% dei tempi di elaborazione per un flusso di back-office CPG.

AutoGen

AutoGenè un framework open source rilasciato inizialmente da. Microsoft AutoGensi concentra sull'abilitazione di agenti di intelligenza artificiale autonomi conversazionali e collaborativi. Fornisce un'architettura flessibile per la creazione di sistemi multiagente con particolare attenzione alle interazioni asincrone e basate sugli eventi tra agenti per flussi di lavoro autonomi complessi.

Caratteristiche principali di AutoGen

AutoGenoffre le seguenti funzionalità chiave:

- Agenti conversazionali: basati su conversazioni in linguaggio naturale tra agenti autonomi, consentono un ragionamento sofisticato attraverso il dialogo. Per ulteriori informazioni, consulta Multi-agent Conversation Framework nella documentazione. AutoGen
- Architettura asincrona: Event-driven progettazione per interazioni non bloccanti tra agenti autonomi, che supporta flussi di lavoro paralleli complessi. Per ulteriori informazioni, consulta Risoluzione di più attività in una sequenza di chat asincrone nella documentazione. AutoGen
- Human-in-the-loop— Forte supporto alla partecipazione umana opzionale a flussi di lavoro degli agenti altrimenti autonomi, quando necessario. Per ulteriori informazioni, consulta Consentire il feedback umano negli agenti nella AutoGen documentazione.
- Generazione ed esecuzione di codice: funzionalità specializzate per agenti autonomi incentrati sul codice in grado di scrivere ed eseguire codice. Per ulteriori informazioni, consulta Code Execution nella AutoGen documentazione.
- Comportamenti personalizzabili: configurazione flessibile e autonoma degli agenti e controllo delle conversazioni per diversi casi d'uso. Per ulteriori informazioni, consulta agentchat.conversable_agent nella documentazione. AutoGen

- Selezione del modello Foundation: supporto per vari modelli di base, tra cui Anthropic Claude, Amazon Nova (Premier, Pro, Lite e Micro) su Amazon Bedrock e altri per diverse funzionalità di ragionamento autonomo. Per ulteriori informazioni, consulta LLM Configuration nella documentazione. AutoGen
- Integrazione dell'API LLM: configurazione standardizzata per più interfacce di servizio LLM, tra cui Amazon OpenAI Bedrock e. Azure OpenAI Per ulteriori informazioni, consulta oai.openai_utils nel riferimento API. AutoGen
- Elaborazione multimodale: supporto per l'elaborazione di testo e immagini per consentire ricche interazioni multimodali con agenti autonomi. Per ulteriori informazioni, consulta Interagire con i modelli multimodali: nella documentazione. GPT-4V AutoGen AutoGen

Quando usare AutoGen

AutoGenè particolarmente adatto per scenari con agenti autonomi, tra cui:

- Applicazioni che richiedono flussi conversazionali naturali tra agenti autonomi per ragionamenti complessi
- Progetti che richiedono sia un funzionamento completamente autonomo che capacità opzionali di supervisione umana
- Casi d'uso che prevedono la generazione, l'esecuzione e il debug di codice autonomi senza l'intervento umano
- Scenari che richiedono modelli di comunicazione tra agenti autonomi flessibili e asincroni

Approccio di implementazione per AutoGen

AutoGenfornisce un approccio di implementazione conversazionale per gli stakeholder aziendali, come descritto in Getting Started nella AutoGen documentazione. Il framework consente alle organizzazioni di:

- Crea agenti autonomi che comunicano attraverso conversazioni in linguaggio naturale.
- Implementa interazioni asincrone basate sugli eventi tra più agenti.
- Combina un funzionamento completamente autonomo con la supervisione umana opzionale quando necessario.
- Sviluppa agenti specializzati per diverse funzioni aziendali che collaborino attraverso il dialogo.

Questo approccio conversazionale rende il ragionamento del sistema autonomo trasparente e accessibile agli utenti aziendali. Decision-makers può osservare il dialogo tra gli agenti per capire come vengono raggiunte le conclusioni e, facoltativamente, partecipare alla conversazione quando è richiesto il giudizio umano.

Real-world esempio di AutoGen

Magentic-Oneè un sistema multiagente generalista open source progettato per risolvere autonomamente attività complesse e in più fasi in ambienti diversi, come descritto nel blog AI Frontiers. Microsoft Alla base c'è l'agente Orchestrator, che scompone gli obiettivi di alto livello e monitora i progressi utilizzando registri strutturati. Questo agente delega le attività secondarie ad agenti specializzati (comeWebSurfer,, FileSurfer and) e si adatta dinamicamente ripianificando quando necessario. Coder ComputerTerminal

Il sistema è basato sul AutoGen framework ed è indipendente dal modello, l'impostazione predefinita è GPT-4o. Raggiunge prestazioni all'avanguardia su benchmark come, e, il tutto senza regolazioni specifiche per attività. GAIA AssistantBench WebArena Inoltre, supporta l'estensibilità modulare e una valutazione AutoGenBench rigorosa tramite suggerimenti.

LlamaIndex

LlamaIndexè un framework di dati progettato specificamente per collegare modelli linguistici di grandi dimensioni (LLM) con fonti di dati esterne per consentire sofisticate applicazioni di Retrieval Augmented Generation (RAG) e di intelligenza artificiale agentica. Il framework fornisce astrazioni e flussi di lavoro di sviluppo accelerati per sistemi agentici, modelli di orchestrazione personalizzati e integrazioni di sistema che riducono i tempi di produzione per soluzioni di intelligenza artificiale basate sulla conoscenza.

Caratteristiche principali di LlamaIndex

LlamaIndexoffre un set completo di funzionalità che lo rendono particolarmente adatto per le applicazioni di intelligenza artificiale agentica aziendale:

- Data-centric architettura: eccelle nell'acquisizione, indicizzazione e recupero di informazioni da oltre 100 formati di dati tra cui PDF, documenti Word, fogli di calcolo e altro ancora. Microsoft Il framework trasforma i dati aziendali in basi di conoscenza interrogabili ottimizzate per gli agenti di intelligenza artificiale. Per ulteriori informazioni, consulta la documentazione relativa ad LlamaIndex.

- Production-ready implementazione: LlamaIndex offre sia framework open source che servizi gestitiLlamaCloud, fornendo funzionalità di livello aziendale tra cui controlli di sicurezza, scalabilità, integrazioni di osservabilità e flessibilità di implementazione. Per ulteriori informazioni, consulta la documentazione del framework. LlamaIndex
- Elaborazione avanzata dei documenti: LlamaCloud offre funzionalità di analisi, estrazione, indicizzazione e recupero dei documenti che gestiscono layout complessi, tabelle annidate, contenuti multimodali e persino note scritte a mano. Questa sofisticata analisi consente agli agenti di lavorare in modo efficace con documenti aziendali reali che contengono grafici, diagrammi e formattazioni complesse. Per ulteriori informazioni, consulta la documentazione relativa ad LlamaCloud.
- Orchestrazione dei flussi di lavoro: LlamaAgents fornisce un motore di orchestrazione asincrono basato sugli eventi per la creazione di sistemi agentici in più fasi. I flussi di lavoro supportano modelli complessi tra cui loop, esecuzione parallela, ramificazione condizionale e stateful resumption, il che li rende ideali per interazioni sofisticate tra agenti. Per ulteriori informazioni, consulta la documentazione sui flussi di lavoro. LlamaIndex
- Funzionalità di recupero agentico: modalità di recupero avanzate tra cui ricerca ibrida, ricerca semantica e routing automatico che determinano in modo intelligente la migliore strategia di recupero per ogni query. Il framework supporta il recupero composito su più knowledge base con riposizionamento per una maggiore precisione. Per ulteriori informazioni, consultate la documentazione RAG. LlamaIndex
- Osservabilità e valutazione: LlamaIndex si integra con una varietà di strumenti di osservabilità e valutazione. Questa funzionalità di integrazione consente di tracciare ed eseguire il debug delle applicazioni, valutarne le prestazioni e monitorare i costi. Per ulteriori informazioni, consulta la documentazione Tracing and Debugging and Evaluating. LlamaIndex

Quando usare LlamaIndex

LlamaIndexè particolarmente adatto per scenari di intelligenza artificiale agentica che enfatizzano i flussi di lavoro ad alta intensità di dati e la gestione della conoscenza:

- Document-heavy applicazioni che richiedono agli agenti di elaborare, analizzare ed estrarre informazioni da grandi volumi di documenti aziendali come contratti, report, manuali e documenti normativi

- Dalla prototipazione rapida a scenari di produzione in cui le organizzazioni desiderano creare e implementare rapidamente agenti incentrati sui documenti senza sovraccarichi di gestione dell'infrastruttura
- RAG-first architetture che danno priorità all'accuratezza del recupero e alla pertinenza del contesto, soprattutto quando si lavora con documenti complessi e multimodali contenenti tabelle, immagini e dati strutturati
- Multi-agent flussi di lavoro documentali che richiedono agenti specializzati per diversi aspetti dell'elaborazione dei documenti, come l'analisi, il riepilogo e il controllo della conformità

Approccio di implementazione per LlamaIndex

LlamaIndex fornisce sia elementi costitutivi di basso livello che astrazioni di alto livello che si adattano a diversi approcci di implementazione:

- Sviluppo rapido di applicazioni RAG funzionali in poche righe di codice utilizzando API di alto livello. LlamaIndex Questo approccio è LlamaIndex accessibile ai team aziendali e agli sviluppatori che non conoscono l'intelligenza artificiale agentica.
- Integrazione aziendale tramite LlamaHub i sistemi aziendali più diffusi SharePoint, tra cui Amazon Simple Storage Service (Amazon S3), database e API. Questo approccio consente una perfetta integrazione con l'infrastruttura di dati esistente.
- Opzioni di implementazione flessibili tra implementazioni open source con hosting autonomo per il massimo controllo o servizi LlamaCloud gestiti per ridurre i costi operativi e le funzionalità aziendali.
- Le applicazioni possono iniziare con semplici motori di query e aggiungere progressivamente funzionalità agentiche, orchestrazione multiagente e flussi di lavoro complessi man mano che i requisiti evolvono.

Real-world esempio di LlamaIndex

Questo esempio si concentra su una filiale di un'azienda aerospaziale specializzata in soluzioni operative e di navigazione aeronautica. Devono affrontare una sfida crescente che prevede la sperimentazione non coordinata di chatbot di intelligenza artificiale. Le sperimentazioni hanno portato a lavori ripetuti, lunghi cicli di sviluppo, ostacoli alla conformità e implementazioni isolate in tutta l'organizzazione.

Hanno sviluppato un framework di agenti unificato, una soluzione riutilizzabile basata su modelli basata su un framework LlamaIndex open source che rende la creazione di agenti molto più efficiente. Hanno confrontato diversi framework concorrenti, sia orientati alla catena che basati su grafici. Alla fine, hanno scelto tre vantaggi fondamentali: LlamaIndex il design flessibile, i componenti modulari e i controlli di orchestrazione pronti per la produzione.

La piattaforma riduce i tempi di sviluppo e implementazione degli agenti dell'87% da 512 a 64 ore. Questa riduzione è stata ottenuta consentendo ai team di creare agenti con circa 50 righe di codice e un file di configurazione JSON. I team hanno sfruttato un framework unificato con sicurezza integrata, conformità e accesso privilegiato al sistema. Per ulteriori dettagli, consulta i case study LlamaIndexdei clienti.

Confronto tra framework di intelligenza artificiale agentica

Quando scegli un framework di intelligenza artificiale agentica per lo sviluppo di agenti autonomi, considera in che modo ciascuna opzione si allinea ai tuoi requisiti specifici. Considerate non solo le sue capacità tecniche ma anche la sua idoneità organizzativa, tra cui l'esperienza del team, l'infrastruttura esistente e i requisiti di manutenzione a lungo termine. Molte organizzazioni potrebbero trarre vantaggio da un approccio ibrido, che sfrutta più framework per diversi componenti del loro ecosistema di intelligenza artificiale autonomo.

La tabella seguente confronta i livelli di maturità (più forte, forte, adeguato o debole) di ciascun framework in base alle dimensioni tecniche chiave. Per ogni framework, la tabella include anche informazioni sulle opzioni di implementazione in produzione e sulla complessità della curva di apprendimento.

Framework	AWS integration	Supporto multiagente autonomo	Complessità del workflow autonomo	Funzionalità multimodali	Selezione del modello Foundation	Integrazione con l'API LLM	Distribuzione in produzione	Curva di apprendimento
AutoGen	Debole	Forte	Forte	Sufficiente	adeguato	Forte	Fai da te (fai da te)	Ripido

CrewAI	Debole	Forte	Sufficiente	Debole	Sufficiente	adeguato	FAI DA TE	Moderata
LangChain / LangGraph	adeguato	Forte	Il più forte	Il più forte	Il più forte	Il più forte	Piattaforma per il fai da te	Ripido
LlamaIndex	Sufficiente	adeguato	Forte	Sufficiente	Forte	Forte	Piattaforma per fai da te	Moderata
Strands Agents	Il più forte	Forte	Il più forte	Forte	Forte	Il più forte	FAI DA TE	Moderata

Considerazioni sulla scelta di un framework di intelligenza artificiale agentica

Nello sviluppo di agenti autonomi, considera i seguenti fattori chiave:

- AWS integrazione dell'infrastruttura: le organizzazioni in cui hanno investito molto AWS trarranno i maggiori vantaggi dalle integrazioni native di Strands Agents with Servizi AWS per flussi di lavoro autonomi. Per ulteriori informazioni, consulta AWS Weekly Roundup (blog).AWS
- Selezione del modello di base: valuta quale framework offre il supporto migliore per i tuoi modelli di base preferiti (ad esempio, i modelli Amazon Nova su Amazon Bedrock o Anthropic Claude), in base ai requisiti di ragionamento del tuo agente autonomo. Per ulteriori informazioni, consulta Building Effective Agents sul sito Web. Anthropic
- Integrazione dell'API LLM: valuta i framework in base alla loro integrazione con le tue interfacce di servizio LLM (Large Language Model) preferite (ad esempio Amazon Bedrock o) per l'implementazione in produzione. OpenAI Per ulteriori informazioni, consulta Model Interfaces nella documentazione. Strands Agents
- Requisiti multimodali: per gli agenti autonomi che devono elaborare testo, immagini e voce, considera le funzionalità multimodali di ciascun framework. Per ulteriori informazioni, vedete Multimodalità nella documentazione. LangChain

- Complessità del flusso di lavoro autonomo: flussi di lavoro autonomi più complessi con una sofisticata gestione degli stati potrebbero favorire le funzionalità avanzate delle macchine a stati di. LangGraph
- Collaborazione autonoma in team: i progetti che richiedono una collaborazione autonoma esplicita e basata sui ruoli tra agenti specializzati possono trarre vantaggio dall'architettura orientata al team di. CrewAI
- Paradigma di sviluppo autonomo: i team che preferiscono modelli conversazionali e asincroni per agenti autonomi potrebbero preferire l'architettura basata sugli eventi di. AutoGen
- Approccio gestito o basato sul codice: le organizzazioni che desiderano un'esperienza completamente gestita con una codifica minima dovrebbero prendere in considerazione Amazon Bedrock Agents. Le organizzazioni che richiedono una personalizzazione più profonda potrebbero preferire Strands Agents altri framework con funzionalità specializzate che si allineano meglio ai requisiti specifici degli agenti autonomi.
- Preparazione alla produzione per sistemi autonomi: prendete in considerazione le opzioni di implementazione, le funzionalità di monitoraggio e le funzionalità aziendali per gli agenti autonomi di produzione.

Piattaforme

Le piattaforme Agentic AI forniscono i livelli fondamentali di runtime, orchestrazione e integrazione necessari per implementare, scalare e gestire sistemi agentici di livello di produzione. I framework definiscono il modo in cui gli agenti vengono creati e i protocolli regolano il modo in cui comunicano. Le piattaforme forniscono l'ambiente in cui questi agenti operano, collaborano ed evolvono in modo sicuro su larga scala.

Le piattaforme Agentic combinano funzionalità di esecuzione dei modelli, gestione del contesto, integrazione degli strumenti, osservabilità e governance in ambienti unificati. Queste piattaforme consentono alle organizzazioni di passare dalla sperimentazione all'implementazione su scala aziendale.

In questa sezione:

Perché le piattaforme sono importanti

Le piattaforme Agentic AI sono fondamentali per le organizzazioni che cercano di rendere operativi i sistemi autonomi in produzione. Offrono le seguenti funzionalità:

- Forniscono l'orchestrazione del runtime per l'hosting, il ridimensionamento e il coordinamento degli agenti.
- Gestisci lo stato, il contesto e la memoria nei flussi di lavoro multiagente.
- Offri controlli di sicurezza, identità e governance allineati agli standard aziendali.
- Integrazione con ecosistemi di utensili e sistemi esterni tramite standard APIs o protocolli.
- Abilita l'osservabilità e la verificabilità tra le interazioni degli agenti e i flussi di eventi.
- Supporta l'interoperabilità tra modelli, consentendo agli agenti di utilizzare più modelli di base in un unico ambiente.

Queste funzionalità trasformano i singoli agenti in sistemi coordinati e adattivi in grado di operare in modo affidabile entro i confini aziendali e normativi.

Tipi di piattaforme di intelligenza artificiale agentica

Le piattaforme Agentic AI rientrano in genere in una o più delle seguenti categorie:

- Agente gestito: le piattaforme completamente gestite forniscono funzionalità integrate di infrastruttura, memoria e orchestrazione. Riducono il sovraccarico operativo e accelerano i tempi di produzione.
- Orchestrazione open source: le piattaforme agentiche open source offrono flessibilità e trasparenza per le organizzazioni che preferiscono ambienti personalizzabili o l'implementazione locale.
- Impresa ibrida: le piattaforme ibride integrano componenti gestiti e ospitati autonomamente, combinando la scalabilità dei servizi gestiti sul cloud con il controllo dei sistemi aziendali.

Considerazioni sulla selezione della piattaforma

Nella scelta o nella progettazione di una piattaforma di intelligenza artificiale agentica, le organizzazioni devono considerare quanto segue:

- Profondità di integrazione: valuta in che modo la piattaforma si integra con le fonti di dati, gli strumenti e i protocolli esistenti.
- Scalabilità: assicurati che la piattaforma possa scalare dinamicamente per supportare carichi di lavoro autonomi e la collaborazione tra più agenti.
- Sicurezza e conformità: valuta le funzionalità di privacy, crittografia e governance dei dati rispetto ai requisiti organizzativi e regionali.
- Estensibilità: scegli piattaforme con architetture modulari che consentono di aggiungere nuovi strumenti, modelli o agenti nel tempo.
- Osservabilità: preferisci piattaforme che forniscano telemetria dettagliata, tracciabilità e registri di controllo per le interazioni tra agenti.
- Efficienza in termini di costi: prendi in considerazione modelli serverless o basati sull'utilizzo per ottimizzare i costi per carichi di lavoro variabili.

Agent per Amazon Bedrock

Amazon Bedrock Agents è un servizio completamente gestito che ti consente di creare e configurare agenti autonomi nelle tue applicazioni. Può orchestrare le interazioni tra modelli di base, fonti di dati, applicazioni software e conversazioni con gli utenti. Il suo approccio semplificato alla creazione di agenti non richiede la fornitura di capacità, la gestione dell'infrastruttura o la scrittura di codice personalizzato.

Caratteristiche principali di Amazon Bedrock Agents

Amazon Bedrock Agents include le seguenti funzionalità chiave:

- Servizio completamente gestito: gestione completa dell'infrastruttura senza la necessità di fornire capacità o gestire i sistemi sottostanti. Per ulteriori informazioni, consulta Automatizza le attività nella tua applicazione utilizzando agenti AI nella documentazione di Amazon Bedrock.
- Sviluppo basato sulle API: definisci ed esegui gli agenti tramite semplici chiamate API specificando modelli, istruzioni, strumenti e parametri di configurazione. Per ulteriori informazioni, consulta Creare e configurare l'agente manualmente nella documentazione di Amazon Bedrock.
- Gruppi di azioni: definisci azioni specifiche che il tuo agente può eseguire creando gruppi di azioni con schemi API. Per ulteriori informazioni, consulta Utilizzare i gruppi di azioni per definire le azioni che il tuo agente deve eseguire nella documentazione di Amazon Bedrock.
- Integrazione con la Knowledge Base: connettiti senza problemi alle Knowledge Base di Amazon Bedrock per aumentare le risposte degli agenti con i dati della tua organizzazione. Per ulteriori informazioni, consulta Augment Response generation per il tuo agente con knowledge base nella documentazione di Amazon Bedrock.
- Modelli di prompt avanzati: personalizza il comportamento degli agenti tramite modelli di prompt per la preelaborazione, l'orchestrazione, la generazione di risposte della knowledge base e la post-elaborazione. Per ulteriori informazioni, consulta Migliora la precisione dell'agente utilizzando modelli di prompt avanzati in Amazon Bedrock nella documentazione di Amazon Bedrock.
- Tracciamento e osservabilità: monitora il processo di step-by-step ragionamento dell'agente utilizzando funzionalità di tracciamento integrate. Per ulteriori informazioni, consulta il processo di step-by-step ragionamento dell'agente Track che utilizza trace nella documentazione di Amazon Bedrock.
- Controllo delle versioni e alias: crea più versioni del tuo agente e distribuiscile tramite alias per implementazioni controllate. Per ulteriori informazioni, consulta Implementare e utilizzare un agente Amazon Bedrock nella tua applicazione nella documentazione di Amazon Bedrock.

Quando usare Amazon Bedrock Agents

Amazon Bedrock Agents è particolarmente adatto per scenari di agenti autonomi, tra cui:

- Organizzazioni che desiderano un'esperienza completamente gestita per la creazione e l'implementazione di agenti senza gestire l'infrastruttura
- Progetti che richiedono lo sviluppo e l'implementazione rapidi di agenti tramite la configurazione anziché il codice
- Casi d'uso che traggono vantaggio dalla stretta integrazione con altre funzionalità di Amazon Bedrock come Knowledge Bases e Guardrails
- I team non dispongono delle risorse interne necessarie per creare agenti partendo da zero, ma necessitano di funzionalità autonome pronte per la produzione

Approccio di implementazione per Amazon Bedrock Agents

Amazon Bedrock Agents offre un approccio di implementazione basato sulla configurazione per gli stakeholder aziendali. Il servizio consente alle organizzazioni di:

- Definisci gli agenti tramite chiamate Console di gestione AWS o API senza scrivere codice complesso.
- Crea gruppi di azioni che specificano le operazioni APIs e che l'agente può eseguire.
- Connect le knowledge base per fornire all'agente informazioni specifiche del dominio.
- Verifica e ripeti il comportamento degli agenti tramite un'interfaccia visiva.

Questo approccio gestito consente ai team aziendali di sviluppare e implementare rapidamente agenti autonomi senza competenze tecniche approfondite nello sviluppo di modelli di intelligenza artificiale o nella gestione dell'infrastruttura.

Esempio reale di Amazon Bedrock Agents

Una soluzione per le operazioni finanziarie (FinOps) descritta in questo post del AWS blog utilizza il framework multi-agente Amazon Bedrock per creare un assistente per la gestione dei costi del cloud basato sull'intelligenza artificiale. Il conveniente modello di base Amazon Nova alimenta la soluzione in cui un agente FinOps Supervisor centrale delega le attività ad agenti specializzati. Questi agenti recuperano e analizzano i dati di AWS spesa utilizzando AWS Cost Explorer e generano consigli per ridurre i costi utilizzando. AWS Trusted Advisor

Il sistema include l'accesso sicuro degli utenti tramite Amazon Cognito, un front-end ospitato su AWS Amplify, e gruppi di AWS Lambda azione per analisi e previsioni in tempo reale. I team finanziari possono porre domande in linguaggio naturale come «Quali erano i miei costi a febbraio 2025?» Il sistema risponde con interruzioni dettagliate, suggerimenti di ottimizzazione e previsioni, il tutto all'interno di un'architettura scalabile e senza server implementata utilizzando. AWS CloudFormation

Amazon Bedrock AgentCore

Amazon Bedrock AgentCore è una piattaforma agentica per creare, distribuire e gestire agenti altamente capaci in modo sicuro su larga scala utilizzando qualsiasi framework, modello o protocollo. Utilizzando AgentCore, puoi fare quanto segue, il tutto senza alcuna gestione dell'infrastruttura:

- Crea agenti più velocemente.
- Consenti agli agenti di intraprendere azioni su più strumenti e dati.
- Esegui gli agenti in modo sicuro con bassa latenza e tempi di esecuzione prolungati.
- Monitora gli agenti in produzione.

AgentCore elimina l'onere indifferenziato della creazione di un'infrastruttura di agenti specializzati, consentendovi di accelerare la produzione degli agenti. I suoi servizi possono essere utilizzati insieme o indipendentemente e sono compatibili con qualsiasi framework, tra cuiCrewAI, LangGraphLlamaIndex, e. Strands Agents AgentCoreè anche compatibile con qualsiasi modello di base disponibile all'interno o all'esterno di Amazon Bedrock, offrendo la massima flessibilità.

AgentCore è composto da diversi servizi chiave:

- Amazon Bedrock AgentCore Runtime: fornisce un ambiente sicuro, serverless e scalabile per ospitare ed eseguire i tuoi agenti, senza dover gestire alcuna infrastruttura necessaria per la distribuzione e l'esecuzione di agenti o strumenti di intelligenza artificiale.
- Amazon Bedrock AgentCore Memory: offre un sistema di memoria gestita che consente agli agenti di conservare il contesto delle interazioni per conversazioni più personalizzate e coerenti, mantenendo conoscenze sia immediate che a lungo termine.
- Amazon Bedrock AgentCore Gateway: semplifica il processo di creazione, protezione e ricerca degli strumenti giusti per gli agenti. Con AgentCore Gateway, gli sviluppatori possono convertire APIs le funzioni Lambda e i servizi esistenti in strumenti compatibili con il Model Context Protocol (MCP) e renderli disponibili agli agenti.

- Amazon Bedrock AgentCore Identity: fornisce un servizio di gestione delle identità e degli accessi degli agenti sicuro e scalabile che accelera lo sviluppo degli agenti AI. Con AgentCore Identity, puoi assegnare identità uniche e verificabili agli agenti, abilitando un controllo granulare degli accessi e interazioni sicure basate sugli agenti con i sistemi aziendali.
- Strumenti AgentCore integrati di Amazon Bedrock: ti consentono di utilizzare strumenti integrati per migliorare il flusso di lavoro di sviluppo e test. Usa questi strumenti per interagire efficacemente con la tua applicazione, permettendo agli agenti AI di scrivere ed eseguire codice in modo sicuro in ambienti sandbox. Utilizza lo strumento browser per consentire agli agenti AI di interagire con i siti Web su larga scala.
- Amazon Bedrock AgentCore Observability: offre funzionalità di registrazione e monitoraggio, offrendoti visibilità in tempo reale sulle prestazioni e sul comportamento del tuo agente per facilitare il debug e l'ottimizzazione.

Caratteristiche principali di AgentCore

AgentCore include le seguenti funzionalità chiave:

- Completamente gestito ed estensibile: AgentCore è un servizio completamente gestito, il che significa che AWS gestisce l'infrastruttura e la manutenzione sottostanti. È anche estensibile, il che consente di personalizzare e migliorare la funzionalità degli agenti. Per ulteriori informazioni, consulta Get started with AgentCore Runtime nella AgentCore documentazione.
- Memoria a lungo e breve termine: offri interazioni più personalizzate e pertinenti dotando gli agenti di un sistema di memoria per richiamare il contesto dalle conversazioni in corso e dalle conoscenze a lungo termine. Per ulteriori informazioni, consulta Get started with AgentCore Memory nella AgentCore documentazione.
- Sviluppo e integrazione semplificati degli strumenti: consenti ai tuoi agenti di scoprire e utilizzare gli strumenti attraverso un unico endpoint sicuro. Trasforma rapidamente le risorse aziendali esistenti in strumenti pronti per l'uso con poche righe di codice, permettendo agli sviluppatori di concentrarsi sulla creazione di funzionalità uniche. Per ulteriori informazioni, consulta la Guida introduttiva a AgentCore Gateway nella documentazione. AgentCore
- Infrastruttura sicura e scalabile: AgentCore fornisce un ambiente sicuro e scalabile per l'implementazione e il funzionamento degli agenti. Include funzionalità per la gestione delle identità e degli accessi, la crittografia dei dati e la sicurezza della rete. Per ulteriori informazioni, consulta Get started with AgentCore Identity nella AgentCore documentazione.

- Integrazione con un'ampia gamma di strumenti: consente di integrare gli agenti con una varietà di strumenti, tra cui un interprete di codice e uno strumento browser che puoi creare utilizzando gli strumenti AgentCore integrati. Per ulteriori informazioni, consulta Guida introduttiva a AgentCore Code Interpreter e Guida introduttiva a AgentCore Browser nella AgentCore documentazione.
- Osservabilità e monitoraggio completi: ottieni una visibilità approfondita sui tuoi agenti con strumenti completi per tracciare, eseguire il debug e monitorare le loro prestazioni in produzione. Visualizza l'intero percorso di esecuzione dell'agente per verificarne il ragionamento e risolvere gli errori. Utilizza dashboard in tempo reale e dati di telemetria standardizzati per tenere traccia delle principali metriche operative. Per ulteriori informazioni, consulta Aggiungere osservabilità alle AgentCore risorse Amazon Bedrock nella AgentCore documentazione.

Quando usare AgentCore

AgentCore è particolarmente adatto per scenari con agenti autonomi, tra cui:

- Organizzazioni che desiderano accelerare lo sviluppo e ridurre il sovraccarico operativo con un servizio completamente gestito che gestisce infrastruttura, sicurezza, strumenti integrati, osservabilità e scalabilità
- Progetti che richiedono flessibilità con servizi modulari che funzionano insieme o indipendentemente e sono compatibili con qualsiasi framework, come CrewAI orLangGraph, e qualsiasi modello di base da qualsiasi fonte
- Casi d'uso che richiedono agenti conversazionali e statici che devono mantenere il contesto e imparare dalle interazioni passate per fornire risposte personalizzate e pertinenti
- Agenti in grado di eseguire attività complesse grazie alla semplice integrazione con diverse applicazioni, fonti di dati e APIs

Approccio di implementazione per AgentCore

AgentCore è progettato per le organizzazioni che desiderano spostare gli agenti di intelligenza artificiale dalla fase di prototipazione, creata utilizzando framework di agenti open source o personalizzati, alla produzione. Con AgentCore, le organizzazioni possono fare quanto segue:

- Implementa gli agenti in modo sicuro su un'infrastruttura serverless, supportando qualsiasi framework e modello, con isolamento delle sessioni e gestione integrata delle identità e degli accessi per la end-to-end sicurezza e la conformità. Crea rapidamente agenti AgentCore Runtime per i principali framework di agenti utilizzando lo starter toolkit.

- Migliora gli agenti integrando la memoria persistente per la conservazione del contesto, semplificando lo sviluppo e l'integrazione degli strumenti tramite Gateway. AgentCore Sfrutta lo strumento di navigazione integrato e l'interprete di codice per flussi di lavoro avanzati.
- Traccia, esegui il debug e monitora gli agenti AI in produzione utilizzando dashboard di osservabilità basati su Amazon CloudWatch Application Insights e OpenTelemetry tracciando i parametri chiave delle AgentCore risorse (runtime, memoria, gateway e strumenti).
- Accelera l'implementazione e l'innovazione con servizi modulari completamente gestiti, blocchi componibili insieme o indipendentemente, con qualsiasi framework di agenti e provider di modelli. Questa flessibilità aiuta le organizzazioni a passare più rapidamente dal prototipo alla produzione.

Questo approccio gestito consente alle organizzazioni di creare, implementare ed eseguire agenti di intelligenza artificiale e sistemi multiagente di livello aziendale in modo rapido e sicuro su qualsiasi scala.

Esempio reale di AgentCore

AWS ha osservato che una delle maggiori banche dell'America Latina utilizza da AI/ML anni un'esperienza bancaria digitale iperpersonalizzata e sicura. La banca sta ampliando i servizi di intelligenza artificiale agentica utilizzandoli AgentCore per fornire ai clienti interazioni intuitive, maggiore sicurezza e maggiore automazione. Secondo il CTO, AgentCore dovrebbe sostenere i loro sforzi volti a soddisfare gli impegni assunti con i clienti su larga scala. AgentCore fornisce ai propri sviluppatori gli strumenti e la flessibilità necessari per creare e gestire agenti, contribuendo al contempo a garantire la conformità alle normative finanziarie.

Protocolli

Gli agenti di intelligenza artificiale richiedono protocolli di comunicazione standardizzati per interagire con altri agenti e servizi. Le organizzazioni che implementano architetture di agenti devono affrontare sfide significative in termini di interoperabilità, indipendenza dai fornitori e preparazione dei propri investimenti al futuro.

Questa sezione aiuta a navigare nel panorama dei agent-to-agent protocolli con particolare attenzione agli standard aperti che massimizzano la flessibilità e l'interoperabilità. (Per informazioni sui agent-to-tool protocolli, consulta Strategia di integrazione degli strumenti più avanti in questa guida.)

Questa sezione evidenzia il Model Context Protocol (MCP), uno standard aperto originariamente sviluppato Anthropic nel 2024. Oggi supporta AWS attivamente MCP attraverso contributi allo sviluppo e all'implementazione del protocollo. AWS collabora con i principali framework di agenti open source, tra cui, e LangGraph CrewAILlamaIndex, per plasmare il futuro della comunicazione tra agenti sul protocollo. Per ulteriori informazioni, vedere Protocolli aperti per l'interoperabilità degli agenti, parte 1: Comunicazione tra agenti su MCP (Blog).AWS

In questa sezione:

- Perché è importante la selezione del protocollo
- Agent-to-agent protocolli
- Selezione dei protocolli agentici
- Strategia di implementazione per i protocolli agentici
- Guida introduttiva a MCP
- ???

Perché è importante la selezione del protocollo

La selezione dei protocolli determina fondamentalmente il modo in cui è possibile creare ed evolvere l'architettura degli agenti AI. Scegliendo protocolli che supportano la portabilità tra framework di agenti, ottieni la flessibilità necessaria per combinare diversi sistemi di agenti e flussi di lavoro per soddisfare le tue esigenze specifiche.

I protocolli aperti consentono di integrare gli agenti su più framework. Ad esempio, utilizzali LangChain per la prototipazione rapida e l'implementazione di sistemi di produzione comunicando tramite un protocollo comuneStrands Agents, come MCP o il protocollo Agent2Agent (A2A). Questa flessibilità riduce la dipendenza da specifici provider di intelligenza artificiale, semplifica l'integrazione con i sistemi esistenti e consente di migliorare le capacità degli agenti nel tempo.

I protocolli ben progettati stabiliscono inoltre modelli di sicurezza coerenti per l'autenticazione e l'autorizzazione in tutto l'ecosistema degli agenti. Soprattutto, la portabilità del protocollo preserva la libertà di adottare nuovi framework e funzionalità degli agenti man mano che emergono. La scelta di protocolli aperti protegge l'investimento nello sviluppo di agenti mantenendo al contempo l'interoperabilità con i sistemi di terze parti.

Vantaggi dei protocolli aperti

Quando si implementano le proprie estensioni o si creano sistemi di agenti personalizzati, i protocolli aperti offrono vantaggi convincenti:

- Documentazione e trasparenza: in genere forniscono una documentazione completa e implementazioni trasparenti
- Supporto comunitario: accesso a comunità di sviluppatori più ampie per la risoluzione dei problemi e le migliori pratiche
- Garanzie di interoperabilità: maggiore garanzia che le estensioni funzionino su diverse implementazioni
- Compatibilità futura: riduzione del rischio di interruzione delle modifiche o di obsolescenza
- Influenza sullo sviluppo: opportunità di contribuire all'evoluzione del protocollo

Agent-to-agent protocolli

La tabella seguente fornisce una panoramica dei protocolli agentici che consentono a più agenti di collaborare, delegare attività e condividere informazioni.

Protocollo	Ideale per	Considerazioni
Comunicazione tra agenti MCP	Organizzazioni alla ricerca di modelli flessibili di collabora zione tra agenti	• Un'estensione del Model Context Protocol (MCP) proposta da That AWS si

		basa sulle sue basi esistenti per la comunicazione agent-to-agent • Consente una collaborazione senza interruzioni tra gli agenti con una sicurezza basata sulla sicurezza OAuth
Protocollo A2A	Ecosistemi di agenti multipiattaforma	• Supportato da Google • Standard più recente con un'adozione più limitata rispetto a MCP

Decidere tra le opzioni di protocollo

Quando implementate agent-to-agent la comunicazione, abbinate i vostri requisiti di comunicazione specifici alle funzionalità del protocollo appropriate. Modelli di interazione diversi richiedono funzionalità di protocollo diverse. La tabella seguente descrive i modelli di comunicazione più comuni e consiglia le scelte di protocollo più adatte per ogni scenario.

Pattern	Descrizione	Scelta del protocollo ideale
Richiesta e risposta semplici	Interazioni una tantum tra agenti	MCP con flussi stateless
Dialoghi statici	Conversazioni continue con contesto	MCP con gestione delle sessioni
Collaborazione tra più agenti	Interazioni complesse tra più agenti	Interagente MCP o AutoGen
Flussi di lavoro basati sul team	Team di agenti gerarchici con ruoli definiti	Interagente MCP, oppure CrewAI AutoGen

Oltre ai modelli di comunicazione, diversi fattori tecnici e organizzativi possono influenzare la selezione del protocollo. La tabella seguente riporta le considerazioni chiave che possono aiutarvi a valutare quale protocollo si allinea maggiormente ai vostri requisiti di implementazione specifici.

Considerazione	Descrizione	Esempio
Modello di sicurezza	Requisiti di autenticazione e autorizzazione	OAuth 2.0 in MCP
Ambiente di implementazione	Dove gli agenti funzioneranno e comunicheranno	Macchina distribuita o singola
Compatibilità con l'ecosistema	Integrazione con i framework di agenti esistenti	LangChain o Strands Agents
Esigenze di scalabilità	Crescita prevista delle interazioni tra agenti	Funzionalità di streaming di MCP

Selezione dei protocolli agentici

Per la maggior parte delle organizzazioni che creano sistemi di agenti di produzione, il Model Context Protocol (MCP) offre la base per la comunicazione più completa e ben supportata. agent-to-agent MCP beneficia dei contributi attivi allo sviluppo AWS e della comunità open source.

La selezione dei protocolli agentici giusti è importante per le organizzazioni che desiderano implementare l'intelligenza artificiale agentica in modo efficace. Le considerazioni variano in base al contesto organizzativo.

Considerazioni sulla selezione del protocollo agentico

Le organizzazioni devono prendere in considerazione le seguenti best practice nella scelta dei protocolli per i sistemi di intelligenza artificiale agentici:

- Dare priorità agli standard aperti: le organizzazioni dovrebbero adottare protocolli aperti come MCP per garantire l'interoperabilità e l'estensibilità a lungo termine e ridurre il rischio di dipendenza da un fornitore.

- Equilibra velocità e flessibilità: le startup e i primi utenti possono iniziare con protocolli proprietari ben supportati per uno sviluppo rapido, ma dovrebbero definire un percorso di migrazione verso gli standard aperti man mano che i sistemi maturano.
- Implementazione di livelli di astrazione: le aziende dovrebbero implementare l'astrazione del protocollo per semplificare la migrazione, consentire l'adozione ibrida e strategie di integrazione a prova di futuro.
- Enfatizzare la sicurezza e la conformità: le organizzazioni dei settori regolamentati devono selezionare protocolli con solide funzionalità di autenticazione, crittografia e audit per soddisfare i requisiti di governance e conformità.
- Valuta la maturità dell'ecosistema: tutte le organizzazioni devono valutare lo stato di salute, l'adozione e il supporto della comunità di ciascun protocollo per garantire la sostenibilità e ridurre al minimo il debito tecnico.
- Impegnarsi nello sviluppo degli standard: le organizzazioni dovrebbero partecipare a enti di normazione o comunità open source per contribuire a plasmare l'evoluzione del protocollo e influenzare le migliori pratiche.
- Tieni conto della sovranità dei dati: i governi e i settori regolamentati dovrebbero garantire che le scelte di protocollo siano in linea con i requisiti di residenza e sovranità dei dati nelle regioni di implementazione.
- Sfrutta i servizi gestiti: ove possibile, utilizza implementazioni gestite o senza server di protocolli agentici per ridurre la complessità operativa e accelerare l'implementazione.

Strategia di implementazione per i protocolli agentici

Per implementare efficacemente i protocolli agentici in tutta l'organizzazione, prendete in considerazione i seguenti passaggi strategici:

1. Inizia con l'allineamento degli standard: adotta protocolli aperti consolidati, ove possibile.
2. Crea livelli di astrazione: implementa adattatori tra i tuoi sistemi e protocolli specifici.
3. Contribuisci agli standard aperti: partecipa alle comunità di sviluppo dei protocolli.
4. Monitora l'evoluzione del protocollo: rimani informato sugli standard e sugli aggiornamenti emergenti.
5. Verifica regolarmente l'interoperabilità: verifica che le implementazioni rimangano compatibili.

Guida introduttiva a MCP

AWS supporta attivamente il Model Context Protocol (MCP) attraverso contributi allo sviluppo e all'implementazione del protocollo. AWS collabora con i principali framework di agenti open source, tra cui, e LangGraph CrewAILlamaIndex, per plasmare il futuro della comunicazione tra agenti sul protocollo.

Per implementare l'MCP nell'architettura degli agenti, intraprendi le seguenti azioni:

1. Esplora le implementazioni MCP in framework come l'SDK. Strands Agents
2. Consulta la documentazione tecnica del Model Context Protocol.
3. Leggi Open Protocols for Agent Interoperability Part 1: Inter-Agent Communication on MCP (AWS Blog) per saperne di più sull'interoperabilità degli agenti.
4. Unisciti alla community MCP per influenzare l'evoluzione del protocollo.

MCP fornisce un livello di comunicazione che consente agli agenti di interagire con dati e servizi esterni e può essere utilizzato anche per consentire agli agenti di interagire con altri agenti. L'implementazione del trasporto HTTP Streamable del protocollo offre agli sviluppatori un set completo di modelli di interazione senza dover reinventare la ruota. Questi modelli supportano sia i request/response flussi stateless che la gestione delle sessioni stateful con persistent. IDs

Adottando protocolli aperti come MCP, consentite alla vostra organizzazione di creare sistemi di agenti che rimangano flessibili, interoperabili e adattabili di pari passo con l'evoluzione della tecnologia AI. Per informazioni sull'implementazione del agent-to-tool protocollo, consulta la strategia di integrazione degli strumenti più avanti in questa guida.

Guida introduttiva a A2A

Il protocollo Agent2Agent (A2A) consente la collaborazione decentralizzata tra agenti attraverso un livello semantico condiviso. Invece di affidare tutto il lavoro a un orchestratore centrale, A2A consente agli agenti di scoprirsi a vicenda, pubblicizzare le proprie capacità, negoziare attività e condividere il contesto utilizzando un protocollo leggero basato su JSON. Ogni agente pubblica un manifesto delle funzionalità.

L'esempio seguente mostra un manifesto di funzionalità A2A semplificato che pubblicizza le azioni supportate da un agente, gli input richiesti e i metadati operativi per consentire l'individuazione e la negoziazione delle attività:

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

Questo modello consente l'abbinamento dinamico delle capacità, la delega a metà attività e la collaborazione interorganizzativa. Gli agenti possono organizzarsi autonomamente in base alle attività, formare gruppi di lavoro temporanei e adattarsi all'ingresso o all'uscita di nuove funzionalità dal sistema.

A2A supporta interazioni che vanno da semplici richieste stateless a sessioni di negoziazione in più fasi, tra cui:

- Messaggistica diretta per una collaborazione a bassa latenza peer-to-peer
- Negoziazione semantica delle attività, in cui gli agenti selezionano il peer più adatto
- Scoperta basata sulle capacità, che consente una divisione emergente del lavoro
- Ancoraggio della sessione per interazioni statiche in più fasi

Adottando protocolli aperti e agent-native come A2A, le organizzazioni creano sistemi di intelligenza artificiale modulari, interoperabili e in grado di collaborare a livello transfrontaliero. A2A garantisce che gli ecosistemi degli agenti rimangano flessibili e possano evolversi man mano che vengono introdotti nuovi agenti, team o sistemi esterni, senza richiedere livelli di orchestrazione rigidi o accoppiamenti precedenti.

Per implementare il protocollo A2A nell'architettura degli agenti, intraprendi le seguenti azioni:

1. Consulta le specifiche del protocollo A2A: leggi l'ultima versione della specifica del protocollo Agent2Agent (A2A) per scoprire come funzionano i manifesti di capacità, i flussi di negoziazione e l'handshake dell'agente.
2. Esplora i runtime compatibili con A2A: valuta framework come Strands Agents SDK o livelli di runtime personalizzati che supportano i manifesti di funzionalità e la negoziazione in stile A2A. peer-to-peer
3. Implementa un manifesto di funzionalità per i tuoi agenti: definisci i campi di ciascun agente per consentire la scopertacan, needs il matchmaking e la collaborazione a livello di intenti. meta

4. Sperimenta con i modelli di negoziazione A2A: utilizza il ciclo richiesta-offerta-accettazione, le interrogazioni strutturate sulle capacità o la scoperta basata sul gossip per capire come gli agenti ragionano su chi debba gestire un'attività.
5. Testa A2A in un ambiente di infrastruttura mista: combina la negoziazione tra pari A2A con il routing degli eventi nativo tramite AWS Amazon per valutare modelli di coordinamento ibridi. EventBridge
6. Unisciti alla community A2A: partecipa al gruppo di lavoro aperto per rimanere aggiornato su estensioni, consigli di sicurezza e miglioramenti dell'interoperabilità tra fornitori e contribuisci allo sviluppo del protocollo.

Tools (Strumenti)

Gli agenti di intelligenza artificiale offrono valore interagendo con strumenti esterni e fonti di dati per eseguire attività utili. APIs La giusta strategia di integrazione degli strumenti ha un impatto diretto sulle capacità, sul livello di sicurezza e sulla flessibilità a lungo termine degli agenti.

Questa sezione vi aiuta a orientarvi nel panorama dell'integrazione degli strumenti, con particolare attenzione agli standard aperti che massimizzano la libertà e la flessibilità. La sezione evidenzia il Model Context Protocol (MCP) per l'integrazione degli strumenti e esamina gli strumenti specifici del framework e i meta-strumenti specializzati che migliorano i flussi di lavoro degli agenti.

In questa sezione:

- Categorie di strumenti
- Strumenti basati su protocolli
- Strumenti nativi del framework
- Meta-strumenti
- Strategia di integrazione degli strumenti
- Migliori pratiche di sicurezza per l'integrazione degli strumenti

Categorie di strumenti

I sistemi Building Agent comprendono tre categorie principali di strumenti.

Strumenti basati su protocolli

Gli strumenti basati su protocolli utilizzano protocolli di comunicazione standardizzati: agent-to-tool

- Strumenti MCP: strumenti standard aperti che funzionano su più framework con opzioni di esecuzione sia locali che remote.
- OpenAIfunction calling: strumenti proprietari specifici per i modelli. OpenAI
- Anthropictools — Una variante della richiesta di OpenAI funzioni per strumenti proprietari specifici dei modelli di Claude. Anthropic

Strumenti nativi del framework

Gli strumenti nativi del framework sono integrati direttamente in framework di agenti specifici:

- Strands Agents strumenti: strumenti leggeri e specifici quick-to-implement per il framework. Strands Agents
- LangChainstrumenti: strumenti Python basati su strumenti che sono strettamente integrati con l'LangChainecosistema.
- LlamaIndexstrumenti: strumenti ottimizzati per il recupero e l'elaborazione dei dati all'interno. LlamaIndex

Meta-strumenti

I meta-strumenti migliorano i flussi di lavoro degli agenti senza intraprendere azioni esterne direttamente:

- Strumenti per il flusso di lavoro: gestisci il flusso di esecuzione degli agenti, la logica di ramificazione e la gestione dello stato.
- Strumenti grafici per agenti: coordina più agenti in flussi di lavoro complessi.
- Strumenti di memoria: forniscono l'archiviazione e il recupero persistenti delle informazioni tra le sessioni degli agenti.
- Strumenti di riflessione: consentono agli agenti di analizzare e migliorare le proprie prestazioni.

Strumenti basati su protocolli

Quando si considerano gli strumenti basati sul protocollo, il Model Context Protocol (MCP) fornisce la base più completa e flessibile per l'integrazione degli strumenti. Come affermato nel post del blog AWS Open Source sull'interoperabilità degli agenti, AWS ha adottato MCP come protocollo strategico, contribuendo attivamente al suo sviluppo.

La tabella seguente descrive le opzioni per l'implementazione degli strumenti MCP.

Modello di distribuzione	Descrizione	Ideale per	Implementazione

Basato su uno studio locale	Gli strumenti vengono eseguiti nello stesso processo dell'agente	Sviluppo, test e strumenti semplici	Rapido da implement are senza sovraccar ico di rete
Basato su eventi inviati dal server locale (SSE)	Gli strumenti vengono eseguiti localment e ma comunicano tramite HTTP	Strumenti locali più complessi con separazione delle preoccupazioni	Migliore isolamento ma comunque bassa latenza
Streamable HTTP remoto	Gli strumenti vengono eseguiti su server remoti	Ambienti di produzion e e strumenti condivisi	Scalabile e gestito centralmente

Gli MCP ufficiali SDKs sono disponibili per la creazione di strumenti MCP:

- PythonSDK: implementazione completa con supporto completo del protocollo
- TypeScriptSDK —JavaScript/TypeScriptimplementazione per applicazioni web
- JavaSDK: implementazione Java per applicazioni aziendali

Questi SDKs forniscono gli elementi costitutivi per la creazione di strumenti compatibili con MCP nel linguaggio preferito, con implementazioni coerenti delle specifiche del protocollo.

Inoltre, AWS ha implementato MCP nell'SDK. Strands Agents L'Strands AgentsSDK offre un modo semplice per creare e utilizzare strumenti compatibili con MCP. La documentazione completa è disponibile nel repository. Strands Agents GitHub Per casi d'uso più semplici o quando si lavora al di fuori del Strands Agents framework, gli MCP ufficiali SDKs offrono implementazioni dirette del protocollo in più lingue.

Funzionalità di sicurezza degli strumenti MCP

Le funzionalità di sicurezza degli strumenti MCP includono quanto segue:

- OAuth Autenticazione 2.0/2.1: autenticazione standard del settore
- Ambito delle autorizzazioni: controllo granulare degli accessi per gli strumenti
- Scoperta delle funzionalità degli strumenti: individuazione dinamica degli strumenti disponibili
- Gestione strutturata degli errori: modelli di errore coerenti

Guida introduttiva agli strumenti MCP

Per implementare MCP per l'integrazione degli strumenti, intraprendi le seguenti azioni:

1. Esplora l'Strands AgentsSDK per un'implementazione MCP pronta per la produzione.
2. Consulta la documentazione tecnica MCP per comprendere i concetti fondamentali.
3. Usa gli esempi pratici descritti in questo post del blog AWS Open Source.
4. Inizia con semplici strumenti locali prima di passare a strumenti remoti.
5. Unisciti alla community MCP per influenzare l'evoluzione del protocollo.

Esplora AgentCore Gateway

Amazon Bedrock AgentCore Gateway offre agli sviluppatori un modo semplice e sicuro per creare, implementare, scoprire e connettersi a strumenti MCP e ad altri endpoint target su larga scala. Con AgentCore Gateway, gli sviluppatori possono convertire APIs AWS Lambda funzioni e servizi esistenti in strumenti compatibili con MCP. Quindi, con poche righe di codice, possono rendere questi strumenti disponibili agli agenti tramite gli endpoint AgentCore Gateway. AgentCore Gateway supporta OpenAPI e Lambda come tipi di input ed è l'unica soluzione che fornisce sia l'autenticazione completa in ingresso che l'autenticazione in uscita in un servizio completamente gestito. Smithy

Strumenti nativi del framework

Sebbene il Model Context Protocol (MCP) fornisca la base più flessibile, gli strumenti nativi del framework offrono vantaggi per casi d'uso specifici.

L'Strands AgentsSDK offre strumenti Python basati su un design leggero che richiede un sovraccarico minimo per operazioni semplici. Consentono un'implementazione rapida e consentono agli sviluppatori di creare strumenti con poche righe di codice. Inoltre, sono strettamente integrati per funzionare perfettamente all'interno del Strands Agents framework.

L'esempio seguente mostra come creare un semplice strumento meteorologico utilizzando. Strands Agents Gli sviluppatori possono trasformare rapidamente Python le funzioni in strumenti accessibili tramite agenti con un sovraccarico di codice minimo e generare automaticamente la documentazione appropriata dalla docstring della funzione.

```
#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."
```

Per la prototipazione rapida o per casi d'uso semplici, gli strumenti nativi del framework possono accelerare lo sviluppo. Tuttavia, per i sistemi di produzione, gli strumenti MCP offrono una migliore interoperabilità e flessibilità future rispetto agli strumenti nativi del framework.

La tabella seguente fornisce una panoramica di altri strumenti specifici del framework.

Framework	Tipo di utensile	Vantaggi	Considerazioni
AutoGen	Definizioni delle funzioni	Forte supporto multiagente	Microsoftecosistema
LangChain	Pythonclassi	Ampio ecosistema di strumenti predefiniti	Framework lock-in
LlamaIndex	Funzioni Python	Ottimizzato per le operazioni relative ai dati	Limitato a LlamaIndex

Meta-strumenti

I meta-strumenti non interagiscono direttamente con i sistemi esterni. Al contrario, migliorano le capacità degli agenti implementando modelli agentici. Questa sezione illustra il flusso di lavoro, il grafico degli agenti e i meta-strumenti di memoria.

Meta-strumenti per il flusso di lavoro

I meta-strumenti del flusso di lavoro gestiscono il flusso di esecuzione degli agenti:

- Gestione dello stato: mantenimento del contesto tra più interazioni tra agenti

- Logica di ramificazione: abilita percorsi di esecuzione condizionali
- Meccanismi di ripetizione dei tentativi: gestisci gli errori con sofisticate strategie di ripetizione dei tentativi

I framework di esempio con meta-strumenti per il flusso di lavoro includono funzionalità di workflow. LangGraphStrands Agents

Meta-strumenti Agent Graph

I meta-strumenti Agent Graph coordinano più agenti che lavorano insieme:

- Delega delle attività: assegna attività secondarie ad agenti specializzati
- Aggregazione dei risultati: combina gli output di più agenti
- Risoluzione dei conflitti: risolvi i disaccordi tra agenti

I framework apprezzano AutoGene sono CrewAIspecializzati nella coordinazione grafica degli agenti.

Meta-strumenti di memoria

I meta-strumenti di memoria forniscono archiviazione e recupero persistenti:

- Cronologia delle conversazioni: mantieni il contesto tra le sessioni
- Basi di conoscenza: archivia e recupera informazioni specifiche del dominio
- Archivi vettoriali: abilitano le funzionalità di ricerca semantica

Il sistema di risorse di MCP offre un modo standardizzato per implementare meta-strumenti di memoria che funzionano su diversi framework di agenti.

Strategia di integrazione degli strumenti

La strategia di integrazione degli strumenti scelta influisce direttamente su ciò che gli agenti possono realizzare e sulla facilità con cui il sistema può evolversi. Dai priorità ai protocolli aperti come il Model Context Protocol (MCP) utilizzando strategicamente strumenti e meta-strumenti nativi del framework. In questo modo, puoi creare un ecosistema di strumenti che rimanga flessibile e potente man mano che la tecnologia di intelligenza artificiale avanza.

Il seguente approccio strategico all'integrazione degli strumenti massimizza la flessibilità soddisfacendo al contempo le esigenze immediate dell'organizzazione:

1. Adottate MCP come base: MCP offre un modo standardizzato per collegare gli agenti a strumenti con potenti funzionalità di sicurezza. Inizia con MCP come protocollo di strumenti principale per:
   - Strumenti strategici che verranno utilizzati in più implementazioni di agenti.
   - Strumenti sensibili alla sicurezza che richiedono un'autenticazione e un'autorizzazione affidabili.
   - Strumenti che richiedono l'esecuzione remota in ambienti di produzione.
2. Usa strumenti nativi del framework quando appropriato: prendi in considerazione gli strumenti nativi del framework per:
   - Prototipazione rapida durante lo sviluppo iniziale.
   - Strumenti semplici non critici con requisiti di sicurezza minimi.
   - Funzionalità specifiche del framework che sfrutta funzionalità uniche.
3. Implementa meta-strumenti per flussi di lavoro complessi: aggiungi meta-strumenti per migliorare l'architettura degli agenti:
   - Inizia in modo semplice con modelli di flusso di lavoro di base.
   - Aggiungi complessità man mano che i tuoi casi d'uso maturano.
   - Standardizza le interfacce tra agenti e meta-strumenti.
4. Pianifica l'evoluzione: costruisci pensando alla flessibilità futura:
   - Interfacce degli strumenti documentali indipendentemente dalle implementazioni.
   - Crea livelli di astrazione tra agenti e strumenti.
   - Stabilisci percorsi di migrazione dai protocolli proprietari a quelli aperti.

Le migliori pratiche di sicurezza per l'integrazione degli strumenti

L'integrazione degli strumenti influisce direttamente sul tuo livello di sicurezza. Questa sezione descrive le best practice da prendere in considerazione per la propria organizzazione.

Autenticazione e autorizzazione

Utilizzate i seguenti robusti controlli di accesso:

- Usa OAuth 2.0/2.1: implementa l'autenticazione standard di settore per gli strumenti remoti.
- Implementa il privilegio minimo: concedi agli strumenti solo le autorizzazioni di cui hanno bisogno.

- Ruota le credenziali: aggiorna regolarmente le chiavi API e i token di accesso.

Protezione dei dati

Per contribuire alla protezione dei dati, adotta le seguenti misure:

- Convalida input e output: implementa la convalida dello schema per tutte le interazioni con gli strumenti.
- Crittografa i dati sensibili: utilizza TLS per tutte le comunicazioni con strumenti remoti.
- Implementa la riduzione al minimo dei dati: trasmetti solo le informazioni necessarie agli strumenti.

Monitoraggio e controllo

Mantieni la visibilità e il controllo utilizzando questi meccanismi:

- Registra tutte le chiamate agli strumenti: mantieni percorsi di controllo completi.
- Monitoraggio delle anomalie: rileva modelli di utilizzo degli strumenti insoliti.
- Implementa la limitazione della velocità: previene gli abusi tramite un numero eccessivo di strumenti.

Il modello di sicurezza Model Context Protocol (MCP) risponde a queste preoccupazioni in modo completo. Per ulteriori informazioni, consulta Considerazioni sulla sicurezza nella documentazione MCP.

Conclusioni

Il panorama dell'intelligenza artificiale agentica continua a evolversi rapidamente, offrendo alle organizzazioni nuovi modi potenti per costruire sistemi intelligenti e autonomi. Questa guida ha esplorato tre componenti essenziali per un'implementazione di successo: framework che forniscono le basi, piattaforme che forniscono l'ambiente, protocolli che consentono la comunicazione e strumenti che estendono le funzionalità.

Man mano che i framework maturano, è possibile aspettarsi una maggiore interoperabilità, la standardizzazione su protocolli come il Model Context Protocol (MCP) e funzionalità di orchestrazione più sofisticate per agenti autonomi. Le organizzazioni che oggi acquisiscono competenze con questi framework saranno ben posizionate per creare agenti sempre più autonomi e intelligenti che offrano un valore aziendale significativo.

Le piattaforme forniscono l'ambiente di esecuzione, governance e ciclo di vita in cui operano i sistemi agentici. Gestiscono questioni come l'identità, i limiti di sicurezza, l'osservabilità, la gestione della memoria, il fondamento delle sessioni e l'interazione sicura con strumenti e dati. Negli AWS ambienti, piattaforme come i runtime gestiti degli agenti e i servizi di orchestrazione consentono alle organizzazioni di implementare, monitorare, evolvere e governare agenti autonomi e sistemi agentici su larga scala. Le piattaforme collegano i framework fondamentali con i requisiti operativi del mondo reale.

La scelta dei protocolli degli agenti rappresenta una decisione strategica che bilancia le esigenze di sviluppo immediate con la flessibilità e l'interoperabilità a lungo termine. Dando priorità ai protocolli aperti e creando livelli di astrazione appropriati, le organizzazioni possono creare sistemi di agenti che rimangono adattabili alle tecnologie in evoluzione e soddisfano al contempo i requisiti aziendali attuali.

Per la maggior parte delle organizzazioni, MCP rappresenta una solida base grazie al suo standard aperto, all'ecosistema in crescita, al supporto per i modelli di agent-to-agent comunicazione e alle capacità di integrazione degli strumenti. AWS ha adottato MCP e Agent2Agent (A2A) come protocolli strategici, contribuendo attivamente al loro sviluppo e alla loro implementazione attraverso servizi come l'SDK. Strands Agents Utilizzando MCP o A2A insieme a strumenti e meta-strumenti nativi del framework appropriati, è possibile creare sistemi di agenti che offrono valore immediato pur rimanendo adattabili alle innovazioni future.

Resources

Utilizza le seguenti AWS e altre risorse relative allo sviluppo di agenti autonomi.

AWS Blog

- Amazon Bedrock AgentCore Memory: creazione di agenti sensibili al contesto
- Best practice per creare solide applicazioni di intelligenza artificiale generativa con Amazon Bedrock Agents — Parte 1
- Best practice per creare solide applicazioni di intelligenza artificiale generativa con Amazon Bedrock Agents — Parte 2
- Crea potenti pipeline RAG con LlamaIndex Amazon Bedrock
- Crea agenti AI affidabili con Amazon AgentCore Bedrock Observability
- Valuta le risposte RAG con Amazon Bedrock e RAGAS LlamaIndex
- Presentazione di Amazon Bedrock AgentCore Code Interpreter
- Presentazione di Amazon Bedrock AgentCore Gateway: trasformazione dello sviluppo di strumenti per agenti AI aziendali
- Presentazione di Amazon Bedrock AgentCore Identity: protezione dell'intelligenza artificiale agentica su larga scala
- Ti presentiamo un AI Strands Agents Agents SDK open source
- Protocolli aperti per l'interoperabilità degli agenti, parte 1: Comunicazione tra agenti su MCP
- Avvia e ridimensiona in modo sicuro agenti e strumenti su Amazon AgentCore Bedrock Runtime
- AWS Transform per.NET, il primo servizio di intelligenza artificiale agentica per modernizzare le applicazioni.NET su larga scala
- AWS Riepilogo settimanale: Strands Agents

AWS Guida prescrittiva

- Rendere operativa l'intelligenza artificiale agentica su AWS
- Fondamenti dell'IA agentica su AWS
- Modelli e flussi di lavoro di intelligenza artificiale agentica su AWS

- Creazione di architetture serverless per l'intelligenza artificiale agentica su AWS
- Creazione di architetture multi-tenant per l'intelligenza artificiale agentica su AWS
- Sicurezza per l'intelligenza artificiale agentica su AWS
- Opzioni e architetture di Retrieval Augmented Generation su AWS

AWS risorse

- Documentazione Amazon Bedrock
- Documentazione Amazon Bedrock AgentCore
- Amazon Bedrock AgentCore Starter Toolkit (repository) GitHub
- Documentazione Amazon Nova
- AWS Server MCP (GitHubrepository)

Altre risorse

- AutoGendocumentazione () Microsoft
- Creare agenti efficaci (Anthropic)
- CrewAI GitHubdeposito
- Documentazione di LangChain
- LangGraphpiattaforma
- Documentazione di LlamaIndex
- documentazione del Model Context Protocol
- Documentazione di Strands Agents
- Strands AgentsPanoramica degli strumenti
- Strands AgentsGuida rapida

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Nuova sezione	Sezione Piattaforme aggiunta	16 gennaio 2026
Pubblicazione iniziale	—	14 luglio 2025

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.