Frameworks, plateformes, protocoles et outils d'IA agentic sur AWS

AWS Conseils prescriptifs

AWS Conseils prescriptifs: Frameworks, plateformes, protocoles et outils d'IA agentic sur AWS

Table of Contents

Frameworks, plateformes, protocoles et outils d'IA agentic sur AWS

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

Janvier 2026 (historique du document)

L'IA agentique est un puissant paradigme à l'intersection de l'IA, des systèmes distribués et du génie logiciel. Il s'agit d'une classe de systèmes intelligents composés d'agents logiciels autonomes et asynchrones qui utilisent des modèles d'IA et s'intègrent à des outils et à des ressources. Les agents font preuve d'agentivité, peuvent percevoir le contexte, raisonner plutôt que les objectifs, prendre des décisions et prendre des mesures ciblées au nom des utilisateurs ou des systèmes. Ces agents opèrent de manière indépendante, souvent en collaboration, dans des environnements distribués et sont conçus pour poursuivre des objectifs délégués grâce à l'intelligence, à la mémoire et à l'intention intégrées.

En effet AWS, les entreprises peuvent tirer parti de l'IA agentic pour automatiser des flux de travail complexes, améliorer les processus de prise de décision et créer des systèmes plus réactifs.
Ce guide fournit des informations sur les composants clés nécessaires pour créer des solutions d'intelligence artificielle agentiques efficaces :

- Les frameworks dressent le profil des frameworks d'IA agentic actuels, y compris des examens de leurs avantages et de leurs cas d'utilisation. Découvrez comment ces cadres réduisent le transport indifférencié de charges lourdes selon les modèles, les protocoles et les outils. Comprenez les principaux critères de sélection afin de choisir le cadre adapté à vos besoins.
- Platforms fournit une vue d'ensemble des plateformes d'intelligence artificielle agentique (agent géré, orchestration open source et hybride) et des considérations relatives à la sélection ou à la conception.
- Protocoles explore les protocoles de communication standardisés essentiels pour les interactions entre agents. Agent-to-agentdes protocoles émergent, tels que le Model Context Protocol (MCP) open source et Agent2Agent (A2A), ainsi que d'autres implémentations propriétaires. Découvrez comment les protocoles communs permettent aux différents protocoles d'interagir de manière fluide.
- Tools fournit des informations sur les outils basés sur des protocoles (tels que le MCP), les outils natifs du framework et les méta-outils. Organisations peuvent créer une boîte à outils qui s'intègre

aux principaux systèmes de leurs flux de travail, permettant à la fois aux utilisateurs finaux et aux flux de travail agentiques basés sur le serveur.

Public visé

Ce guide s'adresse aux architectes, aux développeurs et aux leaders technologiques qui cherchent à exploiter la puissance des agents logiciels pilotés par l'IA au sein d'applications cloud natives modernes.

Objectifs

Ce guide vous aide à accomplir les tâches suivantes :

- Comparez différents frameworks d'IA agentic pour sélectionner celui qui convient le mieux à votre cas d'utilisation.
- Découvrez les plateformes d'IA agentic qui fournissent des capacités permettant de transformer des agents individuels en systèmes coordonnés et adaptatifs.
- Découvrez les avantages des protocoles ouverts pour créer des architectures d'IA agentiques durables.
- Créez une stratégie d'intégration d'outils appropriée lors de la création de systèmes d'agents.

À propos de cette série de contenus

Ce guide fait partie d'une série sur l'IA agentique sur. AWS Pour plus d'informations et pour consulter les autres guides de cette série, consultez Agentic AI sur le site Web de AWS Prescriptive Guidance.

Cadres

Les fondements de l'IA agentic AWS examinent les principaux modèles et flux de travail qui permettent un comportement autonome et axé sur les objectifs. Le choix du cadre est au cœur de la mise en œuvre de ces modèles. Un framework est la base logicielle du code préécrit qui fournit un environnement structuré et des fonctionnalités communes pour la création et la gestion, les outils et les capacités d'orchestration nécessaires pour créer des agents d'IA autonomes prêts pour la production.

Les frameworks d'IA agentique efficaces fournissent plusieurs fonctionnalités essentielles qui transforment les interactions brutes des grands modèles de langage (LLM) en systèmes coordonnés et intelligents capables de raisonner, de collaborer et d'agir :

- L'orchestration des agents coordonne le flux d'informations et la prise de décision entre un ou plusieurs agents afin d'atteindre des objectifs complexes sans intervention humaine.
- L'intégration d'outils permet aux agents d'interagir avec des systèmes externes, des API et des sources de données afin d'étendre leurs capacités au-delà du traitement du langage. Pour plus d'informations, consultez la section Présentation des outils dans la Strands Agents documentation.
- La gestion de la mémoire fournit un état persistant ou basé sur les sessions afin de maintenir le contexte dans toutes les interactions, ce qui est essentiel pour les tâches de longue durée ou adaptatives. Des frameworks plus avancés intègrent une mémoire à long terme pour stocker les résumés et les préférences des utilisateurs, permettant ainsi des expériences agentiques personnalisées et adaptées au contexte. Pour plus d'informations, consultez la section Comment penser aux frameworks d'agents sur le LangChain blog.
- La définition du flux de travail prend en charge des modèles structurés tels que les chaînes, le routage, la parallélisation et les boucles de réflexion qui permettent un raisonnement autonome sophistiqué.
- Le déploiement et le suivi facilitent le passage du développement à la production grâce à l'observabilité pour les systèmes autonomes. Pour plus d'informations, consultez l'annonce de disponibilité AgentCore générale d'Amazon Bedrock.

Ces fonctionnalités sont mises en œuvre selon différentes approches et approches dans l'ensemble du paysage des frameworks, chacune offrant des avantages distincts pour différents cas d'utilisation d'agents autonomes et contextes organisationnels.

Cette section décrit et compare les principaux frameworks pour la création de solutions d'IA agentiques, en mettant l'accent sur leurs points forts, leurs limites et leurs cas d'utilisation idéaux pour un fonctionnement autonome :

- Agents à mèches
- LangChain et LangGraph
- Équipage AI
- AutoGen
- ???
- Comparaison des frameworks d'IA agentiques

> **Note**
>
> Cette section couvre les cadres qui soutiennent spécifiquement l'agence de l'IA et ne couvre pas les interfaces frontales ou l'IA générative sans agence.

Strands Agents

Strands Agentsest un SDK open source initialement publié par AWS, comme décrit dans le blog AWS Open Source. Strands Agentsest conçu pour créer des agents d'intelligence artificielle autonomes selon une approche axée sur le modèle. Il fournit un cadre flexible et extensible conçu pour fonctionner parfaitement Services AWS tout en restant ouvert à l'intégration avec des composants tiers. Strands Agents est idéal pour créer des solutions totalement autonomes.

Principales fonctionnalités de Strands Agents

Strands Agentsinclut les principales fonctionnalités suivantes :

- Model-first design — Construit autour du concept selon lequel le modèle de base est au cœur de l'intelligence des agents, permettant un raisonnement autonome sophistiqué. Pour plus d'informations, consultez Agent Loop dans la Strands Agents documentation.
- Multi-agent modèles de collaboration : modèles de Built-in coordination tels que les modèles Swarm, Graph et Workflow qui permettent une collaboration et une gouvernance évolutives sur les réseaux d'agents distribués. Pour plus d'informations, consultez la section Multi-agent Patterns dans la documentation Strands Agents.

- Intégration MCP — Support natif du protocole MCP (Model Context Protocol), permettant de fournir un contexte normalisé aux LLM pour un fonctionnement autonome cohérent.
- Service AWS intégration — Connexion fluide à Amazon Bedrock,, AWS Lambda AWS Step Functions, et autres Services AWS pour des flux de travail autonomes complets. Pour plus d'informations, consultez le résuméAWS hebdomadaire (AWS blog).
- Sélection du modèle de base : prend en charge différents modèles de base, notamment Anthropic Claude, Amazon Nova (Premier, Pro, Lite et Micro) sur Amazon Bedrock, et d'autres pour optimiser les différentes capacités de raisonnement autonome. Pour plus d'informations, consultez Amazon Bedrock dans la Strands Agents documentation.
- Intégration de l'API LLM — Intégration flexible avec différentes interfaces de service LLM, notamment Amazon Bedrock, OpenAI, etc. pour le déploiement en production. Pour plus d'informations, consultez Amazon Bedrock Basic Usage dans la Strands Agents documentation.
- Capacités multimodales — Support de plusieurs modalités, notamment le traitement du texte, de la parole et de l'image pour des interactions complètes avec les agents autonomes. Pour plus d'informations, consultez Amazon Bedrock Multimodal Support dans la Strands Agents documentation.
- Écosystème d'outils : ensemble complet d'outils d' Service AWS interaction, avec extensibilité pour les outils personnalisés qui étendent les capacités autonomes. Pour plus d'informations, consultez la section Présentation des outils dans la Strands Agents documentation.

Quand utiliser Strands Agents

Strands Agentsest particulièrement bien adapté aux scénarios d'agents autonomes, notamment :

- Organisations qui s'appuient sur une AWS infrastructure et qui souhaitent une intégration native Services AWS pour des flux de travail autonomes
- Les équipes qui ont besoin de fonctionnalités de sécurité, d'évolutivité et de conformité de niveau professionnel pour les systèmes autonomes de production
- Projets nécessitant une flexibilité dans la sélection de modèles entre différents fournisseurs pour des tâches autonomes spécialisées
- Cas d'utilisation nécessitant une intégration étroite avec les AWS flux de travail et les ressources existants pour des processus autonomes de bout en bout

Approche de mise en œuvre pour Strands Agents

Strands Agentspropose une approche de mise en œuvre simple aux parties prenantes de l'entreprise, comme indiqué dans son guide de démarrage rapide pour Python et son guide de démarrage rapide pour. Typescript Le cadre permet aux organisations de :

- Sélectionnez des modèles de base tels qu'Amazon Nova (Premier, Pro, Lite ou Micro) sur Amazon Bedrock en fonction des besoins commerciaux spécifiques.
- Définissez des outils personnalisés qui se connectent aux systèmes et aux sources de données de l'entreprise.
- Traitez plusieurs modalités, notamment le texte, les images et le discours.
- Déployez des agents capables de répondre de manière autonome aux requêtes commerciales et d'effectuer des tâches.

Cette approche de mise en œuvre permet aux équipes commerciales de développer et de déployer rapidement des agents autonomes sans expertise technique approfondie dans le développement de modèles d'IA.

Real-world exemple de Strands Agents

AWS Transform pour .NET utilise Strands Agents ses capacités de modernisation des applications, comme décrit dans AWS Transform for .NET, le premier service d'intelligence artificielle agentique permettant de moderniser les applications .NET à grande échelle (AWS blog). Ce service de production emploie plusieurs agents autonomes spécialisés. Les agents travaillent ensemble pour analyser les applications .NET existantes, planifier des stratégies de modernisation et exécuter des transformations de code vers des architectures cloud natives sans intervention humaine. AWS Transform for .NET démontre l'état de préparation à la production Strands Agents des systèmes autonomes d'entreprise.

LangChain et LangGraph

LangChainest l'un des frameworks les plus établis de l'écosystème de l'IA agentique. LangGraphétend ses fonctionnalités pour prendre en charge les flux de travail complexes et dynamiques des agents, comme décrit dans le LangChainblog. Ensemble, ils fournissent une solution complète pour créer des agents IA autonomes sophistiqués dotés de riches capacités d'orchestration pour un fonctionnement indépendant.

Principales caractéristiques de LangChain et LangGraph

LangChainet LangGraph incluent les fonctionnalités clés suivantes :

- Écosystème de composants — Vaste bibliothèque de composants prédéfinis pour diverses fonctionnalités d'agents autonomes, permettant le développement rapide d'agents spécialisés. Pour plus d'informations, consultez Quickstart dans la LangChain documentation.
- Sélection du modèle de base — Support pour divers modèles de fondation, notamment les modèles Anthropic Claude, Amazon Nova (Premier, Pro, Lite et Micro) sur Amazon Bedrock, et d'autres pour différentes capacités de raisonnement. Pour plus d'informations, consultez la section Entrées et sorties dans la LangChain documentation.
- Intégration de l'API LLM : interfaces standardisées pour plusieurs fournisseurs de services de grands modèles linguistiques (LLM), notamment Amazon BedrockOpenAI, pour un déploiement flexible. Pour plus d'informations, consultez la section LLM dans la LangChain documentation.
- Traitement multimodal : Built-in prise en charge du traitement du texte, de l'image et du son pour permettre des interactions multimodales riches entre agents autonomes. Pour plus d'informations, consultez la section Multimodalité dans la LangChain documentation.
- Graph-based flux de travail : LangGraph permet de définir les comportements complexes des agents autonomes en tant que machines à états, prenant en charge une logique décisionnelle sophistiquée. Pour plus d'informations, consultez l'annonce de LangGraphPlatform GA.
- Abstractions de mémoire — Plusieurs options pour la gestion de la mémoire à court et à long terme, ce qui est essentiel pour les agents autonomes qui maintiennent le contexte au fil du temps. Pour plus d'informations, consultez Comment ajouter de la mémoire aux chatbots dans la LangChain documentation.
- Intégration d'outils — Écosystème riche d'intégrations d'outils à travers divers services et API, étendant les capacités des agents autonomes. Pour plus d'informations, consultez la section Outils de la LangChain documentation.
- LangGraph plateforme — Solution gérée de déploiement et de surveillance pour les environnements de production, prenant en charge les agents autonomes de longue durée. Pour plus d'informations, consultez l'annonce de LangGraphPlatform GA.

Quand utiliser LangChain et LangGraph

LangChainet LangGraph sont particulièrement bien adaptés aux scénarios d'agents autonomes, notamment :

- Workflows de raisonnement complexes en plusieurs étapes qui nécessitent une orchestration sophistiquée pour une prise de décision autonome
- Projets nécessitant l'accès à un vaste écosystème de composants prédéfinis et d'intégrations pour diverses capacités autonomes
- Équipes disposant d'une infrastructure et d'une expertise existantes en matière d'apprentissage automatique Python basé sur le machine learning (ML) et souhaitant créer des systèmes autonomes
- Cas d'utilisation nécessitant une gestion d'état complexe au cours de sessions d'agent autonomes de longue durée

Approche de mise en œuvre pour LangChain et LangGraph

LangChainet LangGraph fournir une approche de mise en œuvre structurée pour les parties prenantes de l'entreprise, comme indiqué dans la LangGraphdocumentation. Le cadre permet aux organisations de :

- Définissez des graphiques de flux de travail sophistiqués qui représentent les processus métier.
- Créez des modèles de raisonnement en plusieurs étapes avec des points de décision et une logique conditionnelle.
- Intégrez des capacités de traitement multimodales pour gérer divers types de données.
- Mettez en œuvre le contrôle qualité grâce à des mécanismes de révision et de validation intégrés.

Cette approche basée sur des graphiques permet aux équipes commerciales de modéliser des processus décisionnels complexes sous forme de flux de travail autonomes. Les équipes ont une visibilité claire sur chaque étape du processus de raisonnement et sont en mesure d'auditer les parcours décisionnels.

Real-world exemple de LangChain et LangGraph

Vodafonea mis en place des agents autonomes utilisant LangChain (etLangGraph) pour améliorer ses flux de travail d'ingénierie des données et d'exploitation, comme indiqué dans son étude de cas sur l'LangChainentreprise. Ils ont créé des assistants IA internes qui surveillent de manière autonome les indicateurs de performance, extraient des informations des systèmes de documentation et présentent des informations exploitables, le tout par le biais d'interactions en langage naturel.

La Vodafone mise en œuvre utilise des chargeurs de documents LangChain modulaires, l'intégration vectorielle et la prise en charge de plusieurs LLM (OpenAI, LLaMA 3 etGemini) pour prototyper et comparer rapidement ces pipelines. Ils ont ensuite structuré LangGraph l'orchestration multi-agents en déployant des sous-agents modulaires. Ces agents exécutent des tâches de collecte, de traitement, de synthèse et de raisonnement. LangGraphont intégré ces agents via des API dans leurs systèmes cloud.

CrewAI

CrewAIest un framework open source spécifiquement axé sur l'orchestration multi-agents autonome, disponible sur. GitHub Il propose une approche structurée pour créer des équipes d'agents autonomes spécialisés qui collaborent pour résoudre des tâches complexes sans intervention humaine. CrewAImet l'accent sur la coordination basée sur les rôles et la délégation des tâches.

Principales fonctionnalités de CrewAI

CrewAIfournit les fonctionnalités clés suivantes :

- Role-based conception d'agents — Les agents autonomes sont définis avec des rôles, des objectifs et des histoires spécifiques afin de permettre une expertise spécialisée. Pour plus d'informations, consultez la section Création d'agents efficaces dans la CrewAI documentation.
- Délégation de tâches : Built-in mécanismes permettant d'attribuer des tâches de manière autonome aux agents appropriés en fonction de leurs capacités. Pour plus d'informations, consultez la section Tâches de la CrewAI documentation.
- Collaboration entre agents : cadre pour la communication autonome entre agents et le partage des connaissances sans médiation humaine. Pour plus d'informations, consultez la section Collaboration dans la CrewAI documentation.
- Gestion des processus : flux de travail structurés pour l'exécution séquentielle et parallèle de tâches autonomes. Pour plus d'informations, consultez la section Processus dans la CrewAI documentation.
- Sélection du modèle de base — Support de différents modèles de base, notamment les modèles Anthropic Claude, Amazon Nova (Premier, Pro, Lite et Micro) sur Amazon Bedrock, et d'autres pour optimiser les différentes tâches de raisonnement autonomes. Pour plus d'informations, consultez la section LLM dans la CrewAI documentation.
- Intégration de l'API LLM — Intégration flexible avec plusieurs interfaces de service LLM, notamment Amazon BedrockOpenAI, et les déploiements de modèles locaux. Pour plus

d'informations, consultez les exemples de configuration des fournisseurs dans la CrewAI documentation.

- Support multimodal — Capacités émergentes de gestion du texte, des images et d'autres modalités pour des interactions complètes avec les agents autonomes. Pour plus d'informations, consultez la section Utilisation d'agents multimodaux dans la CrewAI documentation.

Quand utiliser CrewAI

CrewAIest particulièrement bien adapté aux scénarios d'agents autonomes, notamment :

- Problèmes complexes bénéficiant d'une expertise spécialisée basée sur les rôles travaillant de manière autonome
- Projets nécessitant une collaboration explicite entre plusieurs agents autonomes
- Cas d'utilisation où la décomposition des problèmes en équipe améliore la résolution autonome des problèmes
- Scénarios nécessitant une séparation claire des préoccupations entre les différents rôles d'agent autonome

Approche de mise en œuvre pour CrewAI

CrewAIfournit une implémentation basée sur les rôles de l'approche des équipes d'agents IA pour les parties prenantes de l'entreprise, comme indiqué dans la section Getting Started de la CrewAI documentation. Le cadre permet aux organisations de :

- Définissez des agents autonomes spécialisés dotés de rôles, d'objectifs et de domaines d'expertise spécifiques.
- Attribuez des tâches aux agents en fonction de leurs capacités spécialisées.
- Établissez des dépendances claires entre les tâches pour créer des flux de travail structurés.
- Orchestrez la collaboration entre plusieurs agents pour résoudre des problèmes complexes.

Cette approche basée sur les rôles reflète les structures humaines des équipes, ce qui la rend intuitive à comprendre et à mettre en œuvre pour les chefs d'entreprise. Organisations peuvent créer des équipes autonomes dotées de domaines d'expertise spécialisés qui collaborent pour atteindre leurs objectifs commerciaux, de la même manière que les équipes humaines fonctionnent. Cependant, l'équipe autonome peut travailler en continu sans intervention humaine.

Real-world exemple de CrewAI

AWS a mis en œuvre des systèmes multi-agents autonomes utilisant CrewAI intégré à Amazon Bedrock, comme indiqué dans l'CrewAIétude de cas publiée. AWS et CrewAI a développé un cadre sécurisé et indépendant du fournisseur. L'architecture CrewAI open source « flows-and-crews » s'intègre parfaitement aux modèles de base, aux systèmes de mémoire et aux dispositifs de conformité d'Amazon Bedrock.

Les principaux éléments de la mise en œuvre sont les suivants :

- Des plans et des sources ouvertes, AWS ainsi que des modèles de référence CrewAI publiés qui associent les CrewAI agents aux modèles et aux outils d'observabilité d'Amazon Bedrock. Ils ont également publié des systèmes exemplaires tels qu'une équipe d'audit de AWS sécurité multi-agents, des flux de modernisation du code et l'automatisation du back-office des biens de consommation (CPG).
- Intégration de la pile d'observabilité : la solution intègre la surveillance avec Amazon CloudWatch et permet la traçabilité et le débogageLangFuse, de la validation du concept à la production. AgentOps
- Retour sur investissement (ROI) démontré — Les premiers projets pilotes présentent des améliorations majeures : exécution 70 % plus rapide pour un projet de modernisation du code de grande envergure et réduction d'environ 90 % du temps de traitement pour un flux de backoffice CPG.

AutoGen

AutoGenest un framework open source initialement publié parMicrosoft. AutoGense concentre sur la mise en place d'agents IA autonomes conversationnels et collaboratifs. Il fournit une architecture flexible pour créer des systèmes multi-agents en mettant l'accent sur les interactions asynchrones et pilotées par des événements entre les agents pour des flux de travail autonomes complexes.

Principales fonctionnalités de AutoGen

AutoGenfournit les fonctionnalités clés suivantes :

- Agents conversationnels : conçus autour de conversations en langage naturel entre agents autonomes, permettant un raisonnement sophistiqué par le biais du dialogue. Pour plus d'informations, consultez Multi-agent Conversation Framework dans la AutoGen documentation.

- Architecture asynchrone : Event-driven conçue pour des interactions non bloquantes avec des agents autonomes, prenant en charge des flux de travail parallèles complexes. Pour plus d'informations, consultez la section Résolution de plusieurs tâches dans une séquence de discussions asynchrones dans la AutoGen documentation.
- Human-in-the-loop— Soutien solide à la participation humaine facultative dans les flux de travail des agents par ailleurs autonomes en cas de besoin. Pour plus d'informations, consultez la section Autorisation du feedback humain dans les agents dans la AutoGen documentation.
- Génération et exécution de code — Fonctionnalités spécialisées pour les agents autonomes axés sur le code qui peuvent écrire et exécuter du code. Pour plus d'informations, consultez la section Exécution du code dans la AutoGen documentation.
- Comportements personnalisables — Configuration flexible des agents autonomes et contrôle des conversations pour divers cas d'utilisation. Pour plus d'informations, consultez agentchat.conversable_agent dans la documentation. AutoGen
- Sélection du modèle de base — Support pour différents modèles de base, notamment les modèles Anthropic Claude, Amazon Nova (Premier, Pro, Lite et Micro) sur Amazon Bedrock, et d'autres pour différentes capacités de raisonnement autonome. Pour plus d'informations, consultez la section Configuration LLM dans la AutoGen documentation.
- Intégration de l'API LLM — Configuration standardisée pour plusieurs interfaces de service LLM, notamment Amazon Bedrock etOpenAI. Azure OpenAI Pour plus d'informations, consultez oai.openai_utils dans le manuel de référence des API. AutoGen
- Traitement multimodal — Support du traitement du texte et de l'image pour permettre des interactions multimodales riches entre agents autonomes. Pour plus d'informations, consultez la section Interaction avec les modèles multimodaux : GPT-4V AutoGen dans la AutoGen documentation.

Quand utiliser AutoGen

AutoGenest particulièrement bien adapté aux scénarios d'agents autonomes, notamment :

- Applications qui nécessitent des flux conversationnels naturels entre agents autonomes pour un raisonnement complexe
- Projets nécessitant à la fois un fonctionnement totalement autonome et des capacités de supervision humaine optionnelles
- Cas d'utilisation impliquant la génération, l'exécution et le débogage de code autonomes sans intervention humaine

- Scénarios nécessitant des modèles de communication flexibles et asynchrones avec des agents autonomes

Approche de mise en œuvre pour AutoGen

AutoGenpropose une approche de mise en œuvre conversationnelle pour les parties prenantes de l'entreprise, comme indiqué dans la section Getting Started de la AutoGen documentation. Le cadre permet aux organisations de :

- Créez des agents autonomes qui communiquent par le biais de conversations en langage naturel.
- Implémentez des interactions asynchrones pilotées par des événements entre plusieurs agents.
- Combinez un fonctionnement entièrement autonome avec une supervision humaine optionnelle en cas de besoin.
- Développez des agents spécialisés pour différentes fonctions commerciales qui collaborent par le biais du dialogue.

Cette approche conversationnelle rend le raisonnement du système autonome transparent et accessible aux utilisateurs professionnels. Decision-makers peut observer le dialogue entre les agents pour comprendre comment les conclusions sont tirées et éventuellement participer à la conversation lorsque le jugement humain est requis.

Real-world exemple de AutoGen

Magentic-Oneest un système multi-agents généraliste open source conçu pour résoudre de manière autonome des tâches complexes en plusieurs étapes dans divers environnements, comme décrit dans le blog AI Frontiers. Microsoft À la base se trouve l'agent Orchestrator, qui décompose les objectifs de haut niveau et suit les progrès à l'aide de registres structurés. Cet agent délègue des sous-tâches à des agents spécialisés (tels queWebSurfer, FileSurferCoder, etComputerTerminal) et s'adapte dynamiquement en replanifiant si nécessaire.

Le système repose sur le AutoGen framework et est indépendant du modèle, avec GPT-4o par défaut. Il atteint des performances de pointe sur des critères tels GAIA que, et, le tout sans réglage spécifique à une tâche. AssistantBench WebArena De plus, il prend en charge l'extensibilité modulaire et une évaluation rigoureuse par le biais de AutoGenBench suggestions.

LlamaIndex

LlamaIndexest un framework de données conçu spécifiquement pour connecter de grands modèles linguistiques (LLM) à des sources de données externes afin de permettre des applications sophistiquées de génération augmentée de récupération (RAG) et d'intelligence artificielle agentique. Le framework fournit des abstractions et des flux de développement accélérés pour les systèmes agentiques, des modèles d'orchestration personnalisés et des intégrations de systèmes qui réduisent les délais de production des solutions d'IA axées sur les connaissances.

Principales fonctionnalités de LlamaIndex

LlamaIndexfournit un ensemble complet de fonctionnalités qui le rendent particulièrement adapté aux applications d'intelligence artificielle agentiques d'entreprise :

- Data-centric architecture : excelle dans l'ingestion, l'indexation et la récupération d'informations provenant de plus de 100 formats de données, notamment des PDF, des documents Microsoft Word, des feuilles de calcul, etc. Le framework transforme les données d'entreprise en bases de connaissances consultables optimisées pour les agents d'intelligence artificielle. Pour plus d'informations, consultez la documentation LlamaIndex.
- Production-ready déploiement : LlamaIndex propose à la fois des frameworks open source et des services gérésLlamaCloud, fournissant des fonctionnalités de niveau entreprise, notamment des contrôles de sécurité, une évolutivité, des intégrations d'observabilité et une flexibilité de déploiement. Pour plus d'informations, consultez la documentation du LlamaIndex framework.
- Traitement avancé des documents : LlamaCloud fournit des fonctionnalités d'analyse, d'extraction, d'indexation et de récupération de documents qui gèrent les mises en page complexes, les tableaux imbriqués, le contenu multimodal et même les notes manuscrites. Cette analyse sophistiquée permet aux agents de travailler efficacement avec des documents d'entreprise réels contenant des graphiques, des diagrammes et des mises en forme complexes. Pour plus d'informations, consultez la documentation LlamaCloud.
- Orchestration des flux de travail : LlamaAgents fournit un moteur d'orchestration asynchrone piloté par les événements pour créer des systèmes agentiques en plusieurs étapes. Les flux de travail prennent en charge des modèles complexes tels que les boucles, l'exécution parallèle, le branchement conditionnel et la reprise dynamique, ce qui les rend idéaux pour les interactions sophistiquées avec les agents. Pour plus d'informations, consultez la documentation sur les LlamaIndex flux de travail.

- Capacités de récupération agentique : modes de récupération avancés, notamment la recherche hybride, la recherche sémantique et le routage automatique, qui déterminent intelligemment la meilleure stratégie de récupération pour chaque requête. Le framework prend en charge la récupération composite dans plusieurs bases de connaissances avec un reclassement pour une précision accrue. Pour plus d'informations, consultez la documentation LlamaIndex RAG.
- Observabilité et évaluation : LlamaIndex s'intègre à une variété d'outils d'observabilité et d'évaluation. Cette fonctionnalité d'intégration vous permet de suivre et de déboguer vos applications, d'évaluer leurs performances et de surveiller les coûts. Pour plus d'informations, consultez la LlamaIndex documentation relative au suivi, au débogage et à l'évaluation.

Quand utiliser LlamaIndex

LlamaIndexest particulièrement bien adapté aux scénarios d'IA agentique qui mettent l'accent sur les flux de travail gourmands en données et la gestion des connaissances :

- Document-heavy applications qui nécessitent que les agents traitent, analysent et extraient des informations à partir de grands volumes de documents d'entreprise tels que des contrats, des rapports, des manuels et des documents réglementaires
- Scénarios du prototypage rapide à la production dans lesquels les entreprises souhaitent créer et déployer rapidement des agents centrés sur les documents sans surcharger la gestion de l'infrastructure
- RAG-first des architectures qui privilégient la précision de l'extraction et la pertinence du contexte, en particulier lorsque vous travaillez avec des documents multimodaux complexes contenant des tableaux, des images et des données structurées
- Multi-agent flux de travail documentaires qui nécessitent des agents spécialisés pour différents aspects du traitement des documents, tels que l'analyse, la synthèse et le contrôle de conformité

Approche de mise en œuvre pour LlamaIndex

LlamaIndex fournit à la fois des éléments de base de base et des abstractions de haut niveau qui s'adaptent à différentes approches de mise en œuvre :

- Développement rapide d'applications RAG fonctionnelles en quelques lignes de code à l'aide d'API de LlamaIndex haut niveau. Cette approche rend LlamaIndex accessible aux équipes commerciales et aux développeurs novices en matière d'IA agentic.

- Intégration d'entreprise via LlamaHub les systèmes d'entreprise les plus courants SharePoint, notamment Amazon Simple Storage Service (Amazon S3), les bases de données et les API. Cette approche permet une intégration parfaite avec l'infrastructure de données existante.
- Des options de déploiement flexibles entre des déploiements open source auto-hébergés pour un contrôle maximal ou des services LlamaCloud gérés pour réduire les frais d'exploitation et les fonctionnalités d'entreprise.
- Les applications peuvent commencer par de simples moteurs de requêtes et ajouter progressivement des fonctionnalités agentiques, une orchestration multi-agents et des flux de travail complexes au fur et à mesure de l'évolution des exigences.

Real-world exemple de LlamaIndex

Cet exemple porte sur une filiale d'une entreprise aérospatiale spécialisée dans les solutions de navigation et d'exploitation aériennes. Ils doivent relever un défi croissant qui consiste à piloter des essais de chatbots basés sur l'IA non coordonnés. Les essais ont donné lieu à des travaux répétés, à de longs cycles de développement, à des obstacles à la conformité et à des mises en œuvre isolées au sein de l'organisation.

Ils ont développé un framework d'agents unifié, une solution réutilisable basée sur des modèles basée sur le framework LlamaIndex open source qui rend la création d'agents beaucoup plus efficace. Ils ont comparé plusieurs frameworks concurrents, à la fois orientés chaîne et basés sur des graphes. En fin de compte, ils ont opté LlamaIndex pour trois avantages essentiels : sa conception flexible, ses composants modulaires et ses commandes d'orchestration prêtes pour la production.

La plate-forme réduit le temps de développement et de déploiement des agents de 87 %, passant de 512 à 64 heures. Cette réduction a été réalisée en permettant aux équipes de créer des agents avec environ 50 lignes de code et un fichier de configuration JSON. Les équipes ont tiré parti d'un cadre unifié intégrant des fonctionnalités de sécurité, de conformité et d'accès privilégié au système. Pour plus de détails, consultez les études de cas LlamaIndex clients.

Comparaison des frameworks d'IA agentiques

Lorsque vous sélectionnez un framework d'IA agentic pour le développement d'agents autonomes, réfléchissez à la manière dont chaque option correspond à vos besoins spécifiques. Tenez compte non seulement de ses capacités techniques, mais également de son adéquation organisationnelle, notamment de l'expertise de l'équipe, de l'infrastructure existante et des exigences de maintenance

à long terme. De nombreuses organisations pourraient bénéficier d'une approche hybride, en tirant parti de plusieurs cadres pour les différents composants de leur écosystème d'IA autonome.

Le tableau suivant compare les niveaux de maturité (le plus fort, le plus fort, le plus adéquat ou le plus faible) de chaque framework selon les principales dimensions techniques. Pour chaque framework, le tableau inclut également des informations sur les options de déploiement en production et la complexité de la courbe d'apprentissage.

Cadre	AWS intégration	Support multi-agents autonome	Complexité du workflow autonome	Capacités multimodales	Sélection du modèle de fondation	Intégration de l'API LLM	Déploiement en production	Courbe d'apprentissage
AutoGen	Faible	Solide	Solide	Suffisant	Suffisant	Solide	Faites-le vous-même (DIY)	Raide
CrewAI	Faible	Solide	Suffisant	Faible	Suffisant	Suffisant	DIY	Modérée
LangChain / LangGraph	Suffisant	Solide	Le plus fort	Le plus fort	Le plus fort	Le plus fort	Plateforme ou bricolage	Raide
LlamaIndex	Suffisant	Suffisant	Solide	Suffisant	Solide	Solide	Plateforme ou bricolage	Modérée
Strands Agents	Le plus fort	Solide	Le plus fort	Solide	Solide	Le plus fort	DIY	Modérée

Considérations à prendre en compte lors du choix d'un framework d'IA agentic

Lorsque vous développez des agents autonomes, tenez compte des facteurs clés suivants :

- AWS intégration de l'infrastructure — Les organisations fortement investies AWS bénéficieront le plus des intégrations natives de Strands Agents with Services AWS pour les flux de travail autonomes. Pour plus d'informations, consultez le résuméAWS hebdomadaire (AWS blog).
- Sélection du modèle de base : déterminez quel framework fournit le meilleur support pour vos modèles de base préférés (par exemple, les modèles Amazon Nova sur Amazon Bedrock ou Anthropic Claude), en fonction des exigences de raisonnement de votre agent autonome. Pour plus d'informations, consultez la section Création d'agents efficaces sur le Anthropic site Web.
- Intégration de l'API LLM : évaluez les frameworks en fonction de leur intégration avec vos interfaces de service LLM (Large Language Model) préférées (par exemple, Amazon Bedrock ouOpenAI) pour le déploiement en production. Pour plus d'informations, consultez la section Model Interfaces dans la Strands Agents documentation.
- Exigences multimodales — Pour les agents autonomes qui ont besoin de traiter du texte, des images et de la parole, tenez compte des capacités multimodales de chaque framework. Pour plus d'informations, consultez la section Multimodalité dans la LangChain documentation.
- Complexité des flux de travail autonomes — Des flux de travail autonomes plus complexes dotés d'une gestion d'état sophistiquée peuvent favoriser les capacités avancées des machines à états. LangGraph
- Collaboration d'équipe autonome — Les projets qui nécessitent une collaboration autonome explicite basée sur les rôles entre des agents spécialisés peuvent bénéficier de l'architecture orientée équipe de. CrewAI
- Paradigme de développement autonome — Les équipes qui préfèrent les modèles conversationnels et asynchrones pour les agents autonomes peuvent préférer l'architecture événementielle de. AutoGen
- Approche gérée ou basée sur le code — Les organisations qui souhaitent une expérience entièrement gérée avec un minimum de codage devraient envisager Amazon Bedrock Agents. Organisations nécessitant une personnalisation plus poussée peuvent Strands Agents préférer d'autres frameworks dotés de fonctionnalités spécialisées qui répondent mieux aux exigences spécifiques des agents autonomes.
- Préparation à la production pour les systèmes autonomes : considérez les options de déploiement, les capacités de surveillance et les fonctionnalités d'entreprise pour les agents autonomes de production.

Plateformes

Les plateformes Agentic AI fournissent les couches d'exécution, d'orchestration et d'intégration de base nécessaires au déploiement, à la mise à l'échelle et à la gestion des systèmes agentic de niveau production. Les frameworks définissent la manière dont les agents sont créés et les protocoles régissent leur mode de communication. Les plateformes fournissent l'environnement dans lequel ces agents opèrent, collaborent et évoluent en toute sécurité à grande échelle.

Les plateformes Agentic combinent des fonctionnalités d'exécution de modèles, de gestion du contexte, d'intégration d'outils, d'observabilité et de gouvernance dans des environnements unifiés. Ces plateformes permettent aux entreprises de passer de l'expérimentation au déploiement à l'échelle de l'entreprise.

Dans cette section :

Pourquoi les plateformes sont importantes

Les plateformes d'IA agentic sont essentielles pour les organisations qui cherchent à opérationnaliser des systèmes autonomes en production. Ils offrent les fonctionnalités suivantes :

- Fournissez une orchestration d'exécution pour l'hébergement, le dimensionnement et la coordination des agents.
- Gérez l'état, le contexte et la mémoire dans les flux de travail multi-agents.
- Proposez des contrôles de sécurité, d'identité et de gouvernance conformes aux normes de l'entreprise.
- Intégrez les écosystèmes d'outillage et les systèmes externes par le biais de normes APIs ou de protocoles.
- Favorisez l'observabilité et l'auditabilité des interactions avec les agents et des flux d'événements.

- Support de l'interopérabilité entre modèles, permettant aux agents d'utiliser plusieurs modèles de base dans un seul environnement.

Ces fonctionnalités transforment les agents individuels en systèmes coordonnés et adaptatifs capables de fonctionner de manière fiable dans les limites de l'entreprise et des réglementations.

Types de plateformes d'IA agentique

Les plateformes d'IA agentic appartiennent généralement à une ou plusieurs des catégories suivantes :

- Agent géré : les plateformes entièrement gérées fournissent une infrastructure, une mémoire et des capacités d'orchestration intégrées. Ils réduisent les frais d'exploitation et accélèrent le délai de production.
- Orchestration open source — Les plateformes agentiques open source offrent flexibilité et transparence aux entreprises qui préfèrent des environnements personnalisables ou un déploiement sur site.
- Entreprise hybride : les plateformes hybrides intègrent des composants gérés et auto-hébergés, alliant l'évolutivité des services gérés dans le cloud au contrôle des systèmes d'entreprise.

Considérations relatives au choix des plateformes

Lors de la sélection ou de la conception d'une plateforme d'IA agentic, les organisations doivent prendre en compte les points suivants :

- Profondeur de l'intégration : évaluez dans quelle mesure la plateforme s'intègre aux sources de données, aux outils et aux protocoles existants.
- Évolutivité — Assurez-vous que la plateforme peut évoluer de manière dynamique pour prendre en charge les charges de travail autonomes et la collaboration entre plusieurs agents.
- Sécurité et conformité : évaluez les fonctionnalités de confidentialité, de chiffrement et de gouvernance des données par rapport aux exigences organisationnelles et régionales.
- Extensibilité — Choisissez des plateformes dotées d'architectures modulaires qui permettent d'ajouter de nouveaux outils, modèles ou agents au fil du temps.
- Observabilité — Préférez les plateformes qui fournissent une télémétrie détaillée, une traçabilité et des journaux d'audit pour les interactions agentiques.

- Rentabilité — Envisagez des modèles sans serveur ou basés sur l'utilisation afin d'optimiser le coût des charges de travail variables.

Agents Amazon Bedrock

Amazon Bedrock Agents est un service entièrement géré qui vous permet de créer et de configurer des agents autonomes dans vos applications. Il peut orchestrer les interactions entre les modèles de base, les sources de données, les applications logicielles et les conversations avec les utilisateurs. Son approche rationalisée de création d'agents ne vous oblige pas à fournir de la capacité, à gérer l'infrastructure ou à écrire du code personnalisé.

Principales fonctionnalités d'Amazon Bedrock Agents

Amazon Bedrock Agents inclut les fonctionnalités clés suivantes :

- Service entièrement géré : gestion complète de l'infrastructure sans qu'il soit nécessaire de fournir de la capacité ou de gérer les systèmes sous-jacents. Pour plus d'informations, consultez Automatiser les tâches de votre application à l'aide d'agents d'intelligence artificielle dans la documentation Amazon Bedrock.
- Développement piloté par API : définissez et exécutez des agents via de simples appels d'API en spécifiant des modèles, des instructions, des outils et des paramètres de configuration. Pour plus d'informations, consultez la section Créer et configurer un agent manuellement dans la documentation Amazon Bedrock.
- Groupes d'actions : définissez les actions spécifiques que votre agent peut effectuer en créant des groupes d'actions avec des schémas d'API. Pour plus d'informations, consultez la section Utiliser des groupes d'actions pour définir les actions que votre agent doit effectuer dans la documentation Amazon Bedrock.
- Intégration aux bases de connaissances — Connectez-vous facilement aux bases de connaissances Amazon Bedrock pour augmenter les réponses des agents grâce aux données de votre organisation. Pour plus d'informations, consultez Augmentez la génération de réponses pour votre agent grâce à la base de connaissances de la documentation Amazon Bedrock.
- Modèles d'invite avancés : personnalisez le comportement des agents grâce à des modèles d'invite pour le prétraitement, l'orchestration, la génération de réponses dans la base de connaissances et le post-traitement. Pour plus d'informations, consultez la section Améliorer la précision de l'agent à l'aide de modèles d'invite avancés dans Amazon Bedrock dans la documentation Amazon Bedrock.

- Traçabilité et observabilité : suivez le processus de step-by-step raisonnement de l'agent à l'aide des fonctionnalités de suivi intégrées. Pour plus d'informations, consultez la section Suivi du processus de step-by-step raisonnement de l'agent à l'aide de trace dans la documentation Amazon Bedrock.
- Gestion des versions et alias : créez plusieurs versions de votre agent et déployez-les via des alias pour des déploiements contrôlés. Pour plus d'informations, consultez la section Déployer et utiliser un agent Amazon Bedrock dans votre application dans la documentation Amazon Bedrock.

Quand utiliser Amazon Bedrock Agents

Amazon Bedrock Agents est particulièrement adapté aux scénarios d'agents autonomes, notamment :

- Organisations qui souhaitent bénéficier d'une expérience entièrement gérée pour créer et déployer des agents sans gérer l'infrastructure
- Projets nécessitant un développement et un déploiement rapides d'agents par le biais de la configuration plutôt que du code
- Cas d'utilisation bénéficiant d'une intégration étroite avec d'autres fonctionnalités d'Amazon Bedrock, telles que les bases de connaissances et les garde-corps
- Des équipes qui ne disposent pas des ressources internes nécessaires pour créer des agents à partir de zéro, mais qui ont besoin de capacités autonomes prêtes à être mises en production

Approche de mise en œuvre pour Amazon Bedrock Agents

Amazon Bedrock Agents propose une approche de mise en œuvre basée sur la configuration pour les parties prenantes de l'entreprise. Le service permet aux organisations de :

- Définissez les agents via les appels d'API Console de gestion AWS or sans écrire de code complexe.
- Créez des groupes d'actions qui spécifient les opérations APIs et que l'agent peut effectuer.
- Connectez les bases de connaissances pour fournir des informations spécifiques au domaine à l'agent.
- Testez et répétez le comportement des agents via une interface visuelle.

Cette approche gérée permet aux équipes commerciales de développer et de déployer rapidement des agents autonomes sans expertise technique approfondie en matière de développement de modèles d'IA ou de gestion d'infrastructure.

Exemple concret d'Amazon Bedrock Agents

Une solution d'opérations financières (FinOps) décrite dans ce billet de AWS blog utilise le framework multi-agents Amazon Bedrock pour créer un assistant de gestion des coûts dans le cloud piloté par l'IA. Le modèle économique de base d'Amazon Nova alimente la solution dans le cadre de laquelle un agent FinOps superviseur central délègue des tâches à des agents spécialisés. Ces agents récupèrent et analysent les données de AWS dépenses en utilisant AWS Cost Explorer et génèrent des recommandations de réduction des coûts en utilisant. AWS Trusted Advisor

Le système inclut un accès utilisateur sécurisé via Amazon Cognito, une interface hébergée sur AWS Amplify, et des groupes d' AWS Lambda action pour des analyses et des prévisions en temps réel. Les équipes financières peuvent poser des questions en langage naturel telles que « Quels étaient mes coûts en février 2025 ? » Le système répond par des ventilations détaillées, des suggestions d'optimisation et des prévisions, le tout dans le cadre d'une architecture évolutive et sans serveur déployée par l'utilisateur. AWS CloudFormation

Amazon Bedrock AgentCore

Amazon Bedrock AgentCore est une plateforme agentique permettant de créer, de déployer et d'exploiter des agents hautement performants en toute sécurité et à grande échelle, en utilisant n'importe quel framework, modèle ou protocole. En utilisant AgentCore, vous pouvez effectuer les opérations suivantes, le tout sans aucune gestion d'infrastructure :

- Créez des agents plus rapidement.
- Permettez aux agents de prendre des mesures sur l'ensemble des outils et des données.
- Exécutez les agents en toute sécurité avec une faible latence et des durées d'exécution prolongées.
- Surveillez les agents en production.

AgentCore élimine le fardeau indifférencié lié à la mise en place d'une infrastructure d'agents spécialisés, ce qui vous permet d'accélérer la mise en production de vos agents. Ses services peuvent être utilisés ensemble ou indépendamment et sont compatibles avec n'importe quel framework, y compris CrewAILangGraph,LlamaIndex, etStrands Agents. AgentCoreest également

compatible avec tous les modèles de fondations disponibles dans ou en dehors d'Amazon Bedrock, offrant ainsi une flexibilité ultime.

AgentCore est composé de plusieurs services clés :

- Amazon Bedrock AgentCore Runtime — Fournit un environnement sécurisé, évolutif et sans serveur pour héberger et exécuter vos agents, sans avoir à gérer l'infrastructure requise pour le déploiement et l'exécution d'agents ou d'outils d'IA.
- Amazon Bedrock AgentCore Memory : propose un système de mémoire géré, permettant aux agents de conserver le contexte des interactions pour des conversations plus personnalisées et cohérentes en conservant des connaissances à la fois immédiates et à long terme.
- Amazon Bedrock AgentCore Gateway — Simplifie le processus de création, de sécurisation et de recherche des bons outils pour les agents. Avec AgentCore Gateway, les développeurs peuvent convertir APIs les fonctions Lambda et les services existants en outils compatibles avec le Model Context Protocol (MCP) et les mettre à la disposition des agents.
- Amazon Bedrock AgentCore Identity — Fournit un service de gestion des identités et des accès des agents sécurisé et évolutif qui accélère le développement d'agents IA. Avec AgentCore Identity, vous pouvez attribuer des identités uniques et vérifiables aux agents, ce qui permet un contrôle d'accès précis et des interactions sécurisées pilotées par les agents avec les systèmes de l'entreprise.
- Outils AgentCore intégrés Amazon Bedrock : vous permet d'utiliser des outils intégrés pour améliorer votre flux de travail de développement et de test. Utilisez ces outils pour interagir efficacement avec votre application, en permettant aux agents d'intelligence artificielle d'écrire et d'exécuter du code en toute sécurité dans des environnements sandbox. Utilisez l'outil de navigation pour permettre aux agents d'intelligence artificielle d'interagir avec des sites Web à grande échelle.
- Amazon Bedrock AgentCore Observability : fournit des fonctionnalités de journalisation et de surveillance, vous donnant une visibilité en temps réel sur les performances et le comportement de votre agent afin de faciliter le débogage et l'optimisation.

Principales fonctionnalités de AgentCore

AgentCore inclut les principales fonctionnalités suivantes :

- Entièrement géré et extensible : AgentCore il s'agit d'un service entièrement géré, ce qui signifie qu'il AWS gère l'infrastructure et la maintenance sous-jacentes. Il est également extensible, ce

qui vous permet de personnaliser et d'améliorer les fonctionnalités de vos agents. Pour plus d'informations, voir Commencer avec AgentCore Runtime dans la AgentCore documentation.

- Mémoire à long terme et à court terme : offrez des interactions plus personnalisées et pertinentes en dotant les agents d'un système de mémoire capable de mémoriser le contexte des conversations en cours et des connaissances à long terme. Pour plus d'informations, voir Commencer avec AgentCore la mémoire dans la AgentCore documentation.
- Développement et intégration d'outils simplifiés : permettez à vos agents de découvrir et d'utiliser des outils via un point de terminaison unique et sécurisé. Transformez rapidement les ressources existantes de votre entreprise en outils prêts à être utilisés par les agents en quelques lignes de code, ce qui permet aux développeurs de se concentrer sur le développement de fonctionnalités uniques. Pour plus d'informations, voir Commencer avec AgentCore Gateway dans la AgentCore documentation.
- Infrastructure sécurisée et évolutive : AgentCore fournit un environnement sécurisé et évolutif pour le déploiement et l'exploitation des agents. Il inclut des fonctionnalités de gestion des identités et des accès, de chiffrement des données et de sécurité du réseau. Pour plus d'informations, voir Commencer avec AgentCore Identity dans la AgentCore documentation.
- Intégration à un large éventail d'outils : vous permet d'intégrer à vos agents une variété d'outils, notamment un interpréteur de code et un outil de navigateur que vous pouvez créer à l'aide des outils AgentCore intégrés. Pour plus d'informations, voir Commencer avec l'interpréteur de AgentCore code et Commencer avec le AgentCore navigateur dans la AgentCore documentation.
- Observabilité et surveillance complètes : bénéficiez d'une visibilité approfondie sur vos agents grâce à des outils complets permettant de suivre, de déboguer et de surveiller leurs performances en production. Visualisez le parcours d'exécution complet de l'agent pour auditer son raisonnement et résoudre les défaillances. Utilisez des tableaux de bord en temps réel et des données de télémétrie standardisées pour suivre les indicateurs opérationnels clés. Pour plus d'informations, consultez la section Ajouter de l'observabilité à vos AgentCore ressources Amazon Bedrock dans la AgentCore documentation.

Quand utiliser AgentCore

AgentCore est particulièrement bien adapté aux scénarios d'agents autonomes, notamment :

- Organisations qui souhaitent accélérer le développement et réduire les frais opérationnels grâce à un service entièrement géré qui gère l'infrastructure, la sécurité, les outils intégrés, l'observabilité et la mise à l'échelle

- Projets nécessitant de la flexibilité avec des services modulaires qui fonctionnent ensemble ou indépendamment et sont compatibles avec n'importe quel framework, similaire CrewAI ouLangGraph, et avec n'importe quel modèle de base, quelle que soit la source
- Cas d'utilisation nécessitant des agents conversationnels dynamiques qui doivent conserver le contexte et tirer des leçons des interactions passées pour fournir des réponses personnalisées et pertinentes
- Les agents sont en mesure d'effectuer des tâches complexes grâce à une intégration simple à diverses applications, sources de données et APIs

Approche de mise en œuvre pour AgentCore

AgentCore est conçu pour les organisations qui souhaitent faire passer les agents d'intelligence artificielle de la preuve de concept, construite à l'aide de frameworks d'agents open source ou personnalisés, à la production. Grâce à AgentCore ces outils, les organisations peuvent effectuer les opérations suivantes :

- Déployez des agents en toute sécurité sur une infrastructure sans serveur, compatible avec n'importe quel framework et modèle, avec isolation des sessions et gestion intégrée des identités et des accès pour garantir la end-to-end sécurité et la conformité. Créez rapidement des agents AgentCore Runtime pour les principaux frameworks d'agents à l'aide du kit de démarrage.
- Améliorez les agents en intégrant une mémoire persistante pour la rétention du contexte, en simplifiant le développement et l'intégration des outils via AgentCore Gateway. Tirez parti de l'outil de navigateur intégré et de l'interpréteur de code pour des flux de travail avancés.
- Suivez, déboguez et surveillez les agents d'IA en production à l'aide de tableaux de bord d'observabilité optimisés par Amazon CloudWatch Application Insights et OpenTelemetry en suivant les indicateurs clés des AgentCore ressources (temps d'exécution, mémoire, passerelle et outils).
- Accélérez le déploiement et l'innovation avec des services modulaires entièrement gérés, des blocs composables ensemble ou indépendamment, avec n'importe quel framework d'agents et fournisseur de modèles. Cette flexibilité permet aux entreprises de passer plus rapidement du prototype à la production.

Cette approche gérée permet aux entreprises de créer, déployer et exécuter rapidement et en toute sécurité des agents d'intelligence artificielle et des systèmes multi-agents de qualité professionnelle à n'importe quelle échelle.

Exemple concret de AgentCore

AWS a observé que l'une des plus grandes banques d'Amérique latine propose AI/ML depuis des années une expérience bancaire numérique hyperpersonnalisée et sécurisée. La banque développe les services d'intelligence artificielle agentic en les utilisant AgentCore pour fournir aux clients des interactions intuitives, une sécurité renforcée et une automatisation accrue. Selon le CTO, AgentCore il devrait soutenir leurs efforts pour respecter les engagements des clients à grande échelle. AgentCore fournit à leurs développeurs les outils et la flexibilité nécessaires pour créer et gérer des agents, tout en garantissant le respect des réglementations financières.

Protocoles

Les agents d'intelligence artificielle ont besoin de protocoles de communication standardisés pour interagir avec les autres agents et services. Organisations qui mettent en œuvre des architectures d'agents sont confrontées à des défis importants en matière d'interopérabilité, d'indépendance des fournisseurs et de pérennisation de leurs investissements.

Cette section vous aide à naviguer dans le paysage des agent-to-agent protocoles en mettant l'accent sur les normes ouvertes qui maximisent la flexibilité et l'interopérabilité. (Pour plus d'informations sur agent-to-tool les protocoles, voir Stratégie d'intégration des outils plus loin dans ce guide.)

Cette section met en évidence le Model Context Protocol (MCP), un standard ouvert initialement développé Anthropic en 2024. Aujourd'hui, soutient AWS activement le MCP en contribuant au développement et à la mise en œuvre du protocole. AWS collabore avec les principaux frameworks d'agents open source, notammentLangGraph, et CrewAILlamaIndex, pour façonner le futur de la communication inter-agents sur le protocole. Pour plus d'informations, voir Protocoles ouverts pour l'interopérabilité des agents, partie 1 : Communication entre agents sur MCP (AWS blog).

Dans cette section :

- Pourquoi le choix du protocole est important
- Agent-to-agent protocoles
- Sélection de protocoles agentiques
- Stratégie de mise en œuvre des protocoles agentiques
- Commencer à utiliser MCP
- ???

Pourquoi le choix du protocole est important

La sélection du protocole façonne fondamentalement la manière dont vous pouvez créer et faire évoluer votre architecture d'agent d'IA. En choisissant des protocoles garantissant la portabilité entre les frameworks d'agents, vous bénéficiez de la flexibilité nécessaire pour combiner différents systèmes d'agents et flux de travail pour répondre à vos besoins spécifiques.

Les protocoles ouverts vous permettent d'intégrer des agents dans plusieurs frameworks. Par exemple, utilisez-le LangChain pour le prototypage rapide et implémentez des systèmes de production communiquant via un protocole commun, tel que MCP ou le protocole Agent2Agent (A2A). Strands Agents Cette flexibilité réduit la dépendance à l'égard de fournisseurs d'IA spécifiques, simplifie l'intégration aux systèmes existants et vous permet d'améliorer les capacités des agents au fil du temps.

Des protocoles bien conçus établissent également des modèles de sécurité cohérents pour l'authentification et l'autorisation au sein de votre écosystème d'agents. Plus important encore, la portabilité des protocoles préserve votre liberté d'adopter de nouveaux frameworks et fonctionnalités d'agents au fur et à mesure de leur apparition. Le choix de protocoles ouverts protège votre investissement dans le développement d'agents tout en maintenant l'interopérabilité avec les systèmes tiers.

Avantages des protocoles ouverts

Lorsque vous implémentez vos propres extensions ou que vous créez des systèmes d'agents personnalisés, les protocoles ouverts offrent des avantages indéniables :

- Documentation et transparence — Fournissez généralement une documentation complète et des implémentations transparentes
- Support communautaire : accès à des communautés de développeurs plus larges pour le dépannage et les meilleures pratiques
- Garanties d'interopérabilité : meilleure assurance que vos extensions fonctionneront sur différentes implémentations
- Compatibilité future : réduction du risque d'interruption des modifications ou de dépréciation
- Influence sur le développement — Possibilité de contribuer à l'évolution du protocole

Agent-to-agent protocoles

Le tableau suivant fournit une vue d'ensemble des protocoles agentiques qui permettent à plusieurs agents de collaborer, de déléguer des tâches et de partager des informations.

Protocole	Idéal pour	Considérations

Communication entre agents MCP	Organisations à la recherche de modèles de collaboration flexibles entre agents	• Une extension du protocole MCP (Model Context Protocol) proposée par AWS qui s'appuie sur ses bases de agent-to-agent communication existantes • Permet une collaboration fluide entre les agents grâce OAuth à une sécurité basée
Protocole A2A	Écosystèmes d'agents multiplateformes	• Soutenu par Google • Norme plus récente avec une adoption plus limitée par rapport au MCP

Choix des options de protocole

Lors de agent-to-agent la mise en œuvre de la communication, adaptez vos exigences de communication spécifiques aux capacités de protocole appropriées. Les différents modèles d'interaction nécessitent des fonctionnalités de protocole différentes. Le tableau suivant décrit les modèles de communication courants et recommande les choix de protocole les plus adaptés à chaque scénario.

Modèle	Description	Choix de protocole idéal
Demande et réponse simples	Interactions ponctuelles entre agents	MCP avec flux apatrides
Dialogues dynamiques	Conversations continues avec le contexte	MCP avec gestion de session
Collaboration multi-agents	Interactions complexes entre plusieurs agents	Inter-agent MCP ou AutoGen
Flux de travail basés sur l'équipe	Des équipes d'agents hiérarchiques avec des rôles définis	Inter-agent MCP, ou CrewAI AutoGen

Au-delà des modèles de communication, plusieurs facteurs techniques et organisationnels peuvent influencer le choix de votre protocole. Le tableau suivant décrit les principales considérations qui peuvent vous aider à déterminer quel protocole correspond le mieux à vos exigences de mise en œuvre spécifiques.

Considération	Description	Exemple
Modèle de sécurité	Exigences en matière d'authentification et d'autorisation	OAuth 2,0 en MCP
Environnement de déploiement	Où les agents courront et communiqueront	Machine distribuée ou unique
Compatibilité avec les écosystèmes	Intégration avec les frameworks d'agents existants	LangChain ou Strands Agents
Besoins d'évolutivité	Croissance attendue des interactions entre agents	Capacités de streaming de MCP

Sélection de protocoles agentiques

Pour la plupart des organisations qui créent des systèmes d'agents de production, le protocole MCP (Model Context Protocol) constitue la base de communication la plus complète et la mieux prise en charge. agent-to-agent MCP bénéficie des contributions de développement actives de la part de la communauté open source AWS et de celle-ci.

Il est important de sélectionner les bons protocoles agentic pour les organisations qui cherchent à mettre en œuvre efficacement l'IA agentic. Les considérations varient en fonction du contexte organisationnel.

Considérations relatives à la sélection du protocole agentic

Organisations devraient prendre en compte les meilleures pratiques suivantes lors de la sélection de protocoles pour les systèmes d'IA agentic :

- Prioriser les standards ouverts — Les organisations devraient adopter des protocoles ouverts tels que le MCP pour garantir l'interopérabilité et l'extensibilité à long terme et pour réduire le risque de dépendance vis-à-vis d'un fournisseur.
- Équilibre entre rapidité et flexibilité — Les entreprises en démarrage et les premiers utilisateurs peuvent commencer par utiliser des protocoles propriétaires bien pris en charge pour un développement rapide, mais doivent définir une voie de migration vers des standards ouverts à mesure que les systèmes arrivent à maturité.
- Implémentation de couches d'abstraction : les entreprises doivent mettre en œuvre l'abstraction des protocoles pour simplifier la migration, permettre l'adoption hybride et mettre en œuvre des stratégies d'intégration pérennes.
- Mettre l'accent sur la sécurité et la conformité — Les organisations des secteurs réglementés doivent sélectionner des protocoles dotés de fonctionnalités d'authentification, de chiffrement et d'audit robustes pour répondre aux exigences de gouvernance et de conformité.
- Évaluer la maturité de l'écosystème — Toutes les organisations devraient évaluer l'état de santé, l'adoption et le soutien communautaire de chaque protocole afin de garantir la durabilité et de minimiser la dette technique.
- S'engager dans l'élaboration de normes — Les organisations devraient participer à des organismes de normalisation ou à des communautés open source pour contribuer à façonner l'évolution des protocoles et influencer les meilleures pratiques.
- Tenez compte de la souveraineté des données — Le gouvernement et les secteurs réglementés doivent veiller à ce que les choix de protocoles soient conformes aux exigences de résidence et de souveraineté des données dans les régions de déploiement.
- Tirez parti des services gérés : dans la mesure du possible, utilisez des implémentations gérées ou sans serveur de protocoles agentic pour réduire la complexité opérationnelle et accélérer le déploiement.

Stratégie de mise en œuvre des protocoles agentiques

Pour mettre en œuvre efficacement les protocoles agentic au sein de votre organisation, considérez les étapes stratégiques suivantes :

1. Commencez par l'alignement des normes — Adoptez des protocoles ouverts établis dans la mesure du possible.

2. Créez des couches d'abstraction : implémentez des adaptateurs entre vos systèmes et des protocoles spécifiques.
3. Contribuez aux standards ouverts — Participez aux communautés de développement de protocoles.
4. Surveillez l'évolution des protocoles : restez informé des nouvelles normes et des mises à jour.
5. Testez régulièrement l'interopérabilité : vérifiez que vos implémentations restent compatibles.

Commencer à utiliser MCP

AWS soutient activement le Model Context Protocol (MCP) en contribuant au développement et à la mise en œuvre du protocole. AWS collabore avec les principaux frameworks d'agents open source, notammentLangGraph, et CrewAILlamaIndex, pour façonner le futur de la communication inter-agents sur le protocole.

Pour implémenter le MCP dans l'architecture de votre agent, effectuez les actions suivantes :

1. Explorez les implémentations de MCP dans des frameworks tels que le Strands Agents SDK.
2. Consultez la documentation technique du Model Context Protocol.
3. Lisez Protocoles ouverts pour l'interopérabilité des agents, partie 1 : communication entre agents sur MCP (AWS blog) pour en savoir plus sur l'interopérabilité des agents.
4. Rejoignez la communauté MCP pour influencer l'évolution du protocole.

Le MCP fournit une couche de communication qui permet aux agents d'interagir avec des données et des services externes et peut également être utilisée pour permettre aux agents d'interagir avec d'autres agents. L'implémentation du transport HTTP Streamable du protocole fournit aux développeurs un ensemble complet de modèles d'interaction sans avoir à réinventer la roue. Ces modèles prennent en charge à la fois les request/response flux apatrides et la gestion dynamique des sessions avec persistance. IDs

En adoptant des protocoles ouverts tels que le MCP, vous positionnez votre organisation pour créer des systèmes d'agents qui restent flexibles, interopérables et adaptables à mesure que la technologie de l'IA évolue. Pour plus d'informations sur la mise en œuvre agent-to-tool du protocole, voir Stratégie d'intégration des outils plus loin dans ce guide.

Commencer à utiliser A2A

Le protocole Agent2Agent (A2A) permet une collaboration décentralisée entre les agents via une couche sémantique partagée. Au lieu d'acheminer tout le travail via un orchestrateur central, A2A permet aux agents de se découvrir, de promouvoir leurs capacités, de négocier des tâches et de partager le contexte à l'aide d'un protocole léger basé sur JSON. Chaque agent publie un manifeste de capacités.

L'exemple suivant montre un manifeste de fonctionnalités A2A simplifié qui annonce les actions prises en charge par un agent, les entrées requises et les métadonnées opérationnelles pour permettre la découverte et la négociation des tâches :

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

Ce modèle permet l'appariement dynamique des capacités, la délégation en milieu de tâche et la collaboration interorganisationnelle. Les agents peuvent s'auto-organiser autour des tâches, former des groupes de travail temporaires et s'adapter à l'entrée ou à la sortie de nouvelles fonctionnalités dans le système.

A2A prend en charge des interactions allant de simples demandes apatrides à des sessions de négociation en plusieurs étapes, notamment :

- peer-to-peerMessagerie directe pour une collaboration à faible latence
- Négociation sémantique des tâches, où les agents sélectionnent le pair le plus approprié
- Découverte basée sur les capacités, permettant une division émergente du travail
- Ancrage de session pour des interactions dynamiques en plusieurs étapes

En adoptant des protocoles ouverts natifs pour les agents tels que l'A2A, les entreprises créent des systèmes d'IA modulaires, interopérables et capables de collaboration transfrontalière. L'A2A garantit que les écosystèmes d'agents restent flexibles et peuvent évoluer à mesure que de nouveaux agents, équipes ou systèmes externes sont introduits, sans nécessiter de couches d'orchestration rigides ni de couplage préalable.

Pour implémenter le protocole A2A dans l'architecture de votre agent, effectuez les actions suivantes :

1. Consultez la spécification du protocole A2A — Lisez la dernière version de la spécification du protocole Agent2Agent (A2A) pour savoir comment fonctionnent les manifestes de capacités, les flux de négociation et la poignée de main des agents.
2. Explorez les environnements d'exécution compatibles avec A2A : évaluez les frameworks tels que le SDK Strands Agents ou les couches d'exécution personnalisées qui prennent en charge les manifestes de capacités et la négociation de style A2A. peer-to-peer
3. Implémentez un manifeste de capacités pour vos agents : définissez les `meta` champs et les champs de `can` chaque agent pour permettre la découverte, le matchmaking et la collaboration au niveau de l'intention. `needs`
4. Testez les modèles de négociation A2A : utilisez la boucle demande-offre-acceptation, les requêtes de capacité structurées ou la découverte basée sur des ragots pour comprendre comment les agents réfléchissent à qui doit gérer une tâche.
5. Testez l'A2A dans un environnement d'infrastructure mixte : associez la négociation A2A entre pairs à un routage d'événements natif via AWS Amazon EventBridge pour évaluer les modèles de coordination hybrides.
6. Rejoignez la communauté A2A : participez au groupe de travail ouvert pour vous tenir au courant des extensions, des recommandations de sécurité et des améliorations de l'interopérabilité entre fournisseurs, et contribuer au développement du protocole.

Outils

Les agents d'intelligence artificielle apportent de la valeur en interagissant avec des APIs outils externes et des sources de données pour effectuer des tâches utiles. La bonne stratégie d'intégration des outils a un impact direct sur les capacités de votre agent, son niveau de sécurité et sa flexibilité à long terme.

Cette section vous aide à naviguer dans le paysage de l'intégration des outils en mettant l'accent sur les normes ouvertes qui maximisent votre liberté et votre flexibilité. La section met en évidence le protocole MCP (Model Context Protocol) pour l'intégration des outils et passe en revue les outils spécifiques au framework et les méta-outils spécialisés qui améliorent les flux de travail des agents.

Dans cette section :

Catégories d'outils

Les systèmes d'agents de construction impliquent trois catégories principales d'outils.

Outils basés sur des protocoles

Les outils basés sur des protocoles utilisent des protocoles normalisés pour la agent-to-tool communication :

- Outils MCP : outils standard ouverts qui fonctionnent dans tous les frameworks avec des options d'exécution locales et distantes.
- OpenAIappel de fonctions — Outils propriétaires spécifiques aux OpenAI modèles.
- Anthropicoutils — Variation d'une OpenAI fonction faisant appel à des outils propriétaires spécifiques aux modèles de Anthropic Claude.

Outils natifs du framework

Les outils natifs du framework sont intégrés directement dans des frameworks d'agents spécifiques :

- Strands Agents outils — quick-to-implement Outils légers, spécifiques au Strands Agents framework.
- LangChainoutils : des outils Python basés sur des outils étroitement intégrés à l'LangChainécosystème.
- LlamaIndexoutils — Outils optimisés pour la récupération et le traitement des données au sein LlamaIndex de.

Méta-outils

Les méta-outils améliorent les flux de travail des agents sans effectuer directement d'actions externes :

- Outils de flux de travail : gérez le flux d'exécution des agents, la logique de branchement et la gestion des états.
- Outils de création de graphes d'agents : coordonnez plusieurs agents dans des flux de travail complexes.
- Outils de mémoire : fournissent un stockage permanent et une récupération d'informations entre les sessions des agents.
- Outils de réflexion : permettez aux agents d'analyser et d'améliorer leurs propres performances.

Outils basés sur des protocoles

En ce qui concerne les outils basés sur des protocoles, le Model Context Protocol (MCP) fournit la base la plus complète et la plus flexible pour l'intégration des outils. Comme indiqué dans le billet de blog AWS Open Source sur l'interopérabilité des agents, AWS a adopté le protocole MCP en tant que protocole stratégique, contribuant ainsi activement à son développement.

Le tableau suivant décrit les options de déploiement de l'outil MCP.

Modèle de déploiement	Description	Idéal pour	Mise en œuvre

Basé sur un studio local	Les outils s'exécutent selon le même processus que l'agent	Développement, tests et outils simples	Rapide à mettre en œuvre sans surcharge réseau
Basé sur des événements envoyés par le serveur local (SSE)	Les outils s'exécutent localement mais communiquent via HTTP	Outils locaux plus complexes avec séparation des préoccupations	Meilleure isolation mais faible latence
Diffusable via HTTP à distance	Outils exécutés sur des serveurs distants	Environnements de production et outils partagés	Évolutif et géré de manière centralisée

Les MCP officiels SDKs sont disponibles pour créer des outils MCP :

- PythonSDK — Implémentation complète avec prise en charge complète des protocoles
- TypeScriptSDK —JavaScript/TypeScriptimplémentation pour les applications Web
- JavaSDK — Implémentation de Java pour les applications d'entreprise

Ils SDKs fournissent les éléments de base pour créer des outils compatibles MCP dans votre langage préféré, avec des implémentations cohérentes de la spécification du protocole.

En outre, AWS a implémenté le MCP dans le Strands AgentsSDK. Le Strands Agents SDK fournit un moyen simple de créer et d'utiliser des outils compatibles avec MCP. Une documentation complète est disponible dans le Strands Agents GitHub référentiel. Pour des cas d'utilisation plus simples ou lorsque vous travaillez en dehors du Strands Agents cadre, les MCP officiels SDKs proposent des implémentations directes du protocole dans plusieurs langues.

Caractéristiques de sécurité des outils MCP

Les fonctionnalités de sécurité des outils MCP sont les suivantes :

- OAuth Authentification 2.0/2.1 — Authentification conforme aux normes du secteur
- Étendue des autorisations : contrôle d'accès précis pour les outils
- Découverte des capacités des outils — Découverte dynamique des outils disponibles
- Gestion structurée des erreurs — Modèles d'erreur cohérents

Commencer à utiliser les outils MCP

Pour implémenter le MCP pour l'intégration des outils, effectuez les actions suivantes :

1. Explorez le Strands AgentsSDK pour une implémentation MCP prête pour la production.
2. Consultez la documentation technique du MCP pour comprendre les concepts de base.
3. Utilisez les exemples pratiques décrits dans ce billet de blog AWS Open Source.
4. Commencez par de simples outils locaux avant de passer aux outils distants.
5. Rejoignez la communauté MCP pour influencer l'évolution du protocole.

Découvrez AgentCore Gateway

Amazon Bedrock AgentCore Gateway fournit aux développeurs un moyen simple et sécurisé de créer, déployer, découvrir et se connecter à des outils MCP et à d'autres points de terminaison cibles à grande échelle. Avec AgentCore Gateway, les développeurs peuvent convertir APIs les AWS Lambda fonctions et les services existants en outils compatibles avec MCP. Ensuite, avec seulement quelques lignes de code, ils peuvent mettre ces outils à la disposition des agents via les points de terminaison AgentCore Gateway. AgentCore Gateway prend en charge OpenAPISmithy, et Lambda en tant que types d'entrée, et constitue la seule solution qui fournit à la fois une authentification d'entrée et une authentification de sortie complètes dans un service entièrement géré.

Outils natifs du framework

Bien que le protocole MCP (Model Context Protocol) constitue la base la plus flexible, les outils natifs du framework offrent des avantages pour des cas d'utilisation spécifiques.

Le Strands AgentsSDK propose des outils Python basés sur des outils qui se caractérisent par leur conception légère qui nécessite une surcharge minimale pour des opérations simples. Ils permettent une mise en œuvre rapide et permettent aux développeurs de créer des outils avec seulement quelques lignes de code. De plus, ils sont étroitement intégrés pour fonctionner parfaitement dans le Strands Agents cadre.

L'exemple suivant montre comment créer un outil météo simple à l'aide deStrands Agents. Les développeurs peuvent rapidement transformer les Python fonctions en outils accessibles aux agents avec une surcharge de code minimale et générer automatiquement la documentation appropriée à partir de la docstring de la fonction.

```
#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."
```

Pour le prototypage rapide ou les cas d'utilisation simples, les outils natifs du framework peuvent accélérer le développement. Toutefois, pour les systèmes de production, les outils MCP offrent une meilleure interopérabilité et une flexibilité future par rapport aux outils natifs du framework.

Le tableau suivant fournit une vue d'ensemble des autres outils spécifiques au framework.

Cadre	Type d'outil	Avantages	Considérations
AutoGen	Définitions des fonctions	Support multi-agents puissant	Microsoftécosystème
LangChain	Pythoncours	Vaste écosystème d'outils prédéfinis	Verrouillage du cadre
LlamaIndex	Fonctions Python	Optimisé pour les opérations de données	Limité à LlamaIndex

Méta-outils

Les méta-outils n'interagissent pas directement avec les systèmes externes. Ils améliorent plutôt les capacités des agents en mettant en œuvre des modèles agentiques. Cette section traite du flux de travail, du graphe de l'agent et des méta-outils de mémoire.

Méta-outils de flux de travail

Les méta-outils du flux de travail gèrent le flux d'exécution des agents :

- Gestion des états — Maintien du contexte lors des interactions entre plusieurs agents
- Logique de branchement — Activer les chemins d'exécution conditionnels
- Mécanismes de nouvelle tentative — Gérez les échecs grâce à des stratégies de relance sophistiquées

Les exemples de frameworks dotés de méta-outils de flux de travail incluent des fonctionnalités de Strands Agentsflux LangGraphde travail.

Méta-outils Agent Graph

Les méta-outils Agent Graph coordonnent la collaboration de plusieurs agents :

- Délégation de tâches — Attribuer des sous-tâches à des agents spécialisés
- Agrégation des résultats — Combinez les sorties de plusieurs agents
- Résolution des conflits — Résoudre les désaccords entre les agents

Les frameworks aiment AutoGenet CrewAIse spécialisent dans la coordination des graphes d'agents.

Méta-outils de mémoire

Les méta-outils de mémoire fournissent un stockage et une récupération persistants :

- Historique des conversations : maintenez le contexte entre les sessions
- Bases de connaissances — Stockez et récupérez des informations spécifiques au domaine
- Magasins vectoriels — Activez les fonctionnalités de recherche sémantique

Le système de ressources de MCP fournit un moyen standardisé d'implémenter des méta-outils de mémoire qui fonctionnent avec différents frameworks d'agents.

Stratégie d'intégration des outils

Le choix de la stratégie d'intégration des outils a un impact direct sur ce que vos agents peuvent accomplir et sur la facilité avec laquelle votre système peut évoluer. Priorisez les protocoles ouverts tels que le Model Context Protocol (MCP) tout en utilisant stratégiquement des outils et des méta-outils natifs du framework. Ainsi, vous pouvez créer un écosystème d'outils qui reste flexible et puissant à mesure que la technologie de l'IA progresse.

L'approche stratégique suivante en matière d'intégration des outils maximise la flexibilité tout en répondant aux besoins immédiats de votre organisation :

1. Adoptez MCP comme base : MCP fournit un moyen standardisé de connecter les agents à des outils dotés de fonctionnalités de sécurité puissantes. Commencez par utiliser MCP comme protocole d'outil principal pour :
   - Des outils stratégiques qui seront utilisés dans le cadre de la mise en œuvre de plusieurs agents.
   - Outils sensibles à la sécurité qui nécessitent une authentification et une autorisation robustes.
   - Outils nécessitant une exécution à distance dans les environnements de production.
2. Utilisez des outils natifs du framework le cas échéant — Envisagez des outils natifs du framework pour :
   - Prototypage rapide lors du développement initial.
   - Des outils simples et non critiques avec des exigences de sécurité minimales.
   - Fonctionnalité spécifique au framework qui tire parti de capacités uniques.
3. Implémentez des méta-outils pour des flux de travail complexes — Ajoutez des méta-outils pour améliorer l'architecture de vos agents :
   - Commencez simplement avec les modèles de flux de travail de base.
   - Ajoutez de la complexité à mesure que vos cas d'utilisation évoluent.
   - Standardisez les interfaces entre les agents et les méta-outils.
4. Planifiez l'évolution — Construisez en gardant à l'esprit la flexibilité future :
   - Documentez les interfaces des outils indépendamment des implémentations.
   - Créez des couches d'abstraction entre les agents et les outils.
   - Établissez des voies de migration entre des protocoles propriétaires et des protocoles ouverts.

Bonnes pratiques de sécurité pour l'intégration des outils

L'intégration des outils a un impact direct sur votre niveau de sécurité. Cette section décrit les meilleures pratiques à prendre en compte pour votre organisation.

Authentification et autorisation

Utilisez les contrôles d'accès robustes suivants :

- Utiliser OAuth 2.0/2.1 — Implémenter une authentification conforme aux normes du secteur pour les outils distants.
- Implémenter le moindre privilège : accordez aux outils uniquement les autorisations dont ils ont besoin.
- Rotation des informations d'identification : mettez régulièrement à jour les clés d'API et les jetons d'accès.

Protection des données

Pour aider à protéger les données, adoptez les mesures suivantes :

- Valider les entrées et les sorties : implémentez la validation du schéma pour toutes les interactions avec les outils.
- Chiffrez les données sensibles : utilisez le protocole TLS pour toutes les communications avec les outils distants.
- Implémenter la minimisation des données : ne transmettez que les informations nécessaires aux outils.

Surveillance et audit

Maintenez la visibilité et le contrôle en utilisant les mécanismes suivants :

- Enregistrez toutes les invocations d'outils : maintenez des pistes d'audit complètes.
- Surveillez les anomalies : détectez les modèles d'utilisation inhabituels des outils.
- Implémentez la limitation du débit : empêchez les abus liés à des appels d'outils excessifs.

Le modèle de sécurité MCP (Model Context Protocol) répond à ces préoccupations de manière globale. Pour plus d'informations, consultez la section Considérations relatives à la sécurité dans la documentation MCP.

Conclusion

Le paysage de l'IA agentique continue d'évoluer rapidement, offrant aux entreprises de nouveaux moyens puissants de créer des systèmes intelligents et autonomes. Ce guide a exploré trois éléments essentiels pour une mise en œuvre réussie : les cadres qui fournissent les bases, les plateformes qui fournissent l'environnement, les protocoles qui permettent la communication et les outils qui étendent les capacités.

À mesure que les frameworks arrivent à maturité, vous pouvez vous attendre à une interopérabilité accrue, à une standardisation autour de protocoles tels que le Model Context Protocol (MCP) et à des capacités d'orchestration plus sophistiquées pour les agents autonomes. Organisations ayant acquis une expertise avec ces frameworks aujourd'hui seront bien placées pour créer des agents de plus en plus autonomes et intelligents offrant une valeur commerciale significative.

Les plateformes fournissent l'environnement d'exécution, de gouvernance et de cycle de vie dans lequel les systèmes agentic fonctionnent. Ils répondent à des préoccupations telles que l'identité, les limites de sécurité, l'observabilité, la gestion de la mémoire, l'ancrage des sessions et l'interaction sécurisée avec les outils et les données. Dans AWS les environnements, les plateformes telles que les environnements d'exécution des agents gérés et les services d'orchestration permettent aux entreprises de déployer, de surveiller, de faire évoluer et de gouverner des agents autonomes et des systèmes agentiques à grande échelle. Les plateformes relient les cadres fondamentaux aux exigences opérationnelles réelles.

Le choix des protocoles d'agent représente une décision stratégique qui équilibre les besoins de développement immédiats avec la flexibilité et l'interopérabilité à long terme. En donnant la priorité aux protocoles ouverts et en créant des couches d'abstraction appropriées, les entreprises peuvent créer des systèmes d'agents qui restent adaptables à l'évolution des technologies tout en répondant aux exigences commerciales actuelles.

Pour la plupart des organisations, MCP représente une base solide en raison de son standard ouvert, de son écosystème en pleine croissance, de sa prise en charge des modèles de agent-to-agent communication et de ses capacités d'intégration d'outils. AWS a adopté MCP et Agent2Agent (A2A) en tant que protocoles stratégiques, contribuant activement à leur développement et à leur mise en œuvre dans des services tels que le SDK. Strands Agents En utilisant MCP ou A2A avec des outils et des méta-outils natifs du framework appropriés, vous pouvez créer des systèmes d'agents qui offrent une valeur immédiate tout en restant adaptables aux innovations futures.

Ressources

Utilisez les ressources suivantes AWS et d'autres ressources liées au développement d'agents autonomes.

AWS Blogues

- Amazon Bedrock AgentCore Memory : création d'agents sensibles au contexte
- Meilleures pratiques pour créer des applications d'IA générative robustes avec Amazon Bedrock Agents — Partie 1
- Meilleures pratiques pour créer des applications d'IA générative robustes avec Amazon Bedrock Agents — Partie 2
- Construisez de puissants pipelines RAG avec Amazon LlamaIndex Bedrock
- Créez des agents d'IA fiables avec Amazon Bedrock Observability AgentCore
- Évaluez les réponses RAG avec Amazon, Bedrock et RAGAS LlamaIndex
- Présentation de l'interpréteur de AgentCore code Amazon Bedrock
- Présentation d'Amazon Bedrock AgentCore Gateway : transformer le développement d'outils d'agent d'intelligence artificielle pour les entreprises
- Présentation d'Amazon Bedrock AgentCore Identity : sécuriser l'IA agentique à grande échelle
- Présentation Strands Agents d'un SDK Open Source pour les agents AI
- Protocoles ouverts pour l'interopérabilité des agents, partie 1 : communication entre agents sur MCP
- Lancez et faites évoluer vos agents et outils en toute sécurité sur Amazon Bedrock Runtime AgentCore
- AWS Transform pour .NET, le premier service d'intelligence artificielle agentique permettant de moderniser les applications .NET à grande échelle
- AWS Tour d'horizon hebdomadaire : Strands Agents

AWS Directives prescriptives

- Opérationnaliser l'IA agentique sur AWS
- Les fondements de l'IA agentic sur AWS

- Modèles et flux de travail d'IA agentic sur AWS
- Création d'architectures sans serveur pour l'IA agentic sur AWS
- Création d'architectures multi-locataires pour l'IA agentic sur AWS
- Sécurité pour l'IA agentic sur AWS
- Récupérez les options et architectures de génération augmentée sur AWS

AWS ressources

- Documentation Amazon Bedrock
- Documentation Amazon Bedrock AgentCore
- Boîte à outils de AgentCore démarrage Amazon Bedrock (GitHub référentiel)
- Documentation Amazon Nova
- AWS Serveurs MCP (GitHubréférentiel)

Autres ressources

- AutoGendocumentation (Microsoft)
- Création d'agents efficaces (Anthropic)
- CrewAI GitHubréférentiel
- Documentation LangChain
- LangGraphplateforme
- Documentation LlamaIndex
- Documentation du protocole Model Context
- Documentation Strands Agents
- Strands AgentsVue d'ensemble des outils
- Strands AgentsGuide de démarrage rapide

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Nouvelle section	Section « Plateformes » ajoutée	16 janvier 2026
Publication initiale	—	14 juillet 2025

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.