



Modèles et flux de travail d'IA agentique sur AWS

AWS Conseils prescriptifs



AWS Conseils prescriptifs: Modèles et flux de travail d'IA agentic sur AWS

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

Table of Contents

Introduction	1
Public visé	1
Objectifs	1
À propos de cette série de contenus	2
Modèles d'agents	3
Agents de raisonnement de base	4
Architecture	4
Description	5
Fonctionnalités	6
Limitations	6
Cas d'utilisation courants	6
Directives d'implémentation	7
Résumé	7
Architecture	7
Description	8
Fonctionnalités	9
Cas d'utilisation courants	9
Directives d'implémentation	9
Résumé	10
Tool-based agents pour les fonctions d'appel	10
Architecture	10
Description	11
Fonctionnalités	12
Cas d'utilisation courants	12
Directives d'implémentation	12
Résumé	13
Tool-based agents pour serveurs	13
Architecture	13
Description	14
Fonctionnalités	15
Cas d'utilisation courants	15
Directives d'implémentation	15
Résumé	16
Computer-use agents	16

Architecture	16
Description	17
Fonctionnalités	18
Cas d'utilisation courants	18
Directives d'implémentation	19
Résumé	19
Agents de codage	19
Architecture	19
Description	20
Fonctionnalités	21
Cas d'utilisation courants	21
Directives d'implémentation	22
Résumé	22
Agents vocaux et vocaux	22
Architecture	22
Description	23
Fonctionnalités	24
Cas d'utilisation courants	24
Directives d'implémentation	25
Résumé	25
Agents d'orchestration des flux de travail	25
Architecture	26
Description	26
Fonctionnalités	27
Cas d'utilisation courants	27
Directives d'implémentation	27
Résumé	28
Memory-augmented agents	28
Architecture	28
Description	29
Fonctionnalités	30
Cas d'utilisation courants	30
Implémentation d'agents à mémoire augmentée	30
Implémentation de l'envoi de messages par injection de mémoire	31
Résumé	32
Agents de simulation et de banc d'essai	32

Architecture	32
Description	33
Fonctionnalités	34
Cas d'utilisation courants	34
Directives d'implémentation	35
Résumé	36
Observateurs et agents de surveillance	36
Architecture	37
Description	37
Fonctionnalités	38
Cas d'utilisation courants	38
Directives d'implémentation	39
Résumé	39
Multi-agent collaboration	40
Description	41
Fonctionnalités	42
Cas d'utilisation courants	42
Directives d'implémentation	42
Résumé	43
Conclusion	43
Plats à emporter	44
Flux de travail LLM	45
Vue d'ensemble de la cognition augmentée par le LLM	45
Flux de travail pour un chaînage rapide	46
Description	47
Fonctionnalités	48
Cas d'utilisation courants	48
Flux de travail pour le routage	48
Fonctionnalités	50
Cas d'utilisation courants	50
Workflow pour la parallélisation	50
Fonctionnalités	51
Cas d'utilisation courants	51
Flux de travail pour l'orchestration	52
Fonctionnalités	53
Cas d'utilisation courants	53

Flux de travail pour les évaluateurs et boucles d'affinage par réflexion	53
Cas d'utilisation courants	55
Fonctionnalités	55
Conclusion	55
Modèles de flux de travail agentic	57
Des systèmes pilotés par les événements aux systèmes augmentés par la cognition	57
Architecture basée sur les événements	58
Flux de travail augmentés par la cognition	59
Informations de base	60
Motifs de saga à enchaînement rapide	60
Chorégraphie de saga	61
Schéma de chaînage rapide	62
Chorégraphie d'agent	62
Plats à emporter	64
Acheminement des modèles d'expédition dynamiques	64
Expédition dynamique	65
Routage basé sur le LLM	66
Routeur agent	67
Plats à emporter	68
Modèles de parallélisation et de diffusion	68
Dispersez et collectez	69
Parallélisation basée sur le LLM (cognition par diffusion)	71
Parallélisation des agents	71
Plats à emporter	72
Modèles d'orchestration de Saga	72
Orchestration d'événements	73
Système d'agent basé sur les rôles (orchestrateur)	74
Superviseur	74
Plats à emporter	76
L'évaluateur reflète et affine les modèles de boucles	76
Boucle de contrôle du feedback	77
Boucle de régulation du feedback (évaluateur)	79
Évaluateur	79
Plats à emporter	80
Conception de flux de travail agentiques sur AWS	81
Conclusion	81

Historique du document	83
Glossaire	84
#	84
A	85
B	88
C	90
D	94
E	98
F	100
G	102
H	104
I	105
L	108
M	109
O	114
P	116
Q	119
R	120
S	123
T	127
U	128
V	129
W	129
Z	131
.....	cxxxii

Modèles et flux de travail d'IA agentic sur AWS

Aaron Sempf et Andrew Hooker, Amazon Web Services

Juillet 2025 ([historique du document](#))

Organisations adoptent de grands modèles linguistiques (LLMs) et des agents logiciels pour résoudre des problèmes multidomains dynamiques en utilisant une nouvelle discipline architecturale appelée modèles agentiques. Les modèles agentiques sont des modèles fondamentaux et des structures modulaires utilisés pour concevoir et orchestrer des agents d'IA orientés vers des objectifs dans de nombreux contextes.

Public visé

Ce guide est destiné aux architectes, aux développeurs et aux chefs de produit qui souhaitent créer des applications intelligentes allant au-delà de la logique statique, de la logique symbolique et de l'automatisation déterministe.

Objectifs

Ce guide fournit un cadre de conception et une approche de mise en œuvre pour les systèmes d'agents d'IA qui fonctionnent de manière autonome tout en restant contrôlables et alignés sur vos objectifs. Il relie les modèles architecturaux pilotés par les événements à diverses alternatives agentiques, démontrant ainsi comment créer des systèmes d'agents adaptés à la production à l'aide d'architectures cloud natives. Les sujets suivants sont abordés dans ce guide :

- **Modèles d'agents** : les modèles d'agents sont des modèles de conception réutilisables qui décrivent la structure et le comportement des agents individuels. Cela inclut les agents de raisonnement, les agents de récupération augmentée, les agents de codage, les interfaces vocales, les orchestrateurs de flux de travail et les systèmes multi-agents collaboratifs. Chaque modèle illustre la façon dont les agents perçoivent, raisonnent, agissent et apprennent, selon Services AWS une cartographie.
- **Workflows LLM** — Les flux de travail se concentrent sur la façon dont les agents les utilisent LLMs pour raisonner. Ils explorent les stratégies d'incitation et les mécanismes de planification, et décrivent comment ils LLMs sont utilisés non seulement pour générer du texte, mais également pour générer des comportements structurés, interprétables et fiables au sein d'une boucle d'agents.

- **Modèles de flux de travail agentiques** — Les modèles de flux de travail décrivent comment plusieurs agents, outils et environnements interagissent pour former des systèmes autonomes. Cela inclut les modèles d'orchestration des tâches, de délégation de sous-agents, de coordination basée sur les événements, d'observabilité et de contrôle. Ces aspects favorisent des architectures d'IA évolutives, composables et auditable.

À propos de cette série de contenus

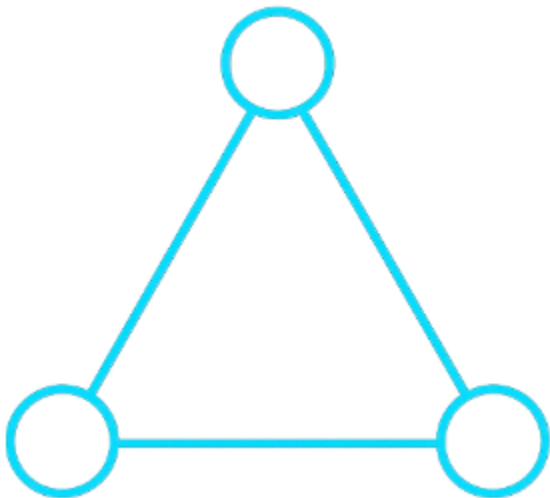
Ce guide fait partie d'une série sur l'IA agentique sur AWS. Pour plus d'informations et pour consulter les autres guides de cette série, consultez [Agentic AI](#) sur le site Web de AWS Prescriptive Guidance.

Modèles d'agents

Les modèles d'agents sont des éléments de base réutilisables et composables qui peuvent être adaptés à des domaines, à des cas d'utilisation et à des niveaux de complexité spécifiques. Les systèmes Agentique diffèrent toutefois des applications traditionnelles. Au cœur de toutes les conceptions d'agents d'IA se trouve un modèle conceptuel fondé sur les trois principes fondamentaux suivants :

- Asynchrone : les agents fonctionnent dans des environnements faiblement couplés et riches en événements
- Autonomie — Les agents agissent de manière indépendante, sans contrôle humain ou externe
- Agence — Les agents agissent dans un but précis, au nom d'un utilisateur ou d'un système, dans le but d'atteindre des objectifs spécifiques

Le triangle du schéma suivant représente les éléments de base d'un agent logiciel : perception, raison et action. Cela permet à un système agentique d'observer, de prendre des décisions et d'agir dans son environnement.



De par leur conception, les modèles agentiques fournissent un langage de conception modulaire pour la création de systèmes d'IA, ce qui signifie qu'ils sont accessibles, opérationnels, extensibles et prêts pour la production. La conception de ces systèmes nécessite une attention particulière aux trois dimensions interdépendantes suivantes, qui sont abordées plus loin dans ce guide.

Dans cette section

- [Agents de raisonnement de base](#)
- [Tool-based agents pour les fonctions d'appel](#)
- [Tool-based agents pour serveurs](#)
- [Computer-use agents](#)
- [Agents de codage](#)
- [Agents vocaux et vocaux](#)
- [Agents d'orchestration des flux de travail](#)
- [Memory-augmented agents](#)
- [Agents de simulation et de banc d'essai](#)
- [Observateurs et agents de surveillance](#)
- [Multi-agent collaboration](#)

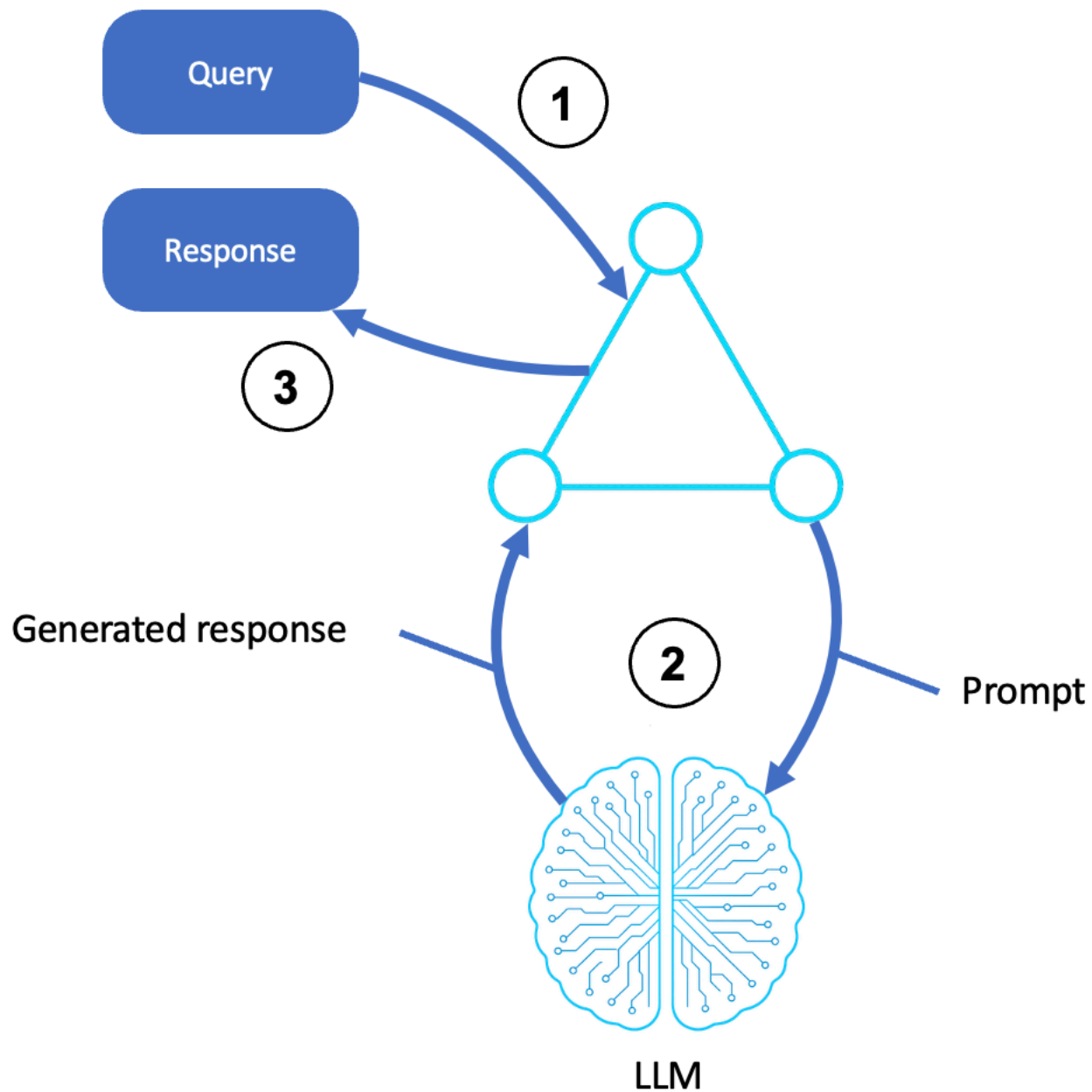
Agents de raisonnement de base

Un agent de raisonnement de base est la forme la plus simple d'IA agentique qui effectue une inférence logique ou prend des décisions en réponse à une requête. Il accepte les entrées d'un utilisateur ou d'un système, traite les requêtes et génère des réponses à l'aide d'instructions structurées.

Ce modèle est utile pour les tâches qui nécessitent un raisonnement, une classification ou une synthèse en une seule étape en fonction d'un contexte donné. Il n'utilise pas de mémoire, d'outils ou de gestion d'état, ce qui le rend apatride, léger et hautement composable dans les grands flux de travail.

Architecture

Le flux d'un agent de raisonnement de base est illustré dans le schéma suivant :



Description

1. Reçoit une entrée

- Un utilisateur, un système ou un agent en amont soumet une requête ou une instruction.
- L'entrée est transmise au shell de l'agent ou à la couche d'orchestration.
- Cette étape inclut tout prétraitement, la création de modèles d'invite et l'identification des objectifs.

2. Invoque le LLM

- L'agent transforme la requête en une invite structurée et l'envoie à un LLM (par exemple, via Amazon Bedrock).
- Le LLM génère une réponse basée sur l'invite en utilisant des connaissances et un contexte préformés.
- Le résultat généré peut inclure des étapes de raisonnement (chaîne de pensée), des réponses finales ou des options classées.

3. Renvoie une réponse

- La sortie générée est transmise à l'interface de l'agent.
- Cela peut inclure le formatage, le post-traitement ou une réponse d'API.

Fonctionnalités

- Supporte le langage naturel ou la saisie structurée
- Utilise une ingénierie rapide pour guider le comportement
- Apatrie et évolutif
- Peut être intégré dans l'interface utilisateur, la CLI, les API et les pipelines

Limitations

- Absence de mémoire ou de conscience historique
- Aucune interaction avec des outils ou des sources de données externes
- Limité à ce que le LLM sait au moment de l'inférence

Cas d'utilisation courants

- Questions et réponses conversationnelles
- Explications et résumés des politiques
- Conseils pour la prise de décisions
- Des flux de chatbot légers et automatisés
- Classification, étiquetage et notation

Directives d'implémentation

Vous pouvez utiliser les outils et services suivants pour créer un agent de raisonnement de base :

- Amazon Bedrock pour l'invocation du LLM (Anthropic, AI21, Meta)
- Amazon API Gateway ou AWS Lambda pour l'exposer en tant que microservice apatride
- Modèles d'invite stockés dans le Parameter Store ou sous forme de code AWS Secrets Manager

Résumé

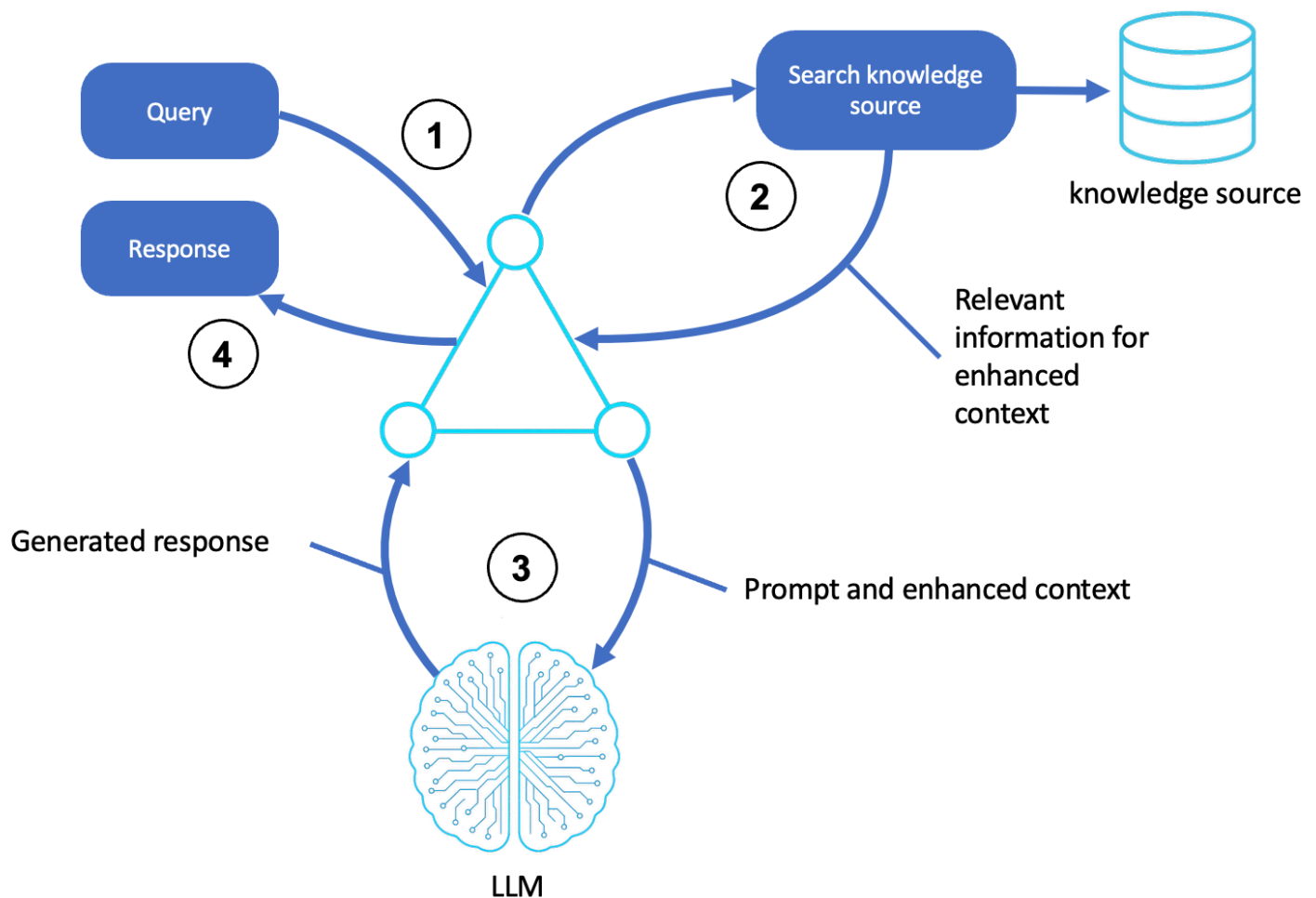
L'agent de raisonnement de base est fondamental en raison de sa structure simple. Il possède des capacités de base qui transforment les objectifs en voies de raisonnement menant à des résultats intelligents. Ce modèle est souvent le point de départ de modèles avancés, tels que les agents basés sur des outils et les agents utilisant la génération augmentée par récupération (RAG). C'est également un composant fiable et modulaire des grands flux de travail.

Agent RAG

Retrieval-augmented La génération (RAG) est une technique qui combine la récupération d'informations avec la génération de texte pour créer des réponses précises et contextuelles. RAG permet aux agents de récupérer des informations externes pertinentes avant d'engager le LLM. Il étend la mémoire effective et la précision du raisonnement d'un agent en fondant ses décisions sur des informations actualisées, factuelles ou spécifiques à un domaine. Contrairement aux LLM apatrides qui s'appuient uniquement sur des poids prédéterminés, RAG dispose d'une couche de recherche de connaissances externe qui améliore dynamiquement les instructions en fonction du contexte.

Architecture

La logique du modèle RAG est illustrée dans le schéma suivant :



Description

1. Reçoit une requête

- Un utilisateur ou un système en amont soumet une requête ou un objectif à l'agent.
- Le shell de l'agent accepte la demande et la met en forme comme une invite à raisonner.

2. Recherche dans une source externe

- L'agent identifie les concepts et l'intention à partir de la requête.
- Il interroge une source de connaissances, telle qu'un magasin de vecteurs, une base de données ou un index de documents, à l'aide d'une recherche sémantique ou d'une correspondance de mots clés.
- Les passages, documents ou entités les plus pertinents sont récupérés pour être utilisés à l'étape suivante.

3. Génère une réponse contextuelle

- L'agent complète l'invite avec les informations récupérées, formant ainsi une entrée contextuelle améliorée pour le LLM.
- Le LLM traite toutes les entrées en utilisant un raisonnement génératif (par exemple, chaîne de pensée ou réflexion) pour produire une réponse précise.

4. Renvoi le résultat final

- L'agent prépare le résultat en l'encapsulant dans n'importe quel en-tête de communication ou dans le formatage requis, puis le renvoie à l'utilisateur ou au système d'appel.
- (Facultatif) Les documents récupérés et la sortie LLM peuvent être enregistrés, notés et stockés en mémoire pour de futures requêtes.

Fonctionnalités

- Fact-grounded production même dans des domaines à long terme ou spécifiques à l'entreprise
- Extension de mémoire sans réglage précis du modèle
- Contexte dynamique basé sur chaque requête et l'état de l'utilisateur
- Entièrement compatible avec les bases de données vectorielles, les index sémantiques et le filtrage des métadonnées

Cas d'utilisation courants

- Assistants du savoir d'entreprise
- Bots de conformité réglementaire
- Copilotes du support client
- Search-enhanced chatbots
- Agents de documentation pour développeurs

Directives d'implémentation

Utilisez les outils et services suivants pour créer un agent utilisant RAG :

- Amazon Bedrock pour l'invocation d'un LLM
- Amazon Kendra ou Amazon Aurora pour la documentation ou une recherche de données structurées OpenSearch

- Amazon Simple Storage Service (Amazon S3) pour le stockage de documents
- AWS Lambda pour orchestrer la recherche, l'invite et l'inférence LLM
- Knowledge-based intégrations avec des agents (en utilisant des plugins de mémoire, des récupérateurs sémantiques ou Amazon Bedrock)

Résumé

L'agent RAG associe le raisonnement du modèle statique à l'intelligence dynamique du monde réel. Il permet aux agents de rechercher ce qu'ils ne savent pas, de synthétiser les réponses à partir des connaissances récupérées et de produire des réponses vérifiables et fiables.

Les modèles RAG constituent la base de la création d'agents intelligents capables d'étendre l'accès aux connaissances sans avoir à se recycler. Il est souvent le précurseur de modèles d'orchestration plus complexes impliquant l'utilisation d'outils, la planification et la mémoire à long terme.

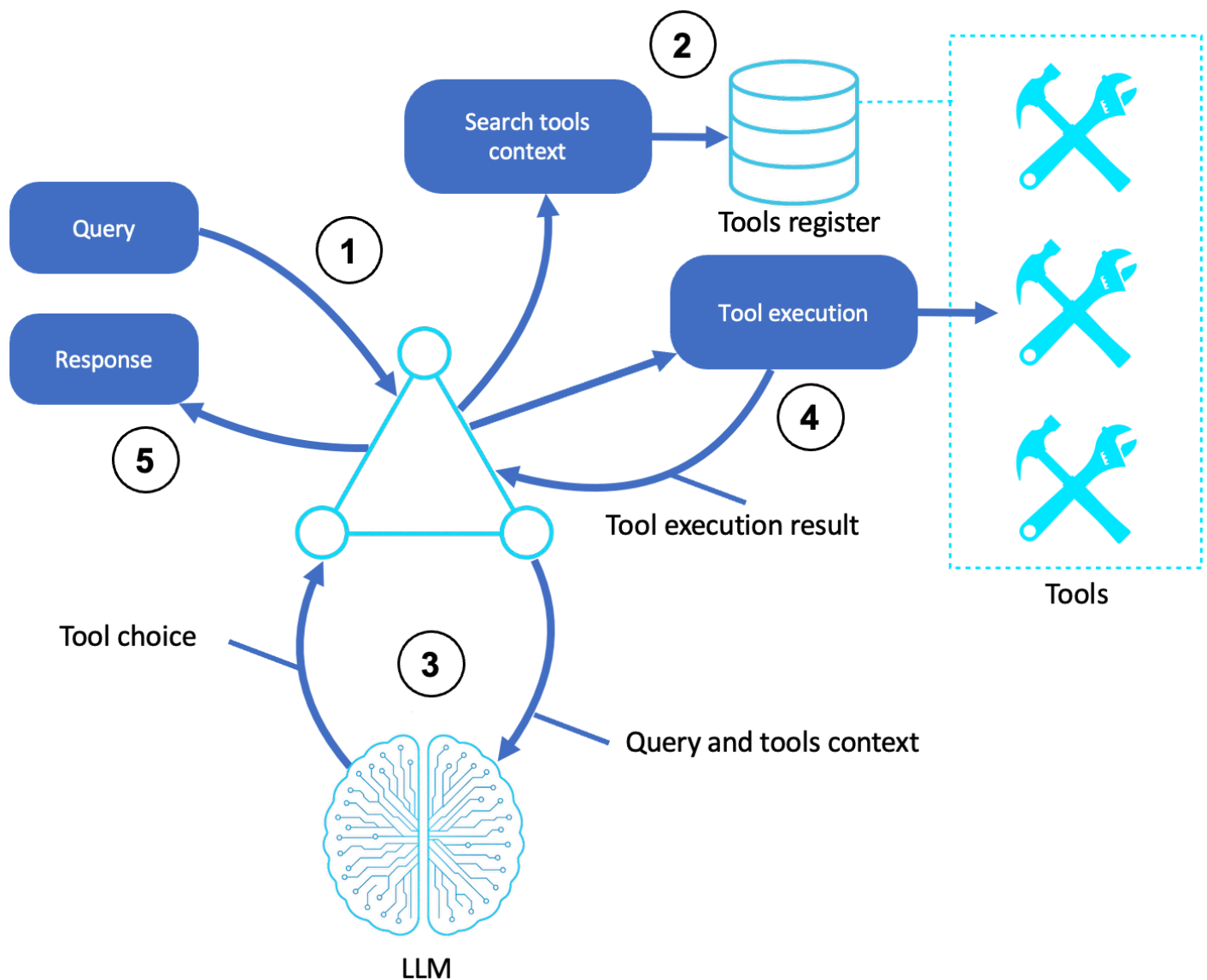
Tool-based agents pour les fonctions d'appel

Tool-based les agents étendent les capacités des agents de raisonnement en invoquant des fonctions externes ou des API pour effectuer des tâches qui vont au-delà du simple raisonnement linguistique. Ce modèle utilise un LLM pour décider de l'outil à utiliser, puis génère des arguments d'appel et intègre le résultat d'un outil dans sa boucle de raisonnement.

Ce modèle permet aux agents d'agir au lieu de simplement fournir des réponses. L'interface de l'outil représente toutes les capacités appelables, qu'il s'agisse de calculs arithmétiques, de recherches de bases de données, d'API externes ou de services cloud.

Architecture

Un agent basé sur un outil permettant d'appeler des fonctions est illustré dans le schéma suivant :



Description

1. Reçoit une requête

- L'agent reçoit une requête ou une tâche en langage naturel de la part de l'utilisateur ou du système d'appel.

2. Recherches d'outils

- L'agent utilise des métadonnées internes ou un registre d'outils pour rechercher les outils, les schémas et les fonctionnalités pertinentes disponibles.

3. Sélectionne et invoque des outils

- Le LLM reçoit les métadonnées de la requête et de l'outil (par exemple, les noms des fonctions, les types d'entrée et les descriptions) dans son invite.
- Il choisit l'outil le plus pertinent, construit des arguments d'entrée et renvoie un appel de fonction structuré.

4. Exécute l'outil choisi

- Le shell de l'agent ou le lanceur d'outils exécute la fonction sélectionnée et renvoie le résultat (par exemple, une sortie d'API, une valeur de base de données ou un calcul).

5. Renvoie une réponse

- Le LLM transmet les résultats à l'agent, soit directement, soit dans le cadre d'une invite mise à jour. Il renvoie ensuite un résultat en langage naturel.

Fonctionnalités

- Sélection d'outils dynamique en fonction du contexte de la tâche
- Schema-based invite (OpenAPI, schéma AWS JSON, interface de fonction)
- Interprétation des résultats et intégration des résultats dans le raisonnement
- Opérations apatrides ou liées à une session

Cas d'utilisation courants

- Assistants virtuels avec accès aux données externes
- Calculateurs et estimateurs financiers
- API-based travailleurs du savoir
- LLM qui invoquent AWS Lambda, SageMaker points de terminaison Amazon et services SaaS

Directives d'implémentation

Utilisez ce qui suit pour créer des agents basés sur des outils pour appeler des fonctions :

- Amazon Bedrock avec assistance pour les appels fonctionnels (Anthropic Claude)
- AWS Lambda en tant que backend d'exécution d'outils
- Amazon API Gateway ou AWS Step Functions pour l'orchestration d'outils

- Amazon DynamoDB ou Amazon Relational Database Service (Amazon RDS) pour les métadonnées des outils sensibles au contexte
- Amazon EventBridge Pipelines ou AWS Step Functions qui mappent les états pour acheminer les sorties

Résumé

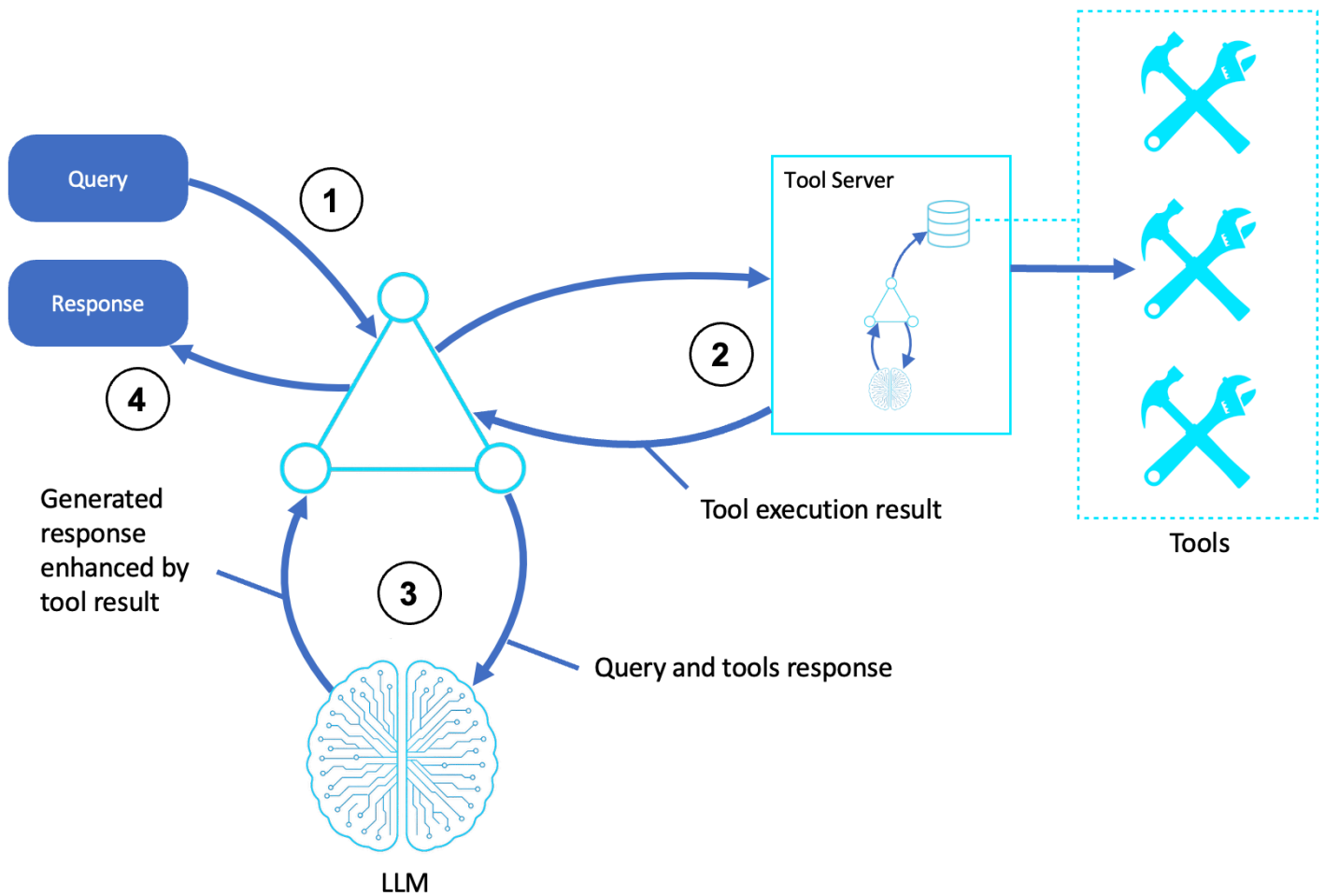
Tool-based les agents d'appel de fonctions représentent un passage de la compréhension du langage à l'exécution d'actions. Ces agents invoquent des outils dynamiques sensibles au contexte tout en maintenant le raisonnement LLM, transformant les assistants passifs en systèmes qui exécutent des tâches, accèdent aux services et intègrent les opérations commerciales. Ce modèle est un élément important de l'IA agentic dans les environnements d'entreprise, en particulier lorsqu'il est associé à des schémas déclaratifs, à des cadres d'autorisation et à des systèmes multi-agents.

Tool-based agents pour serveurs

Tool-based les agents pour serveurs améliorent les agents qui appellent des fonctions en déléguant l'exécution des outils à un serveur externe doté d'un environnement d'exécution dédié aux outils, aux scripts et aux agents composites. Contrairement aux appels de fonctions en ligne que la boucle d'agents sélectionne et invoque, les agents basés sur le serveur externalisent la logique et le pipeline d'exécution à d'autres agents ou systèmes. Cela fournit des fonctionnalités avancées telles que le chaînage multioutils, l'exécution isolée et le raisonnement spécialisé. Les serveurs d'outils sont idéaux pour les actions complexes, dynamiques ou gourmandes en ressources lorsque les outils eux-mêmes peuvent impliquer des modèles d'IA, des règles métier ou des environnements distincts.

Architecture

Voici un modèle d'agents basés sur des outils pour serveurs :



Description

1. Reçoit une requête

- Un utilisateur ou un système envoie une demande à l'agent shell.
- L'agent interprète la requête et prépare son envoi à un serveur d'outils.

2. Exécute les processus du serveur d'outils

- L'agent envoie la tâche, ainsi que les paramètres structurés, à un serveur d'outils.
- Le serveur d'outils peut alors :
 - Exécutez des scripts ou de la logique dans des systèmes informatiques dédiés (par exemple AWS Lambda, des conteneurs ou Amazon SageMaker)
 - Utiliser son propre sous-agent avec un raisonnement LLM pour sélectionner et exécuter des outils
 - Gérez les dépendances, les nouvelles tentatives ou les flux d'exécution en plusieurs étapes

- Transmettre les résultats à l'agent principal lorsque la tâche est terminée
3. Utilise le raisonnement LLM avec les résultats de l'outil
 - L'agent invoque un LLM, en transmettant la requête d'origine et le résultat du serveur d'outils dans le cadre de l'invite.
 - Le LLM synthétise une réponse qui intègre les informations nouvellement acquises.
 4. Renvoie une réponse
 - L'agent renvoie une réponse structurée ou en langage naturel à l'utilisateur ou au système d'appel.
 - (Facultatif) Les résultats peuvent être stockés dans la mémoire ou dans des journaux d'audit.

Fonctionnalités

- Les outils sont invoqués en dehors de la boucle d'exécution de l'agent principal
- L'exécution de l'outil peut impliquer des appels LLM, des chaînes logiques ou des sous-agents
- L'agent agit en tant que contrôleur ou répartiteur, et pas simplement en tant qu'emballage d'outils
- Permet la composabilité, l'évolutivité et l'isolation de la logique

Cas d'utilisation courants

- Orchestration de chaînes de modèles (par exemple, en combinant LLM, vision et code)
- AI-driven pipelines d'automatisation
- DevOps agents assistants avec exécuteurs de scripts
- Agents de calcul, de simulation ou d'optimisation financiers complexes
- Outils multimodaux (par exemple, en combinant audio, documentation et action)

Directives d'implémentation

Vous pouvez créer ce modèle en utilisant ce qui suit Services AWS :

- Amazon Bedrock (agent hôte et inférence LLM)
- AWS Lambda, Amazon ECS ou Amazon SageMaker endpoints en tant qu'environnement d'exécution du serveur d'outils AWS Fargate
- Amazon API Gateway ou AWS App Runner pour exposer les API du serveur d'outils

- Amazon EventBridge pour une messagerie agent-outil découplée
- AWS Step Functions ou AWS AppFabric pour composer une logique multi-agents sur le serveur d'outils

Résumé

Tool-based les agents qui utilisent des serveurs sont hautement modulaires et évolutifs. Ils dissocient la logique de décision de l'exécution, ce qui permet à l'agent principal de rester léger tout en déléguant des actions complexes ou sensibles à d'autres systèmes. Cela est important pour l'IA agentique de niveau entreprise, en particulier dans les environnements qui nécessitent une gouvernance, une observabilité, une isolation, une composition dynamique ou une combinaison des deux.

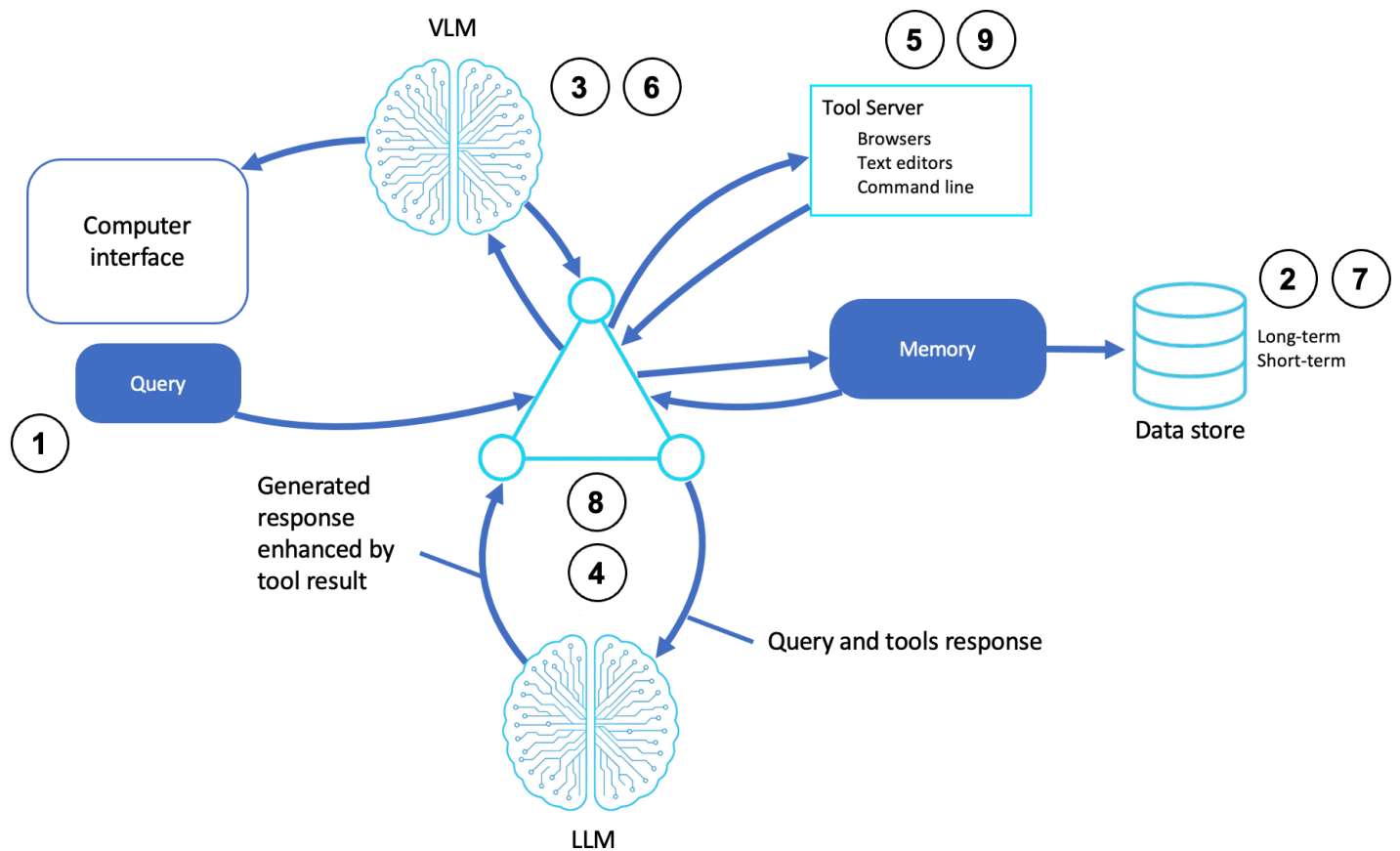
Computer-use agents

Computer-use les agents peuvent simuler ou contrôler des environnements numériques tels que les navigateurs, les terminaux, les systèmes de fichiers et les applications. Ces agents interprètent l'intention de l'utilisateur, interagissent avec les interfaces visuelles et textuelles et exécutent des actions ciblées en combinant le raisonnement LLM, les modèles de langage visuel (VLM) et les serveurs d'outils qui exécutent des commandes ou simulent des événements d'entrée.

Ce modèle est important pour les automatisations pratiques de l'IA, où l'agent fonctionne non seulement comme un assistant, mais également comme un proxy qui exécute des actions comme le ferait un humain, souvent en utilisant les mêmes outils et environnements.

Architecture

Un modèle d'agent utilisé par ordinateur est illustré dans le schéma suivant :



Description

1. Reçoit une requête

- Une tâche ou une demande est fournie via une interface utilisateur, une API ou une interface en langage naturel.

2. Accède à la mémoire

- L'agent récupère la mémoire à court et à long terme pour se souvenir des commandes passées, des objectifs et des états du système.

3. Analyse le contexte visuel

- Un VLM observe l'écran de l'ordinateur, l'état du système ou les éléments de l'interface utilisateur pour comprendre un contexte donné et identifier les éléments exploitables.

4. Raisons grâce à un LLM

- Le LLM combine la requête, l'état de la mémoire, l'outil et la réponse du serveur pour déterminer l'action suivante.

5. Interagit avec le serveur d'outils

- L'agent invoque des outils hébergés sur un serveur, notamment les suivants :
 - Navigateurs (Chrome sans en-tête, par exemple) et environnements shell
 - Éditeurs de texte et de code
 - Interfaces de script personnalisées
- 6. Met à jour les entrées visuelles
 - Si l'interface utilisateur du système change ou si une observation supplémentaire est nécessaire, le VLM peut réanalyser l'état de l'écran ou les tampons de texte.
- 7. Actualise la mémoire
 - Les nouvelles informations, l'état du système ou les commentaires des utilisateurs sont enregistrés dans la mémoire à court et à long terme.
- 8. Formulation des décisions finales et des explications
 - Le LLM synthétise les résultats ou recommande des actions en fonction de la requête et des résultats de l'outil.
- 9. Renvoie une réponse
 - L'agent renvoie les résultats à l'interface (par exemple, une tâche terminée, une confirmation ou du contenu généré).

Fonctionnalités

- Raisonnement multimodal avec entrées visuelles et textuelles
- Contrôle des applications par le biais de données simulées ou d'entrées API-driven
- Gestion de la mémoire pour un état persistant
- Autonomie dans l'exécution séquentielle (flux en plusieurs étapes)

Cas d'utilisation courants

- Développeurs d'IA qui écrivent et exécutent du code dans des IDE
- Computer-use agents pour les flux de travail numériques répétitifs
- Utilisateurs simulés pour les tests logiciels et l'assurance qualité
- Agents d'accessibilité pour naviguer dans les interfaces utilisateur par le biais d'instructions vocales ou de haut niveau
- Automatisation robotique intelligente des processus (RPA) améliorée par le raisonnement

Directives d'implémentation

- Vous pouvez créer ce modèle en utilisant ce qui suit Services AWS :
- Amazon Bedrock pour la LLM-based planification et le raisonnement
- Amazon Elastic Compute Cloud (Amazon EC2) ou ordinateurs portables SageMaker Amazon pour exécuter AWS Lambda des serveurs d'outils avec des environnements d'interface utilisateur simulés
- Amazon Simple Storage Service (Amazon S3) ou Amazon DynamoDB pour la persistance de la mémoire
- Amazon Rekognition (ou modèles personnalisés) pour l'analyse d'images d'interface utilisateur dans des scénarios hybrides
- Amazon CloudWatch Logs ou AWS X-Ray pour l'observabilité et les pistes d'audit

Résumé

Computer-use les agents agissent comme des opérateurs numériques autonomes, comblant le fossé entre les interactions homme-machine et AI-driven les actions. En incorporant de la mémoire, une orchestration d'outils et des VLM, ces agents peuvent interagir de manière adaptative avec des systèmes conçus pour les humains, exécuter des actions, mettre à jour des fichiers, naviguer dans les menus et générer des réponses.

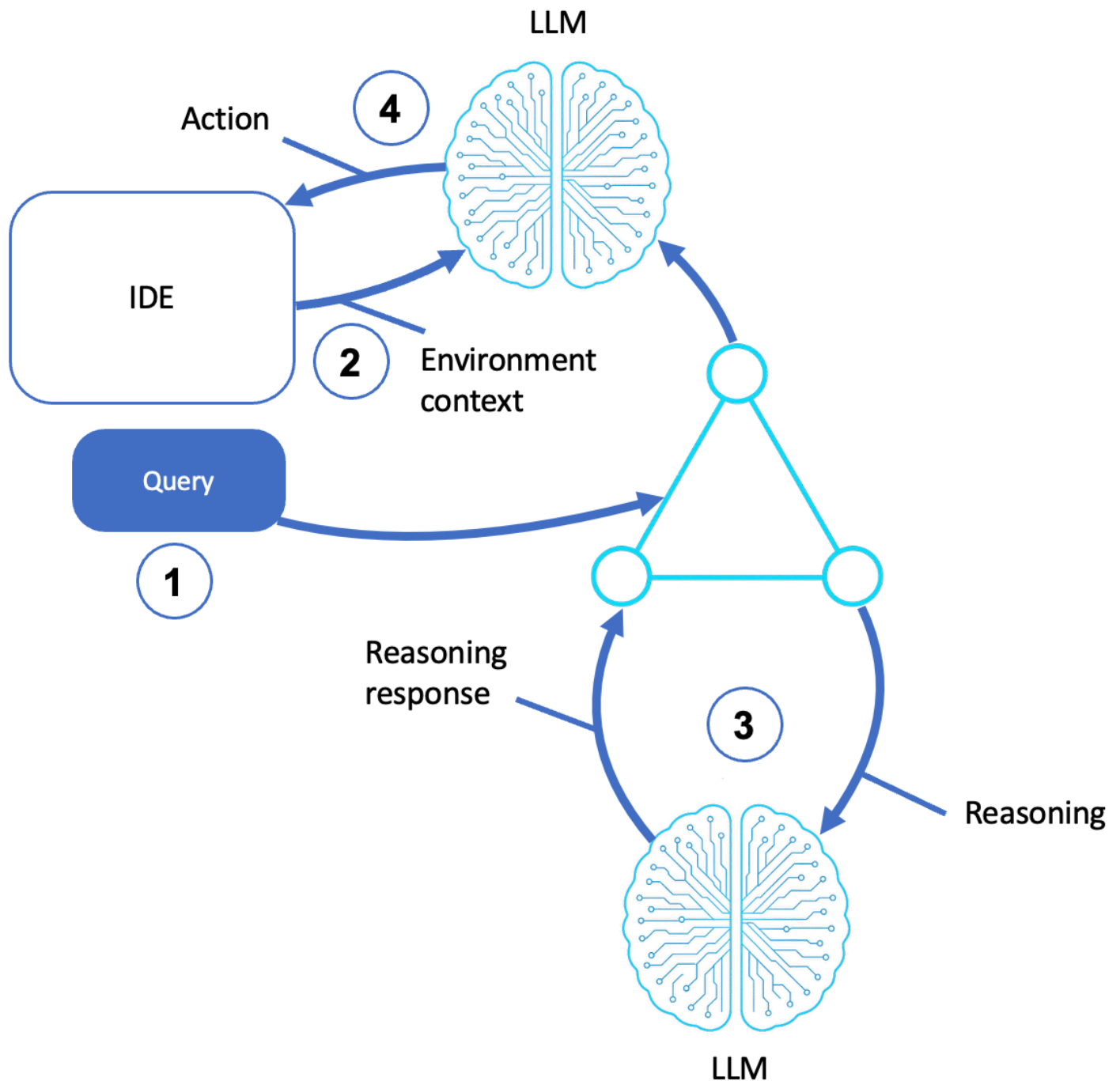
Agents de codage

Les agents de codage peuvent raisonner sur les tâches de programmation, générer ou modifier du code et interagir avec les environnements de développement, tels que les IDE et les CLI. Ces agents associent la compréhension du langage naturel à un raisonnement structuré pour aider, augmenter et automatiser le développement de logiciels, allant de la génération de fonctions à la correction de bogues et à la création de tests.

Contrairement aux outils de saisie semi-automatique, les agents de codage interprètent activement les objectifs des utilisateurs, interrogent l'environnement de développement pour obtenir du contexte (par exemple, il ouvre des fichiers et trace les erreurs), identifient les exigences, puis proposent et exécutent des actions.

Architecture

Un modèle d'agent de codage est illustré dans le schéma suivant :



Description

1. Reçoit une requête

- L'utilisateur fournit des instructions en langage naturel par le biais d'une palette de commandes, d'une fenêtre de discussion ou d'une CLI (par exemple, « Ajouter une journalisation à cette fonction » ou « Refactoriser pour plus de lisibilité »).

2. Extrait le contexte de l'environnement

- L'agent collecte le contexte de l'IDE, notamment les fichiers actifs, la position du curseur, les extraits de code et les tables de symboles.
- Il produit des messages d'erreur, des résultats de test et des sorties provenant d'autres agents.

3. Raisonnement du LLM

- L'agent envoie une invite, y compris la requête et le contexte environnemental, à un LLM.
 - Le LLM effectue une passe de raisonnement pour déterminer ce qui suit :
 - Ce qui doit changer
 - Comment générer une solution
 - Toutes les étapes de refactorisation, de réécriture ou de codage

4. Exécute des actions

- Le LLM renvoie le résultat à l'agent et l'importe dans l'environnement d'exécution ou dans l'environnement d'exécution.
- Cela peut inclure l'insertion ou la modification de code, la génération de commentaires ou de documentation, et le déclenchement de tâches de construction, de test et de linting en aval.

Fonctionnalités

- High-contextual prise de conscience (par exemple, état de l'IDE, curseur et arbre syntaxique)
- Raisonnement itératif des objectifs et du feedback
- Planification du code et séparation des actions facultatives (par exemple, d'abord raisonner puis agir)
- Fonctionne dans des flux de travail de développement synchrones ou asynchrones

Cas d'utilisation courants

- Génération de code à partir des descriptions de tâches
- Refactorisation et optimisation du code
- Test-case génération et validation
- Explications des erreurs et débogage
- Assistants de documentation
- Copilotes de programmation jumelés

Directives d'implémentation

- Vous pouvez créer ce modèle à l'aide des outils suivants et Services AWS :
- Amazon Bedrock pour la LLM-driven génération et le raisonnement
- Amazon Q Developer pour les suggestions de codage et les complétions
- AWS Lambda ou Amazon Elastic Container Service (Amazon ECS) pour exécuter et tester des environnements sandbox
- AWS Cloud9, des extensions VS Code ou des intégrations IDE personnalisées pour héberger et évaluer le contexte
- Amazon Simple Storage Service (Amazon S3) pour stocker les demandes intermédiaires, les réponses et l'historique des révisions

Résumé

Les agents de codage sont de nouveaux outils de AI-powered développement capables d'interpréter le langage naturel, d'analyser le contexte, de générer des modifications de code en plusieurs étapes et de s'intégrer au cycle de vie du développement logiciel.

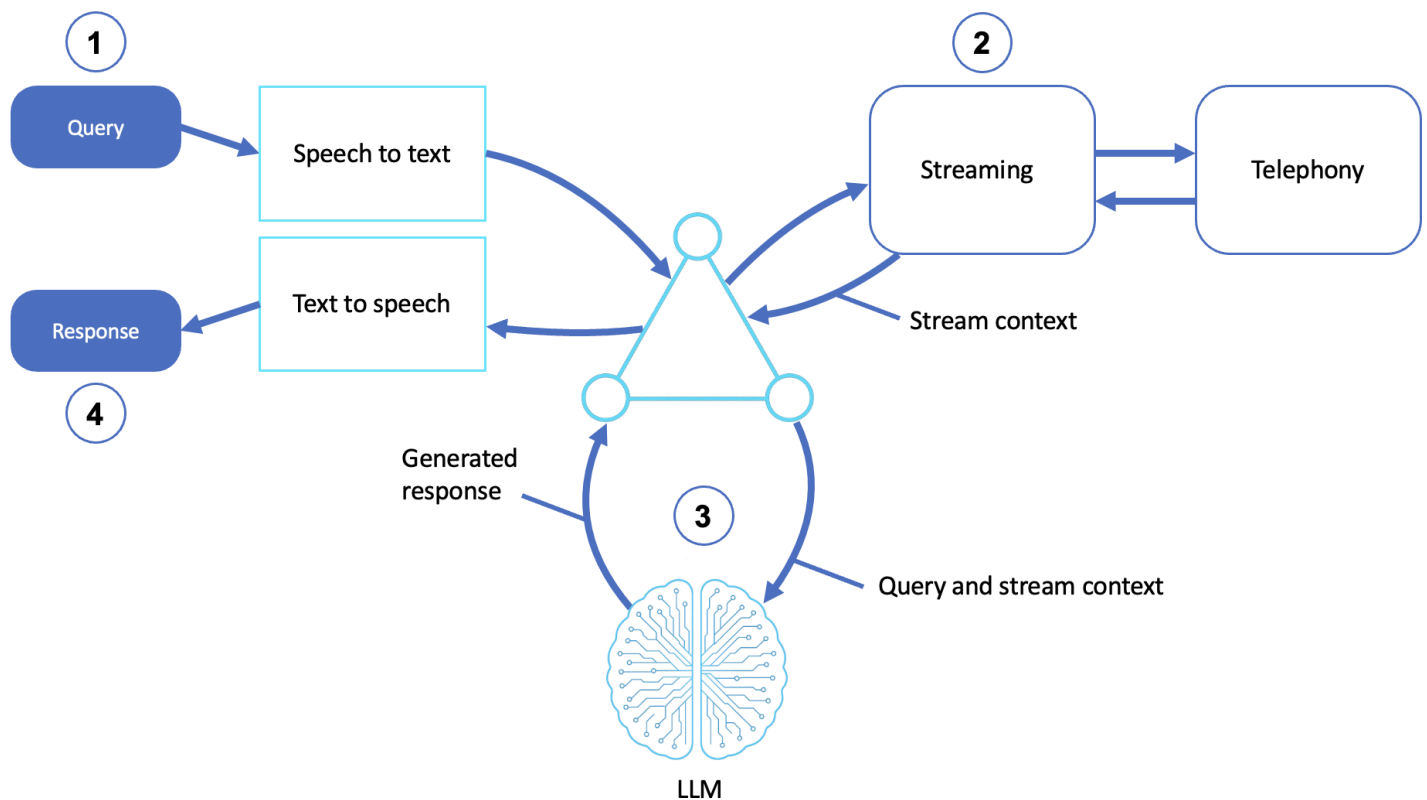
Agents vocaux et vocaux

Les agents vocaux et vocaux interagissent avec les utilisateurs par le biais de dialogues vocaux. Ces agents intègrent la reconnaissance vocale, la compréhension du langage naturel et la synthèse vocale pour permettre l'IA conversationnelle sur la téléphonie, le mobile, le Web et les plateformes intégrées.

Les agents vocaux sont particulièrement efficaces dans les environnements mains libres, en temps réel ou axés sur l'accessibilité. En combinant les interfaces de streaming avec LLM-powered le raisonnement, ils facilitent des interactions riches et dynamiques qui semblent naturelles pour les utilisateurs.

Architecture

Un agent vocal et vocal est illustré dans le schéma suivant :



Description

1. Reçoit une requête vocale

- L'utilisateur adresse une demande à un téléphone, à un microphone ou à un système intégré.
- Un module Speech-to-Text (STT) convertit le son en texte.

2. Intègre le contexte du streaming et de la téléphonie

- L'agent utilise une interface de streaming pour gérer le son I/O en temps réel.
- Si elle est déployée dans un centre d'appels ou dans un contexte de télécommunications, l'intégration de la téléphonie gère le routage des sessions, l'entrée multifréquence bicolore (DTMF) et le transport multimédia.

Remarque : Le DTMF fait référence aux tonalités générées lorsque vous appuyez sur les boutons du clavier d'un téléphone. Dans le contexte du streaming et de l'intégration du contexte téléphonique dans les agents vocaux, le DTMF est utilisé comme mécanisme d'entrée de signal lors d'un appel téléphonique, en particulier dans les systèmes de réponse vocale interactive (IVR). Les entrées DTMF permettent à l'agent de :

- Reconnaissez les sélections de menu (par exemple, « Appuyez sur 1 pour la facturation »). Appuyez sur 2 pour obtenir de l'aide. »)
- Collectez des entrées numériques (par exemple, des numéros de compte, des codes PIN et des numéros de confirmation)
- Déclenchez des workflows ou des transitions d'état dans les flux d'appels
- Passez de la parole à la tonalité tactile si nécessaire

1. Raisons liées au contexte du flux LLM

- La requête est envoyée à l'agent, qui la transmet, avec toutes les métadonnées de session (par exemple, l'identifiant de l'appelant, le contexte antérieur), à un LLM.
- Le LLM génère une réponse, éventuellement en utilisant une stratégie de chaîne de pensée ou une mémoire à plusieurs tours si l'interaction est en cours.

2. Renvoie une réponse vocale

- L'agent convertit sa réponse en parole à l'aide de la synthèse vocale (TTS).
- Il renvoie le son à l'utilisateur via un canal vocal.

Fonctionnalités

- Real-time compréhension et génération de la parole
- Multilingue I/O avec support STT et TTS
- Intégration aux API de téléphonie ou de streaming
- Sensibilisation aux sessions et transfert de mémoire entre les tours

Cas d'utilisation courants

- Systèmes IVR conversationnels
- Réceptionnistes et planificateurs de rendez-vous virtuels
- Voice-driven agents du service d'assistance
- Assistants vocaux portables
- Interfaces vocales pour maisons intelligentes et outils d'accessibilité

Directives d'implémentation

Vous pouvez créer ce modèle à l'aide des outils suivants et Services AWS :

- Amazon Lex V2 ou Amazon Transcribe pour STT
- Amazon Polly pour TTS
- Amazon Chime SDK, Amazon Connect Customer ou Amazon Interactive Video Service (Amazon IVS) pour le streaming et la téléphonie
- Amazon Bedrock pour raisonner avec Anthropic, AI21 ou d'autres modèles de base
- AWS Lambda pour connecter STT, LLM, TTS et le contexte de session

(Facultatif) Les améliorations supplémentaires peuvent inclure les suivantes :

- Amazon Kendra ou OpenSearch pour un RAG sensible au contexte
- Amazon DynamoDB pour la mémoire de session
- Amazon CloudWatch Logs et AWS X-Ray pour la traçabilité

Résumé

Les agents vocaux et vocaux sont des systèmes intelligents qui interagissent par le biais de conversations naturelles. En intégrant les interfaces vocales au raisonnement LLM et à l'infrastructure de streaming en temps réel, les agents vocaux permettent des interactions fluides, accessibles et évolutives.

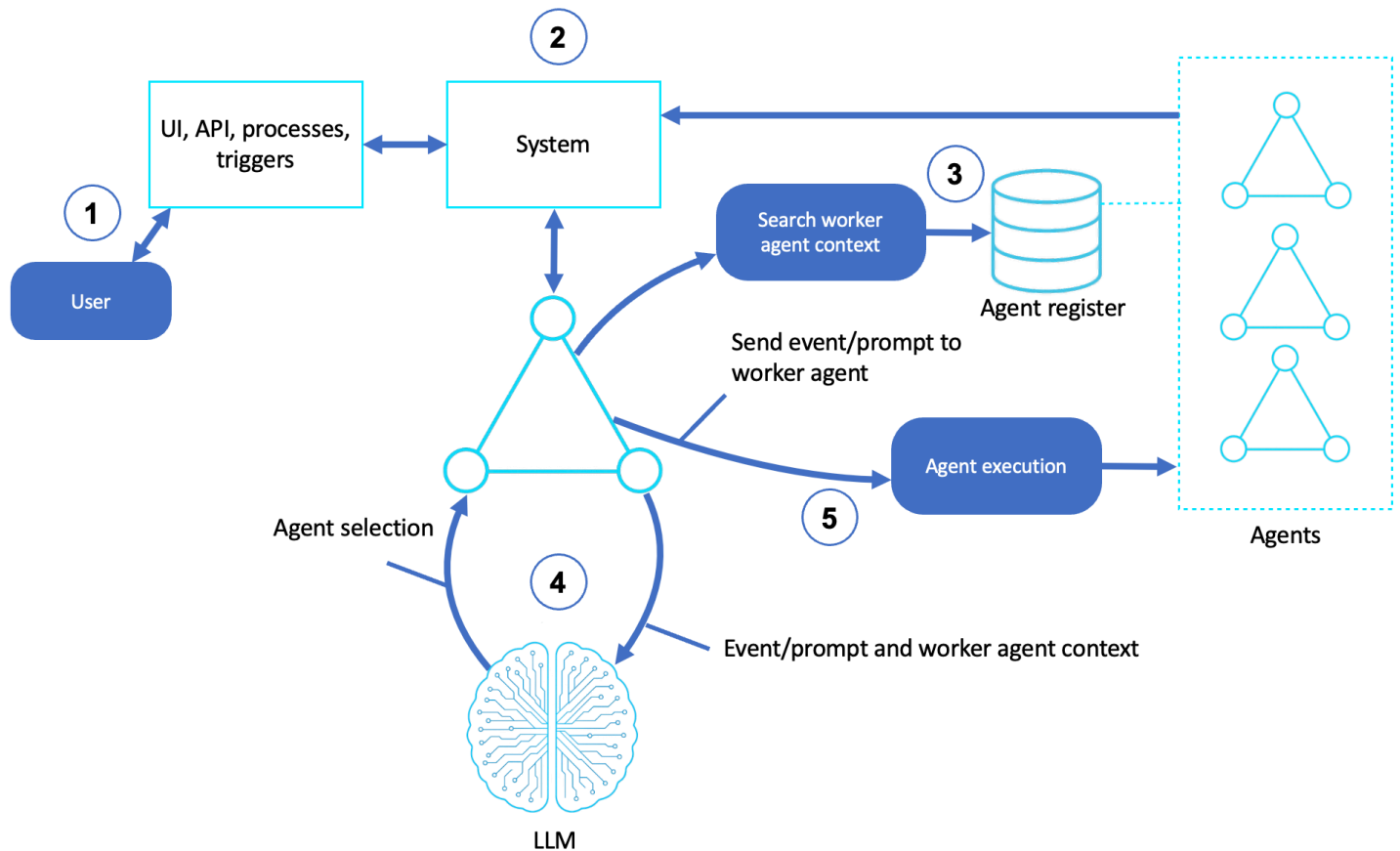
Agents d'orchestration des flux de travail

Les agents d'orchestration des flux de travail gèrent et coordonnent les tâches, les processus et les services en plusieurs étapes sur les systèmes distribués. Plutôt que de raisonner et d'agir de manière isolée, ces agents délèguent le travail à des sous-agents ou à d'autres systèmes, maintiennent le contexte d'exécution et s'adaptent en fonction des résultats intermédiaires.

Ces agents constituent un élément fondamental des flux d'automatisation. Ils sont particulièrement utiles lors de la gestion de tâches de longue durée, de compositions multi-agents et d'intégrations entre domaines où différents agents et outils doivent être appelés en séquence ou de manière conditionnelle.

Architecture

Un agent d'orchestration de flux de travail est illustré dans le schéma suivant :



Description

1. Reçoit les entrées utilisateur

- Un utilisateur (ou un déclencheur externe) lance une tâche via une interface utilisateur, une API ou un événement système.

2. Gère les événements du système

- Un composant du système reçoit la demande et émet un événement ou une commande nécessitant une orchestration.

3. Récupère le contexte

- L'agent de flux de travail interroge les bases de connaissances et les registres d'agents afin de trouver l'agent de travail adapté à la tâche en fonction des métadonnées, du domaine et du taux de réussite antérieur.

4. Sélectionne un agent LLM

- Un LLM permet de sélectionner le meilleur agent ou le meilleur plan de flux de travail en analysant la description de la tâche et les options disponibles.
- Il peut également formuler des instructions spécifiques à une tâche à envoyer à un agent sélectionné.

5. Délègue et exécute

- L'agent de travail choisi reçoit l'événement ou l'invite et commence à exécuter des commandes.
- Il peut suivre l'état d'exécution, réessayer en cas d'échec et transmettre les résultats intermédiaires à l'agent suivant de la séquence.

Fonctionnalités

- Composition des agents (par exemple, superviseurs, agents collaborateurs et outils)
- Event-driven ou exécution planifiée
- Suivi de la mémoire et de l'état au fil du temps
- Orchestration hiérarchique ou parallèle des tâches (synchrone par rapport aux flux de travail asynchrones)
- Sélection et chaînage dynamiques des agents

Cas d'utilisation courants

- Automatisation en plusieurs étapes (par exemple, ingestion de données et création de rapports)
- Routage et escalade du service client (par exemple, agent-en-coordonateur)
- Les agents d'IA coordonnent leurs activités avec les humains et les robots au sein de la même boucle
- Automatise les processus d'entreprise à l'aide LLM-powered de la logique
- Les systèmes hybrides combinent des agents d'intelligence artificielle et des outils d'orchestration traditionnels

Directives d'implémentation

Vous pouvez créer ce modèle à l'aide des outils suivants et Services AWS :

- Amazon Bedrock pour le raisonnement et la sélection des agents

- AWS Step Functions ou Amazon EventBridge pour la composition du flux de travail
- AWS Lambda en tant qu'unités d'exécution ou exécuteurs de tâches
- Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3) ou Amazon RDS pour suivre les états et les résultats
- AWS AppFabric ou Amazon AppFlow pour la coordination entre systèmes
- (Facultatif) Utilisez Amazon SageMaker Run Agent pour héberger des agents de travail spécifiques à un domaine

Résumé

Les agents de flux de travail coordonnent, adaptent et alignent les objectifs dans les environnements multi-agents. Cela signifie que les agents d'IA peuvent collaborer, s'adapter aux conditions d'exécution et obtenir des résultats complexes grâce à des flux de travail modulaires et explicables.

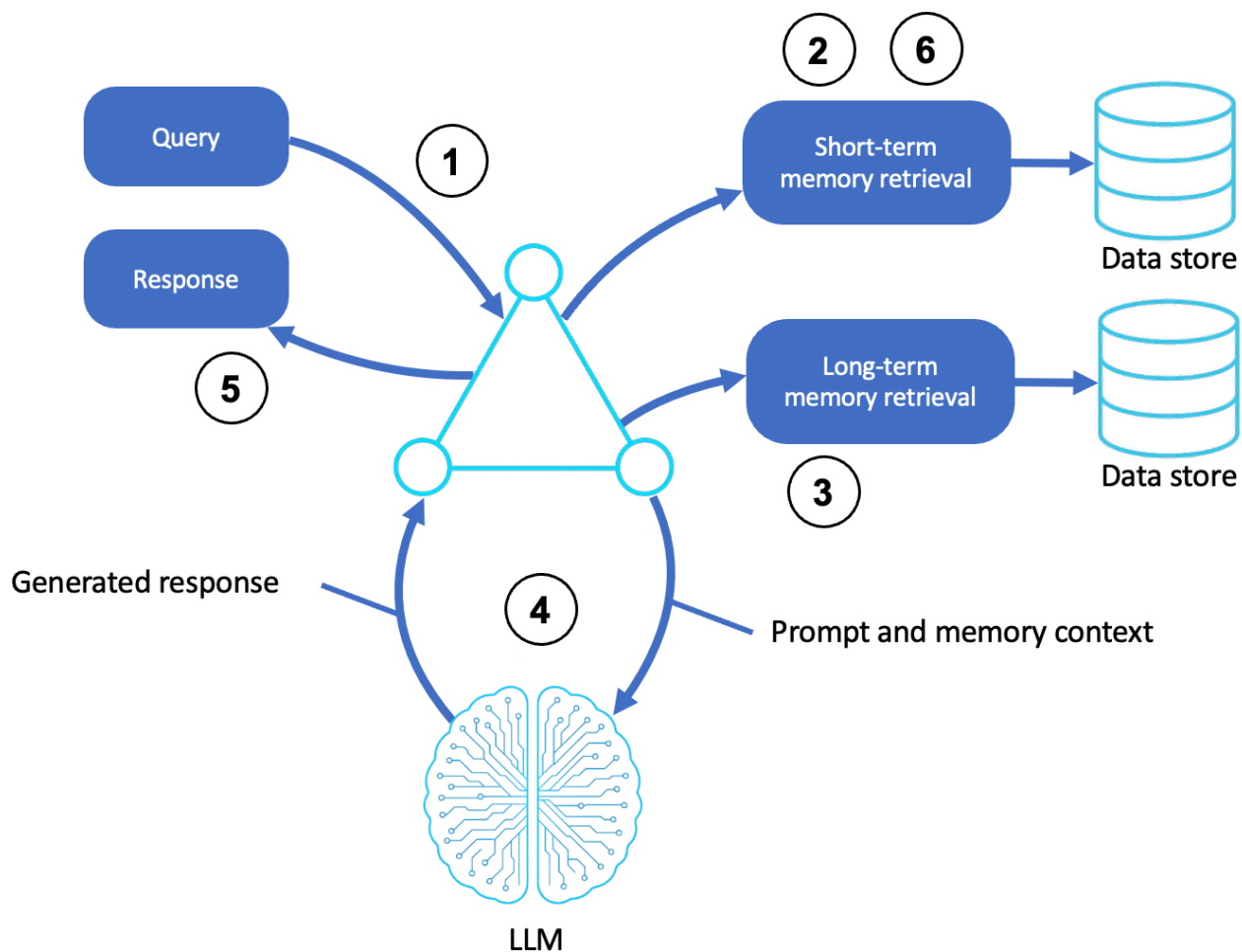
Memory-augmented agents

Memory-augmented les agents sont améliorés grâce à la capacité de stocker, de récupérer et de raisonner en utilisant la mémoire à court et à long terme. Cela leur permet de maintenir le contexte entre plusieurs tâches, sessions et interactions, ce qui produit des réponses plus cohérentes, personnalisées et stratégiques.

Contrairement aux agents apatrides, les agents à mémoire augmentée s'adaptent en faisant référence à des données historiques, en tirant des leçons des résultats antérieurs et en prenant des décisions conformes aux objectifs, aux préférences et à l'environnement de l'utilisateur.

Architecture

Un agent à mémoire augmentée est illustré dans le schéma suivant :



Description

1. Reçoit une entrée ou un événement

- L'agent reçoit une requête utilisateur ou un événement système. Il peut s'agir d'un texte, d'un déclencheur d'API ou d'un changement environnemental.

2. Récupère la mémoire à court terme

- L'agent récupère l'historique des conversations récentes, le contexte des tâches ou l'état du système correspondant à la session ou au flux de travail.

3. Récupère la mémoire à long terme

- L'agent interroge la mémoire à long terme (par exemple, les bases de données vectorielles et les banques de valeurs clés) pour obtenir des informations historiques, telles que les suivantes :
 - Préférences de l'utilisateur
 - Décisions et résultats antérieurs

- Concepts, résumés ou expériences apprises

4. Raisons motivées par le LLM

- Le contexte de mémoire est intégré à l'invite LLM, ce qui permet à l'agent de raisonner en fonction à la fois des entrées actuelles et des connaissances antérieures.

5. Génère des sorties

- L'agent produit une réponse, un plan ou une action adaptés au contexte qui sont personnalisés en fonction de l'historique des tâches et des entrées de l'utilisateur.

6. Actualise la mémoire

- Les nouvelles informations, telles que les objectifs actualisés, les signaux de réussite et d'échec et les réponses structurées, sont stockées pour les tâches futures.

Fonctionnalités

- Continuité des sessions entre les conversations ou les événements
- Persistance des objectifs dans le temps
- Conscience contextuelle basée sur un état en évolution
- Adaptabilité éclairée par les succès et les échecs antérieurs
- Personnalisation alignée sur les préférences et l'historique de l'utilisateur

Cas d'utilisation courants

- Copilotes conversationnels qui mémorisent les préférences de l'utilisateur
- Agents de codage qui suivent les modifications de la base de code
- Des agents de flux de travail qui s'adaptent en fonction de l'historique des tâches
- Des jumeaux numériques qui évoluent à partir de la connaissance du système
- Des agents de recherche qui évitent les récupérations redondantes

Implémentation d'agents à mémoire augmentée

Utilisez les outils suivants et Services AWS pour les agents à mémoire augmentée :

Couche mémoire	Service AWS	Objectif
----------------	-------------	----------

Short-term	Contexte Amazon DynamoDB, Redis, Amazon Bedrock	Récupération rapide des états d'interaction récents
Long-term (structuré)	Amazon Aurora, Amazon DynamoDB, Amazon Neptune	Faits, relations et journaux
Long-term (sémantique)	OpenSearch, PostgreSQL, Pinecone	Embedding-based récupération (c'est-à-dire RAG)
Stockage	Amazon S3	Stockage de transcriptions, de mémoires structurées et de fichiers
Orchestration	AWS Lambda ou AWS Step Functions	Gestion de l'injection de mémoire et du cycle de mise à jour
Raisonnement	Amazon Bedrock	Claude ou Mistral anthropique avec des instructions de mémorisation

Implémentation de l'envoi de messages par injection de mémoire

Pour intégrer la mémoire dans le raisonnement des agents, utilisez une combinaison d'états structurés et d'injection de contexte augmentée par extraction :

- Incluez l'état le plus récent de l'agent et l'historique récent des dialogues en tant qu'entrée structurée lors de la création de l'invite pour le modèle de langage, afin que celui-ci puisse raisonner en tenant compte du contexte complet.
- Utilisez la génération augmentée par extraction (RAG) pour extraire des documents ou des faits pertinents de la mémoire à long terme.
- Résumez les plans, le contexte et les interactions précédents à des fins de compression et de pertinence.
- Injectez des modules de mémoire externes, tels que des magasins de vecteurs ou des journaux structurés, lors de l'inférence pour guider la prise de décision.

Résumé

Memory-augmented les agents maintiennent une continuité de pensée en tirant des leçons de leur expérience et en mémorisant le contexte de l'utilisateur. Ces agents surpassent l'intelligence réactive en utilisant la collaboration à long terme, la personnalisation et le raisonnement stratégique. En termes d'IA agentic, la mémoire permet aux agents de se comporter davantage comme des homologues numériques adaptatifs que comme des outils apatrides.

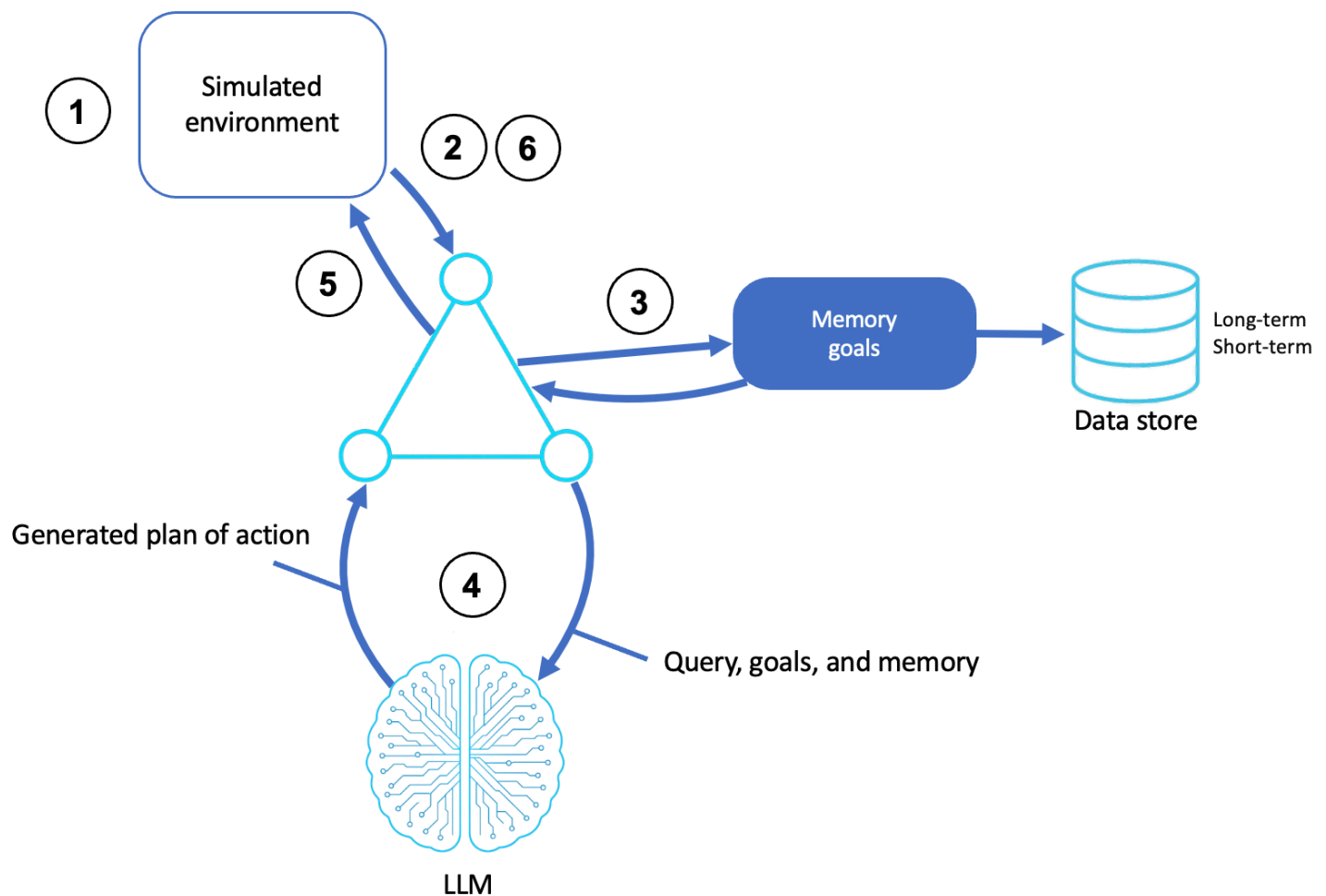
Agents de simulation et de banc d'essai

Les agents de simulation et de banc d'essai opèrent dans des environnements virtualisés ou contrôlés dans lesquels ils raisonnent, agissent et apprennent. Ces agents simulent le comportement, modélisent les résultats et entraînent des stratégies dans des contextes répétables avant de les appliquer à des environnements réels.

Ce modèle est utile pour le développement itératif, l'apprentissage par renforcement (RL), l'évaluation de la prise de décision autonome et les tests comportementaux émergents. Les agents de simulation fonctionnent souvent en boucle fermée, reçoivent des informations de leur environnement et ajustent leur comportement en conséquence, ce qui les rend essentiels pour les tâches impliquant un raisonnement spatial, un contrôle en temps réel ou une dynamique de système complexe.

Architecture

Le schéma suivant montre un agent de simulation ou un agent de banc d'essai :



Description

1. Lance un environnement

- L'agent lance un environnement simulé (par exemple, un monde 3D, un moteur physique, un bac à sable CLI ou un flux de données synthétiques).
- L'agent est chargé dans l'environnement avec une tâche, un objectif ou une politique initiale.

2. Perçoit l'agent

- L'agent perçoit l'état actuel par le biais de la télémétrie de simulation (par exemple, émulation de capteurs, caméra virtuelle et journaux structurés).

3. Récupère l'objectif et la mémoire

- L'agent récupère l'objectif, les instructions du scénario ou l'objectif contextuel qui lui ont été assignés.
- Il peut également récupérer de la mémoire antérieure, notamment les éléments suivants :

- Long-term stratégies ou politiques
- Cartes environnementales ou contraintes connues
- Succès ou échecs antérieurs issus de simulations similaires

4. Raisons et plans

- Un LLM interprète l'état simulé, les objectifs de la tâche et les connaissances apprises.
- Il génère un plan d'action ou une commande de contrôle.

5. Exécute des actions simulées

- L'agent exécute le plan, modifie l'état, navigue dans l'espace ou interagit avec des entités virtuelles.

6. Apprend

- L'agent évalue les résultats des actions
- En fonction de la configuration de l'agent, celui-ci peut effectuer les opérations suivantes :
 - Exécuter une URL
 - Enregistrez les résultats pour les ajustements futurs
 - Adaptez les stratégies en temps réel

Fonctionnalités

- Fonctionne dans des environnements synthétiques ou virtuels
- Soutient l'apprentissage par essais et erreurs, l'affinement des politiques et la modélisation du système
- Low-risk tests de comportement, de gestion des défaillances et de cas extrêmes
- Permet l'analyse du comportement des agents émergents dans les configurations multi-agents
- Supporte à la fois le contrôle en boucle fermée et l'exploration par l'homme en boucle

Cas d'utilisation courants

- Apprentissage par renforcement pour la robotique, les drones et les jeux vidéo
- Formation aux véhicules autonomes sur des routes virtuelles
- UI ou CLI simulées pour des scénarios de banc DevOps d'essai
- Expériences comportementales émergentes dans le cadre de simulations sociales

- Validation de sécurité de la logique de décision avant la production

Directives d'implémentation

Vous pouvez créer un agent de simulation et de banc d'essai à l'aide des outils suivants et : Services AWS

Composant	Service AWS	Objectif
Environnement	Amazon ECS, Amazon EC2 ou un simulateur personnalisé dans le laboratoire Amazon Studio SageMaker	Exécutez des mondes virtuels (Gazebo, Unity, Unreal) ou des CLI sandbox
Logique de l'agent	Amazon Bedrock SageMaker, Amazon ou AWS Lambda	LLM-based planificateurs ou agents RL
Boucle de rétroaction	Amazon SageMaker Reinforce ment Learning CloudWatch, Amazon ou journaux personnalisés	Suivi des récompenses, évaluation des résultats et enregistrement des comportements
Mémoire et rediffusion	Amazon S3, Amazon DynamoDB ou Amazon RDS	État persistant, historique des épisodes ou données de scénario
Visualisation	Tableaux de CloudWatch bord Amazon ou blocs-notes Amazon SageMaker	Observez les changements de politique, les résultats et les indicateurs de formation

Les applications supplémentaires suivantes sont les suivantes :

- [AWS SimSpace Weaver](#) pour des simulations spatiales à grande échelle
- [AWS IoT Core](#) pour tester les appareils fantômes
- [Amazon SageMaker Experiments](#) pour l'évaluation et l'analyse comparative des agents

Résumé

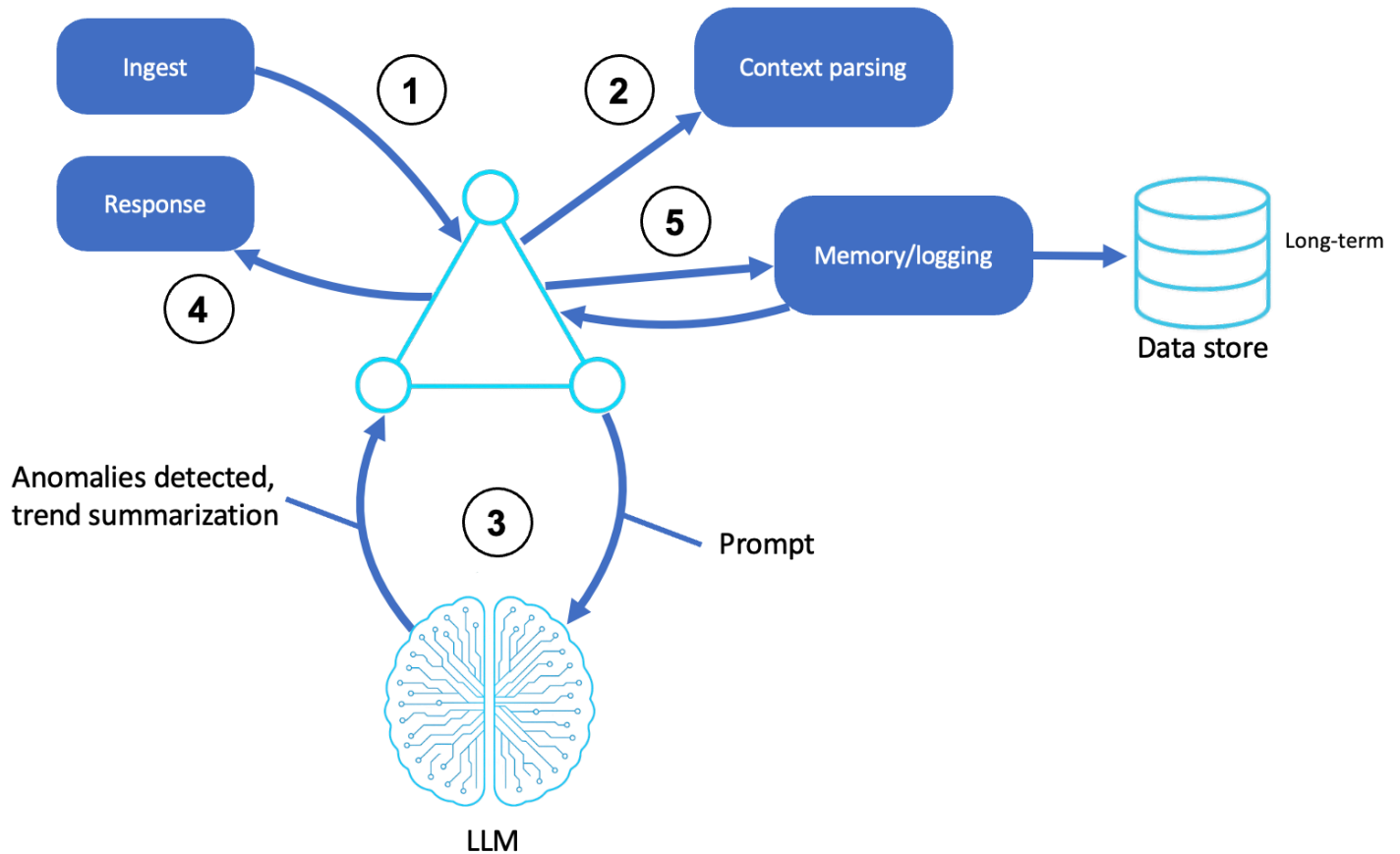
Les agents de simulation et de banc d'essai sont destinés à une exploration structurée avant d'être déployés dans les systèmes de production. Utilisez ces agents pour élaborer des politiques de navigation autonomes, tester des processus métier dans des environnements synthétiques et évaluer les essais pour déterminer les modèles de coordination.

Observateurs et agents de surveillance

Les observateurs et les agents de surveillance observent passivement les systèmes, les environnements et les interactions afin de détecter des modèles, de générer des informations et de déclencher des actions. En tant qu'observateurs intelligents, ils améliorent les alertes, les diagnostics et les audits sans déclencher directement de comportement.

Ces agents excellent là où la surveillance traditionnelle manque d'adaptabilité ou de raisonnement, en particulier pour la AI-in-the-loop surveillance, la détection des anomalies, le contrôle de la conformité et le renseignement de sécurité. Les agents observateurs sont des auditeurs d'événements qui surveillent en permanence la télémétrie du système et les interactions avec les utilisateurs. L'agent dépend de la perception, de l'interprétation et de l'escalade conditionnelle ou du reporting.

Architecture



Description

1. Ingérer la télémétrie

- L'agent reçoit des informations provenant d'une ou de plusieurs sources système, telles que les suivantes :
 - Logs (application, infrastructure, sécurité)
 - Métriques (performances, latence, utilisation)
 - Événements (appels d'API, actions des utilisateurs, données des capteurs)

2. Analyser le contexte

- Les entrées brutes sont analysées, structurées et enrichies de métadonnées, telles que l'horodatage, l'identité de l'acteur, l'état du système et l'identifiant de trace.

3. Les raisons d'utiliser le LLM

- L'agent utilise un LLM ou un module logique pour interpréter les entrées analysées en identifiant les anomalies, en résumant les tendances et en établissant des corrélations entre des traces distribuées ou des fenêtres temporelles.
4. Classer ou alerter
- L'agent détermine si le comportement observé justifie ce qui suit :
 - Une alerte ou une escalade
 - Mise à jour du rapport ou du tableau de bord
 - Un déclencheur de réponse (par exemple, correction automatique et application des politiques)
5. Mémoire journal ou boucles de rétroaction
- L'agent enregistre les événements et les décisions à des fins d'apprentissage à long terme, d'audits ou de futures références pour d'autres agents.

Fonctionnalités

- Passif et non invasif (l'agent n'agit pas directement)
- Hautement évolutif et asynchrone
- AI-driven corrélation entre signaux bruyants ou distribués
- Soutient l'audit, la conformité et les informations en temps réel
- Peut alimenter les agents en aval ou les flux de travail humains

Cas d'utilisation courants

- AI-augmented observabilité pour les microservices et les API
- Surveillance de la dérive du modèle, de la violation des règles ou du comportement hors bande
- Analyse de l'activité des clients ou résumés des interactions
- Agents de révision du code qui surveillent les validations ou les déploiements
- Surveillance des journaux de sécurité ou de conformité à l'aide du raisonnement LLM

Directives d'implémentation

Vous pouvez créer un observateur et un agent de surveillance à l'aide des outils suivants et Services AWS :

Composant	Service AWS	Objectif
Ingestion d'événements	Amazon EventBridge, Amazon CloudWatch Logs, Amazon Kinesis, Amazon S3	Ingérer des données de télémétrie structurées et non structurées
Prétraitement	AWS Lambda, AWS Glue, AWS Step Functions	Transformez les données brutes en instructions structurées
Moteur de raisonnement	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Analysez les événements, classifiez les comportements, générez des informations
Stockage et mémoire	Amazon S3, Amazon DynamoDB, OpenSearch	Observations, résumés et résultats persistants
Alerte et escalade	Amazon SNS, AWS AppFabric, Amazon EventBridge	Déclenchez des systèmes ou des agents en aval

Les applications supplémentaires suivantes sont les suivantes :

- [AWS Security Hub CSPM](#) pour la surveillance des journaux de sécurité
- [Amazon Quick](#) pour visualiser les résultats des agents

Résumé

Les observateurs et les agents de surveillance suivent les systèmes et les comportements en temps réel. Ils détectent les anomalies, audient la sécurité et collectent des informations opérationnelles en identifiant les modèles que les humains ou les règles pourraient ignorer. Cette fonctionnalité permet de créer des systèmes capables de s'adapter aux conditions changeantes et de prendre des décisions basées sur une analyse complète des données.

Multi-agent collaboration

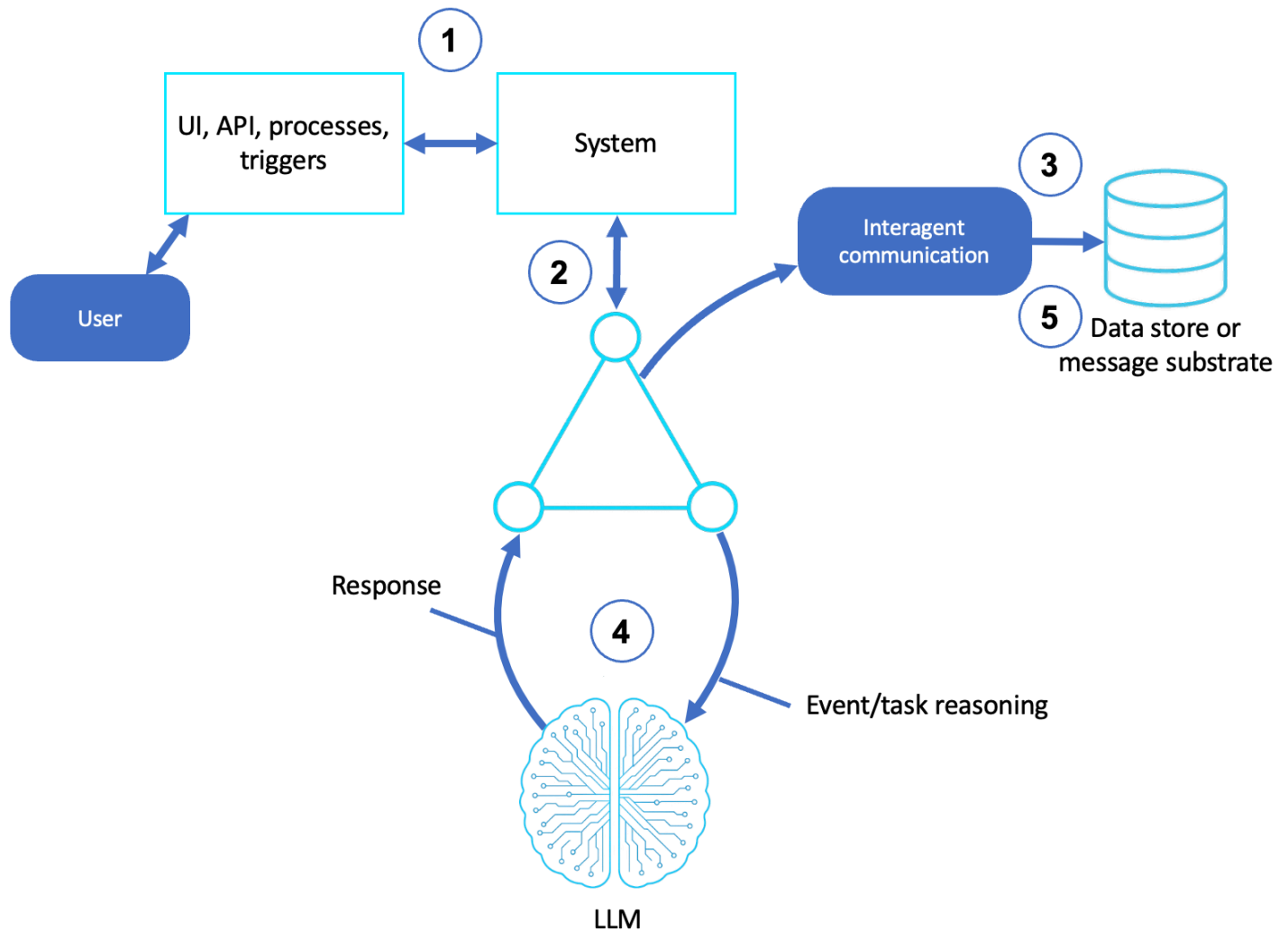
Multi-agent la collaboration fait référence à un modèle dans lequel plusieurs agents autonomes, chacun ayant un rôle, une spécialisation ou un objectif distinct, négocient pour résoudre des tâches complexes. Ces agents peuvent agir indépendamment ou avec d'autres agents en partageant des informations, en répartissant les responsabilités et en raisonnant collectivement en vue d'un objectif.

Ce modèle diffère de celui des agents de flux de travail, qui coordonnent et délèguent des tâches de manière centralisée aux agents subordonnés dans un flux structuré. En revanche, la collaboration multi-agents met l'accent sur la coordination entre pairs ou émergente en permettant l'adaptabilité, le parallélisme et la division de la cognition. Le tableau suivant compare la collaboration multi-agents avec les agents de flux de travail :

Fonctionnalité	Agents de workflow	Objectif
Contrôle	Coordinateur centralisé	Des pairs décentralisés, distribués ou basés sur les rôles
Interaction	Un agent délègue et suit l'exécution	Plusieurs agents négocient, partagent et s'adaptent
Conception	Séquence de tâches prédéfinie	Répartition des tâches innovante et flexible
Coordination	Orchestration procédurale	Interactions coopératives ou compétitives
Cas d'utilisation	Automatisation des processus d'entreprise	Raisonnement complexe, exploration et stratégies émergentes

Architecture

Le schéma suivant illustre la collaboration entre plusieurs agents :



Description

1. Lance une tâche

- Un utilisateur ou un système émet un objectif ou un problème de haut niveau.
- Un agent « manager » ou un contexte initiateur définit l'objectif.

2. Attribue ou découvre des rôles

- Les agents s'attribuent eux-mêmes (logique symbolique ou raisonnement) ou sont délégués (courtier d'événements) à d'autres rôles, tels que ceux de planificateur, de chercheur, d'exécuteur testamentaire, de critique ou d'explicateur.

3. Communique avec les autres agents

- Les agents communiquent via la mémoire partagée, les files d'attente de messagerie ou le chaînage rapide.

- Ils peuvent débattre, se poser des questions ou se proposer des sous-tâches.
4. Utilise un raisonnement spécialisé
 - Chaque agent utilise son propre modèle ou sa propre logique de domaine pour résoudre la partie du problème qui lui est propre.
 - Les agents peuvent utiliser des LLM avec des instructions et de la mémoire spécifiques aux rôles.
 5. Coordonne les résultats ou les objectifs
 - Les agents synthétisent les contributions dans une réponse, un plan ou une action finale.
 - (Facultatif) Un agent de supervision peut valider ou résumer le résultat synthétisé.

Fonctionnalités

- Peer-level agents dotés de rôles ou de compétences spécialisés
- Comportement émergent par le biais de la communication ou de la négociation
- Traitement parallèle de problèmes complexes ou multidimensionnels
- Soutient la délibération, l'autocorrection et l'itération réflexive
- Modélisez la dynamique sociale, la collaboration scientifique ou les rôles des équipes d'entreprise

Cas d'utilisation courants

- Équipes de recherche autonomes (agent de recherche, synthétiseur et validateur)
- Développement de logiciels (planificateur, codeur et testeur)
- Modélisation de scénarios commerciaux (finances, politiques et conformité)
- Négociation, appel d'offres ou raisonnement multipartite
- Tâches multimodales (image, texte et logique)

Directives d'implémentation

Vous pouvez créer un système multi-agents à l'aide des outils suivants et Services AWS :

Composant	Service AWS	Objectif
-----------	-------------	----------

Hébergement d'agents	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Héberger des LLM-driven agents individuels
Couche de communication	Amazon SQS, Amazon, EventBridge AWS AppFabric	Messagerie et coordination entre les agents
Mémoire partagée	Amazon DynamoDB, Amazon S3 ou OpenSearch	Multi-agent système de mémoire ou de tableau noir
Couche d'orchestration	AWS Step Functions, AWS Lambda oléoducs	Logique de lancement, de délai d'expiration, de repli et de nouvelle tentative
Identification de l'agent	Agents Amazon Bedrock (définis par rôle) et API AWS AppConfig Amazon Bedrock Converse (agents extérieurs à Amazon Bedrock)	Role-based Invocation d'un outil ou d'un agent et applicati on des limites
Interaction émergente	EventBridge Pipelines ou registres d'agents Amazon	Activer le routage ou l'escalad e dynamique des tâches

Résumé

Multi-agent la collaboration répartit les tâches de résolution de problèmes entre des agents modulaires et axés sur les rôles. Contrairement à l'orchestration des flux de travail, les modèles de collaboration utilisent l'intelligence émergente, la résilience et l'évolutivité qui reflètent la façon dont les humains résolvent les problèmes. Il est particulièrement utile pour les domaines ouverts, les tâches créatives, le raisonnement multimodal et les environnements qui bénéficient de perspectives diverses.

Conclusion

Les modèles évoqués précédemment illustrent les approches fondamentales des implémentations réelles de l'IA agentique. Du raisonnement de base à l'intelligence augmentée par la mémoire, chaque modèle est configuré de manière unique pour la perception, la cognition et l'action, sur la base de l'autonomie, de l'asynchronie et du libre arbitre.

Ces modèles partagent des vocabulaires et des plans techniques pour créer des systèmes intelligents orientés vers des objectifs. Qu'un modèle soit intégré dans une interface utilisateur, orchestré via des services cloud ou coordonné entre des équipes d'agents, chaque modèle est adaptable et modulaire.

Plats à emporter

- Les modèles d'agents sont composables : la plupart des agents du monde réel combinent deux modèles ou plus (par exemple, un agent vocal doté d'un raisonnement et d'une mémoire basés sur des outils).
- La conception de l'agent est contextuelle : choisissez des modèles en fonction de la surface d'interaction, de la complexité des tâches, de la tolérance à la latence et des contraintes spécifiques au domaine.
- AWS la mise en œuvre native est réalisable — Avec Amazon Bedrock SageMaker, Amazon,, AWS Lambda AWS Step Functions, et les architectures pilotées par les événements, chaque modèle d'agent peut être fourni à grande échelle.

Flux de travail LLM

En ce qui concerne les modèles d'agents, nous avons exploré les modèles d'agents d'IA courants, chacun étant construit autour d'un ensemble de capacités modulaires : perception, action, apprentissage et cognition. Au cœur du module cognitif de nombreux modèles d'agents se trouve un grand modèle de langage (LLM) capable de raisonner, de planifier et de prendre des décisions. Cependant, l'invocation d'un LLM à elle seule n'est pas suffisante pour produire un comportement intelligent et ciblé.

Pour effectuer des tâches complexes de manière fiable, les agents doivent intégrer le LLM dans un flux de travail structuré, dans lequel les capacités du modèle sont augmentées grâce à des outils, à de la mémoire, à des boucles de planification et à une logique de coordination. Ces flux de travail LLM permettent à un agent de définir des objectifs, d'acheminer des sous-tâches, d'appeler des services externes, de réfléchir aux résultats et de se coordonner avec d'autres agents.

Ce chapitre présente les modèles de conception de base pour créer des modules cognitifs robustes, extensibles et intelligents pilotés par le LLM, organisés autour de flux de travail réutilisables.

Dans cette section

- [Vue d'ensemble de la cognition augmentée par le LLM](#)
- [Flux de travail pour un chaînage rapide](#)
- [Flux de travail pour le routage](#)
- [Workflow pour la parallélisation](#)
- [Flux de travail pour l'orchestration](#)
- [Flux de travail pour les évaluateurs et boucles d'affinage par réflexion](#)
- [Conclusion](#)

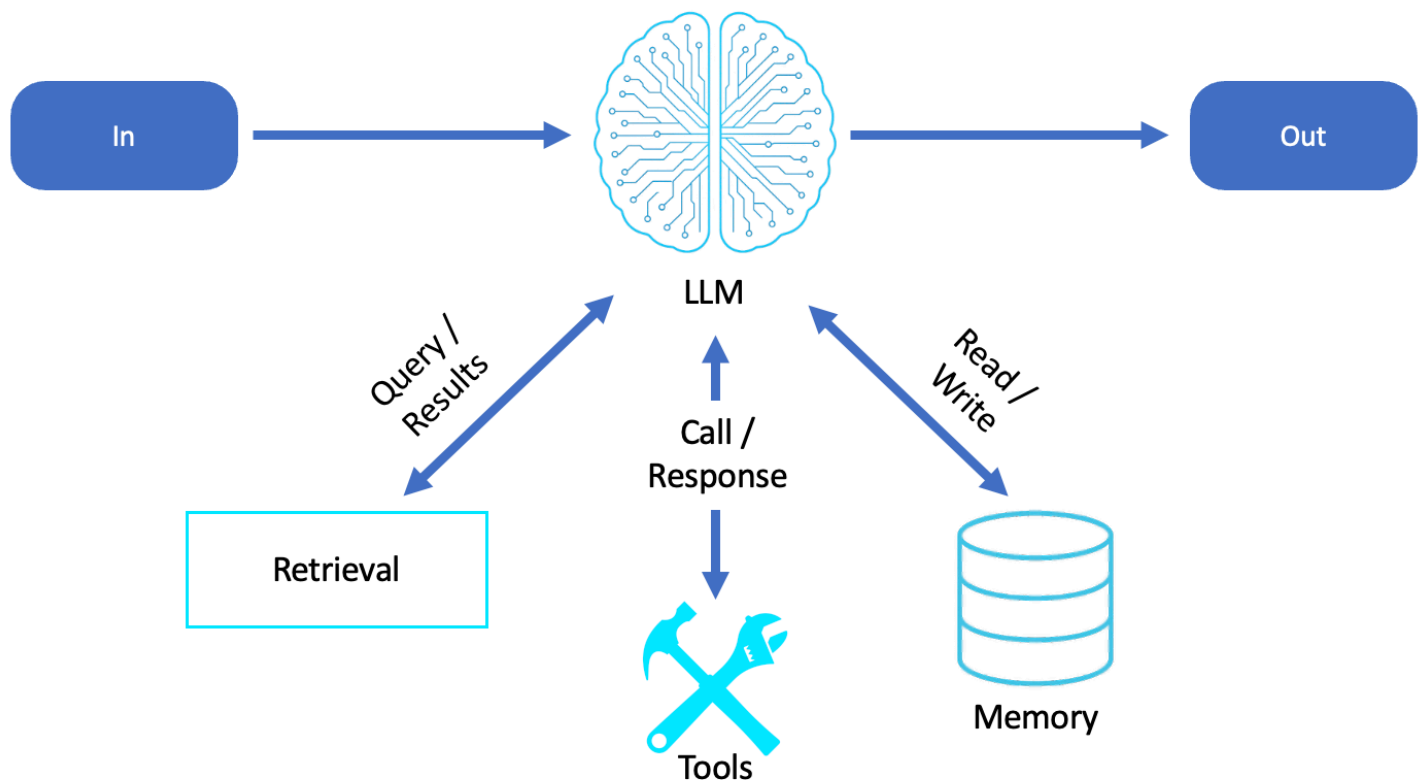
Vue d'ensemble de la cognition augmentée par le LLM

À la base, le module cognitif d'un agent logiciel peut être considéré comme un LLM assorti d'augmentations. L'agent peut utiliser les éléments de base suivants pour raisonner efficacement dans son environnement :

- Invitation — Cadrage de la saisie à l'aide du contexte, des instructions, des exemples et de la mémoire

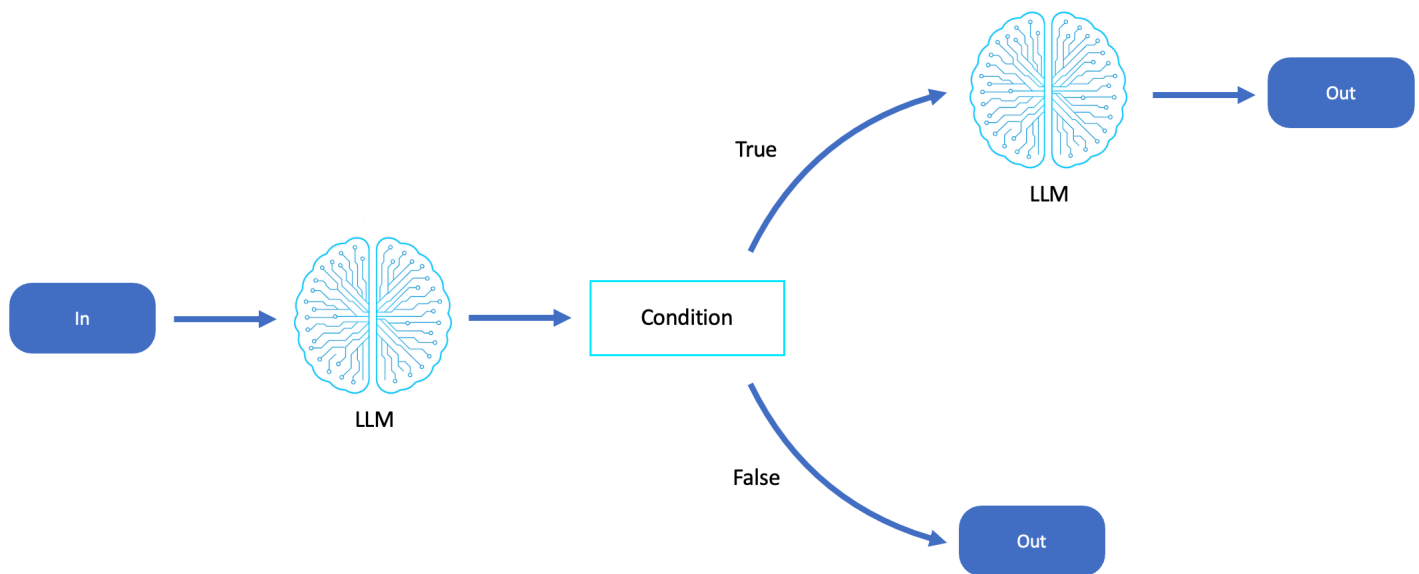
- Récupération — Fournir up-to-date ou fournir des connaissances spécifiques à un domaine à l'invite LLM par le biais de la recherche vectorielle ou de la mémoire sémantique, par exemple, par le biais de la génération augmentée par extraction (RAG)
- Utilisation de l'outil — Permettre au LLM d'invoquer APIs ou d'appeler des fonctions pour récupérer ou agir sur des informations
- Mémoire — Incorporation d'un état persistant ou d'un état basé sur une session dans la boucle de raisonnement, soit à l'aide de bases de données structurées, soit de résumés contextuels

Ces augmentations sont composées de flux de travail qui définissent la manière dont le LLM est utilisé au fil du temps et entre les tâches, le transformant d'un moteur apatride en un agent de raisonnement dynamique.



Flux de travail pour un chaînage rapide

Le chaînage rapide décompose les tâches complexes en une séquence d'étapes, chaque étape étant un appel LLM discret qui traite ou s'appuie sur le résultat de la précédente.



Le flux de travail de chaînage rapide convient aux scénarios dans lesquels les tâches peuvent être logiquement divisées en étapes de raisonnement séquentielles, et dans lesquels les résultats intermédiaires indiquent l'étape suivante. Il excelle dans les flux de travail qui nécessitent une réflexion structurée, une transformation progressive ou une analyse en couches, tels que la révision de documents, la génération de code, l'extraction de connaissances et le raffinement du contenu.

Description

- La complexité de la tâche dépasse la fenêtre contextuelle ou la profondeur de raisonnement d'un seul appel LLM.
- Les résultats d'une étape (par exemple, analyse, synthèse ou planification) deviennent des entrées pour une décision de suivi ou une phase de génération.
- Vous avez besoin de transparence et de contrôle à chaque étape du raisonnement (par exemple, résultats intermédiaires vérifiables).
- Vous souhaitez intégrer une logique de validation, de filtrage ou d'enrichissement externe entre les étapes.
- Il est idéal pour les agents opérant dans des boucles de raisonnement de type pipeline, tels que les agents de recherche, les assistants éditoriaux, les systèmes de planification et les copilotes en plusieurs étapes.

Fonctionnalités

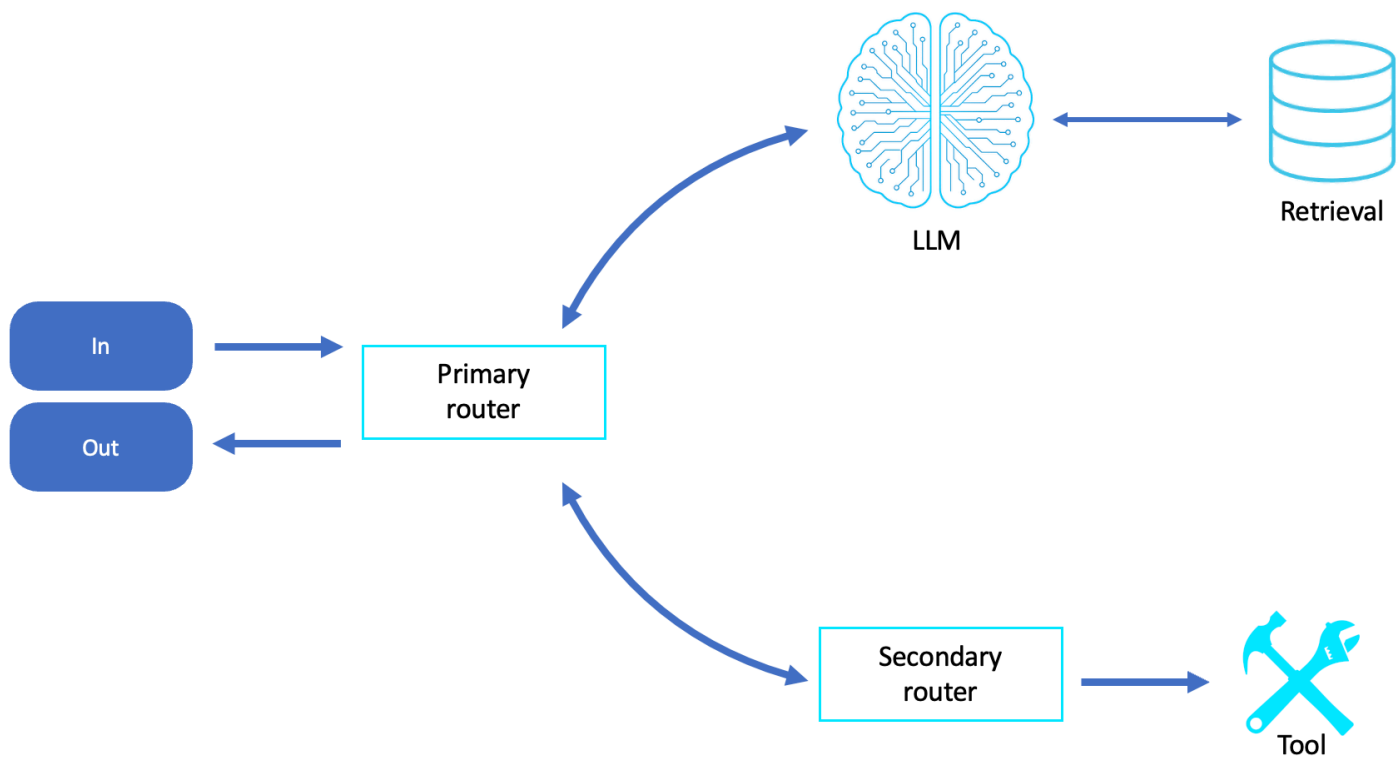
- Chaînes linéaires ou ramifiées d'appels LLM
- Résultats intermédiaires transmis sous forme de saisie structurée ou intégrés dans des instructions de suivi
- Peut être orchestré avec ou avec AWS Step Functions des coureurs AWS Lambda spécifiques à un agent

Cas d'utilisation courants

- Tâches de raisonnement en plusieurs étapes (par exemple, « réécriture d'une critique résumée »)
- Assistants de recherche synthétisant des résultats en couches (par exemple, « rechercher, extraire des faits, répondre à une question »)
- Pipelines de génération de code (« générer un plan, écrire du code, tester le code, expliquer la sortie »)

Flux de travail pour le routage

Dans le schéma de routage, un classificateur ou un agent de routeur utilise un LLM pour interpréter l'intention ou la catégorie d'une requête, puis achemine l'entrée vers une tâche ou un agent spécialisé en aval.



Le flux de travail de routage est utilisé dans les scénarios où un agent doit rapidement classer l'intention d'entrée, le type de tâche ou le domaine, puis déléguer la demande à un sous-agent, un outil ou un flux de travail spécialisé. Il est particulièrement utile pour les agents de capacité, tels que ceux qui font office d'assistants généraux, de portes d'entrée des fonctions de l'entreprise ou d'interfaces d'IA orientées utilisateur couvrant plusieurs domaines.

Le routage est particulièrement efficace lorsque :

- Triage des demandes selon diverses tâches (par exemple, recherche, synthèse, réservation, calculs).
- Les entrées doivent être prétraitées ou normalisées avant d'entrer dans des flux de travail plus spécialisés.
- Les différents types de saisie (par exemple, images par rapport au texte, requêtes structurées par rapport aux requêtes non structurées) nécessitent un traitement personnalisé.
- Un agent agit comme un central conversationnel, déléguant des tâches à des agents spécialisés ou à des microservices.
- Ce flux de travail est courant dans les copilotes spécifiques à un domaine, les robots de support client, les routeurs de services d'entreprise et les agents multimodaux, où le dispatching intelligent détermine à la fois la qualité et l'efficacité du comportement des agents.

Fonctionnalités

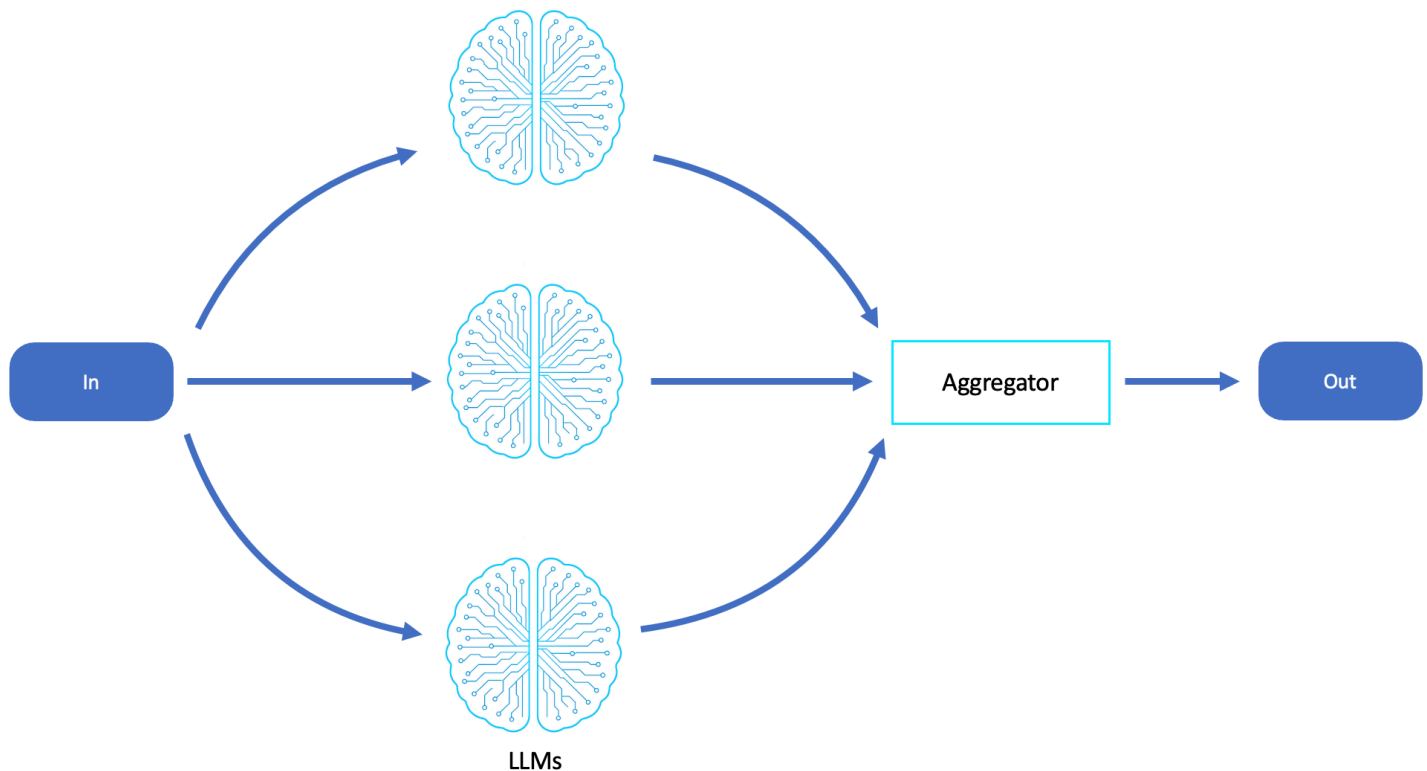
- Un LLM de premier passage agit en tant que répartiteur
- Les itinéraires peuvent invoquer des flux de travail distincts ou même d'autres modèles d'agents
- Prend en charge l'extension modulaire des capacités

Cas d'utilisation courants

- Assistants multidomaines (« s'agit-il d'une question juridique, médicale ou financière ? »)
- Arbres de décision améliorés grâce au raisonnement LLM
- Sélection d'outils dynamique (par exemple, recherche ou génération de code)

Workflow pour la parallélisation

Ce flux de travail consiste à décomposer une tâche en sous-tâches indépendantes qui peuvent être gérées simultanément par plusieurs appels ou agents LLM. Les résultats sont ensuite agrégés par programmation et synthétisés en un résultat.



Le flux de travail de parallélisation est utilisé lorsqu'une tâche peut être divisée en sous-tâches indépendantes et non séquentielles pouvant être traitées simultanément, ce qui améliore considérablement l'efficacité, le débit et l'évolutivité. Il est particulièrement puissant dans les espaces problématiques riches en données, orientés par lots ou à perspectives multiples où l'agent doit analyser ou générer du contenu à partir de plusieurs entrées.

La parallélisation est particulièrement efficace lorsque :

- Les sous-tâches ne dépendent pas des résultats intermédiaires des unes et des autres, ce qui leur permet de s'exécuter en parallèle sans coordination.
- Une tâche implique de répéter le même processus de raisonnement pour de nombreux éléments (par exemple, résumer plusieurs documents ou évaluer une liste d'options).
- De multiples hypothèses ou perspectives sont explorées en parallèle pour promouvoir la diversité, la créativité ou la robustesse.
- Vous devez réduire la latence pour les demandes à volume élevé ou à haute fréquence grâce à l'exécution simultanée du LLM.
- Ce flux de travail est couramment utilisé dans les agents de traitement de documents, les moteurs d'enquête ou de comparaison, les synthétiseurs par lots, les brainstormers multi-agents et les tâches de classification ou d'étiquetage évolutives, en particulier lorsque le raisonnement rapide et parallèle constitue un avantage en termes de performances.

Fonctionnalités

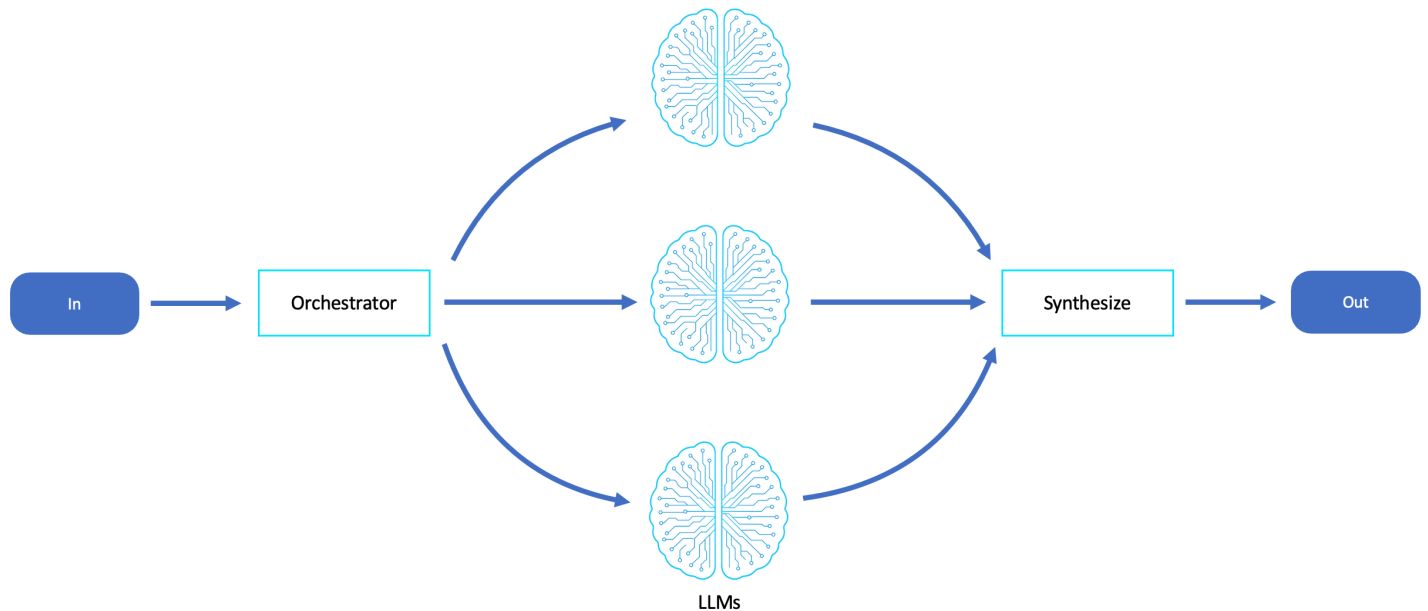
- Exécution parallèle de tâches LLM (en utilisant AWS Lambda AWS Fargate, ou un état de AWS Step Functions carte)
- Nécessite l'alignement, la validation ou la déduplication des résultats au stade de la synthèse
- Convient parfaitement aux boucles d'agents apatrides

Cas d'utilisation courants

- Analyse de plusieurs documents ou perspectives en parallèle
- Génération de brouillons, de résumés ou de plans variés
- Accélérer le débit entre les tâches par lots

Flux de travail pour l'orchestration

Un agent d'orchestration central utilise un LLM pour planifier, décomposer et déléguer des sous-tâches à des agents ou modèles spécialisés, chacun ayant un rôle ou une expertise dans un domaine spécifique. Cela reflète les structures de l'équipe humaine et soutient les comportements émergents de plusieurs agents.



Le flux de travail d'orchestration est idéal pour les scénarios complexes, hiérarchiques ou multidisciplinaires, nécessitant une décomposition structurée et une exécution spécialisée. Il est particulièrement adapté aux tâches qui nécessitent une division du travail, où les différents sous-composants d'une tâche sont mieux gérés par des agents dotés de capacités, de connaissances ou d'ensembles d'outils distincts.

Ce flux de travail est particulièrement efficace lorsque :

- Les tâches peuvent être divisées en sous-tâches dont la portée, le type ou le raisonnement varient (par exemple, planifier, rechercher, implémenter et tester).
- Un LLM ou un méta-agent doit coordonner les autres agents, suivre les progrès et synthétiser les résultats.
- Vous souhaitez modulariser les responsabilités des agents, en permettant l'évolutivité, la réutilisation et un réglage spécialisé.
- Le système nécessite un comportement basé sur les rôles, imitant la façon dont les équipes humaines (par exemple, les chefs de projet, les développeurs et les réviseurs) opèrent en collaboration.

L'orchestration est idéale pour les agents de planification multitours, les copilotes de développement logiciel, les agents de processus d'entreprise et les exécuteurs de projets autonomes. Il est particulièrement utile lors de la mise en œuvre de systèmes multi-agents qui nécessitent une répartition centralisée des tâches mais une logique d'exécution distribuée, ce qui permet une extensibilité et un comportement plus explicable entre les couches d'agents.

Fonctionnalités

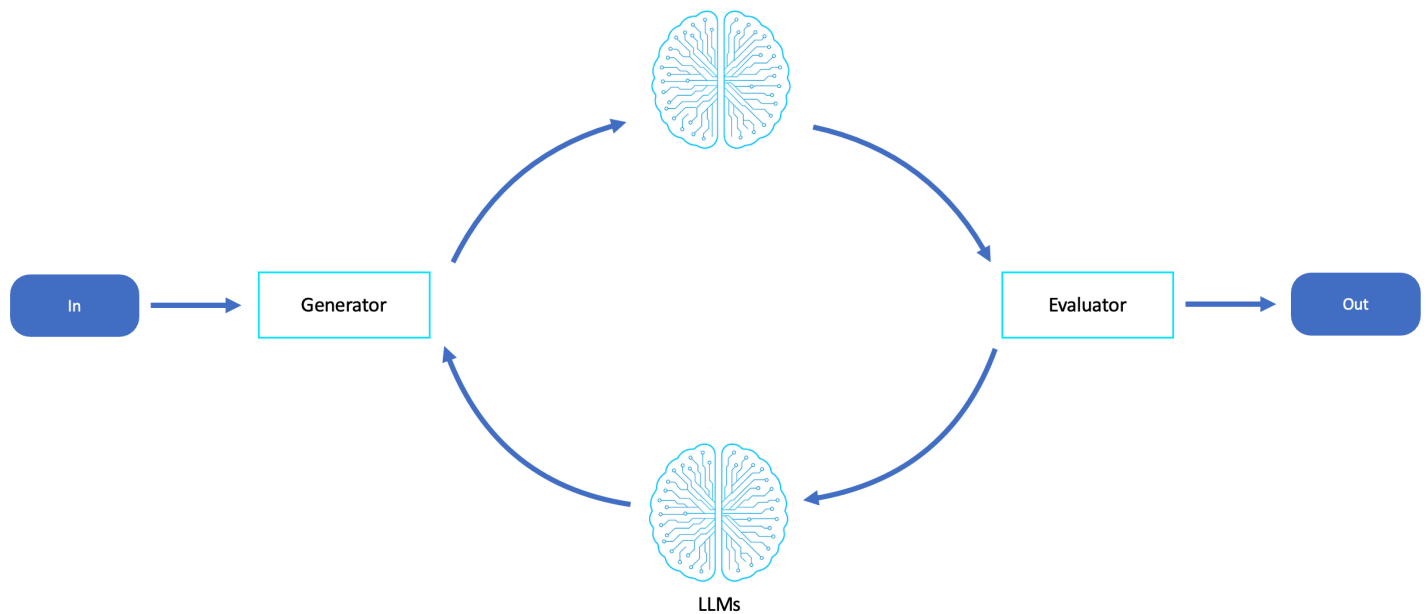
- Orchestrator exécute un méta-raisonnement sur les objectifs
- Les agents de travail peuvent inclure l'accès aux outils, la mémoire ou des instructions spécifiques au domaine
- Peut être hiérarchique (c'est-à-dire délégation de tâches à plusieurs niveaux)

Cas d'utilisation courants

- Chefs de projet, chercheurs coordinateurs, rédacteurs et agents d'assurance qualité
- Copilotes de codage qui combinent la planification, l'exécution et les tests
- Agents qui supervisent les chaînes d'outils ou les modèles d'accès aux API

Flux de travail pour les évaluateurs et boucles d'affinage par réflexion

Ce flux de travail fournit une boucle de rétroaction dans laquelle un LLM génère un résultat et un autre évalue ou critique le résultat. Cela favorise l'autoréflexion, l'optimisation et les améliorations itératives.



Le flux de travail de l'évaluateur est idéal pour les scénarios dans lesquels la qualité, la précision et l'alignement des résultats sont importants et dans lesquels la génération en un seul passage est peu fiable ou insuffisante. Ce flux de travail excelle lorsque les agents doivent s'autocritiquer, itérer et affiner leurs résultats, soit pour répondre à des normes d'exactitude plus élevées, soit pour explorer des alternatives améliorées en fonction des commentaires.

Ce flux de travail est particulièrement efficace lorsque :

- Le résultat implique des mesures de qualité subjectives (par exemple, le style, le ton et la lisibilité) ou des critères objectifs (par exemple, exactitude, sécurité et performance).
- L'agent doit raisonner en faisant des compromis, évaluer les contraintes ou optimiser pour atteindre un objectif.
- Vous avez besoin d'une redondance et d'une assurance qualité intégrées, en particulier dans les domaines réglementés, destinés aux clients ou créatifs.
- Human-in-the-loop la révision coûte cher ou n'est pas disponible, et une validation autonome est souhaitée.

Ce flux de travail est utilisé pour la génération de contenu, la synthèse et la révision du code, l'application des politiques, le contrôle de l'alignement, le réglage des instructions et le post-traitement RAG. Il est également utile pour les agents qui s'améliorent eux-mêmes, où le feedback continu contribue à façonner de meilleures réponses au fil du temps afin de créer des boucles de décision fiables et autonomes.

Cas d'utilisation courants

- Comparaison des agents de l'équipe rouge aux agents de l'équipe bleue
- Agents qui génèrent, évaluent et révisent le code ou les plans
- Assurance qualité, détection des hallucinations et application du style

Fonctionnalités

- Supporte la génération et l'évaluation découplées à l'aide de différents modèles (par exemple, Claude pour la génération et Mistral pour l'évaluation)
- Les commentaires sont structurés et utilisés pour demander des résultats révisés
- Supporte plusieurs itérations ou seuils de convergence

Conclusion

LLMs fournissent le cœur cognitif des agents logiciels modernes, mais l'invocation de modèles bruts ne suffit pas pour obtenir une intelligence ciblée, robuste et contrôlable. Pour passer de la génération de résultats à un raisonnement structuré et à un comportement aligné sur les objectifs, LLMs il doit être intégré à des modèles de flux de travail intentionnels qui définissent la manière dont les modèles traitent les entrées, gèrent les contextes et coordonnent les actions.

Les flux de travail LLM introduisent les bases nécessaires à la création du module cognitif d'un agent :

- Le chaînage rapide décompose le raisonnement complexe en étapes modulaires et contrôlables.
- Le routage permet une classification intelligente des tâches et une délégation ciblée.
- La parallélisation accélère le débit et favorise la diversité des raisonnements.
- L'orchestration des agents structure la collaboration multi-agents par le biais de la décomposition des tâches et de l'exécution basée sur les rôles.
- L'évaluateur (boucle reflect-refine) permet l'auto-amélioration, le contrôle de la qualité et le contrôle de l'alignement.

Chaque flux de travail représente un modèle composable qui peut être adapté aux besoins de l'agent, à la complexité de la tâche et aux attentes de l'utilisateur. Ces flux de travail ne s'excluent pas mutuellement. Ce sont des éléments de base qui sont souvent combinés dans des architectures

hybrides qui prennent en charge le raisonnement dynamique, la coordination multi-agents et une fiabilité de niveau entreprise.

Au fur et à mesure que vous passerez au chapitre suivant sur les modèles de flux de travail agentiques, ces flux de travail LLM réapparaîtront sous forme de structures intégrées dans des systèmes plus importants, prenant en charge la délégation d'objectifs, l'orchestration des outils, les boucles de décision et l'autonomie du cycle de vie. La maîtrise de ces flux de travail LLM est essentielle pour concevoir des agents logiciels qui ne se contentent pas de prédire le texte, mais aussi de raisonner, de s'adapter et d'agir de manière ciblée.

Modèles de flux de travail agentic

Les modèles de flux de travail agentic intègrent des agents logiciels modulaires à des flux de travail structurés basés sur des modèles de langage large (LLM), permettant ainsi un raisonnement et une action autonomes. Tout en s'inspirant des architectures traditionnelles sans serveur et pilotées par les événements, ces modèles font passer la logique de base du code statique à des agents augmentés par le LLM, offrant ainsi une adaptabilité et une prise de décision contextuelle améliorées. Cette évolution transforme les architectures cloud classiques des systèmes déterministes en systèmes capables d'interprétation dynamique et d'augmentation intelligente, tout en maintenant les principes fondamentaux d'évolutivité et de réactivité.

Dans cette section

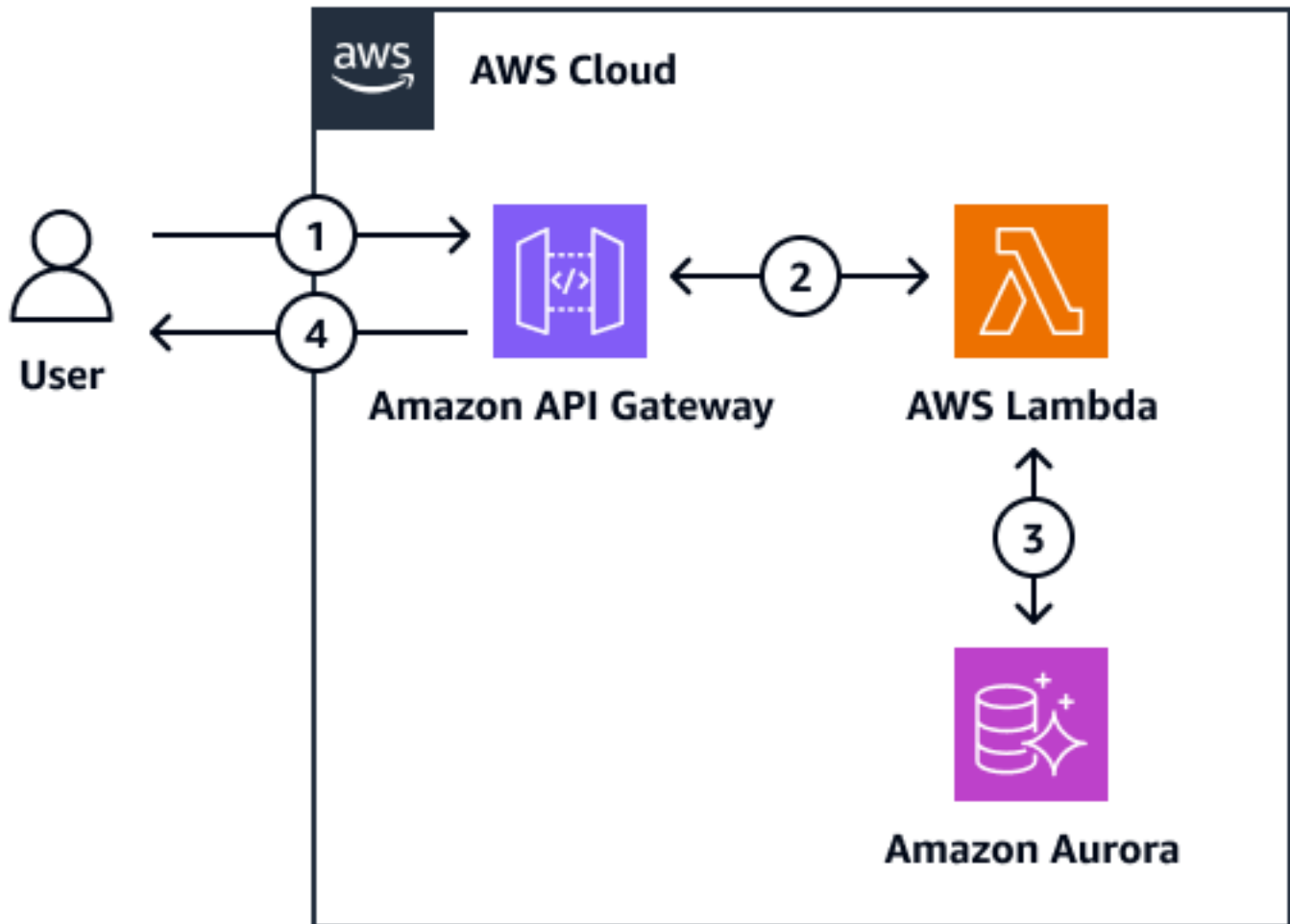
- [Des systèmes pilotés par les événements aux systèmes augmentés par la cognition](#)
- [Motifs de saga à enchaînement rapide](#)
- [Acheminement des modèles d'expédition dynamiques](#)
- [Modèles de parallélisation et de diffusion](#)
- [Modèles d'orchestration de Saga](#)
- [L'évaluateur reflète et affine les modèles de boucles](#)
- [Conception de flux de travail agentic sur AWS](#)
- [Conclusion](#)

Des systèmes pilotés par les événements aux systèmes augmentés par la cognition

Les architectures cloud modernes, en particulier celles basées sur des principes sans serveur et pilotés par les événements, se sont traditionnellement appuyées sur des modèles tels que le routage, le fan-out et l'enrichissement pour créer des systèmes réactifs et évolutifs. Les systèmes d'IA agentic s'appuient sur ces fondements tout en les recadrant autour du raisonnement augmenté par le LLM et de la flexibilité cognitive. Cette approche permet des capacités de résolution de problèmes et d'automatisation plus sophistiquées, susceptibles de révolutionner la manière dont les tâches complexes sont gérées dans les environnements cloud.

Architecture basée sur les événements

Le schéma suivant montre un système distribué typique :

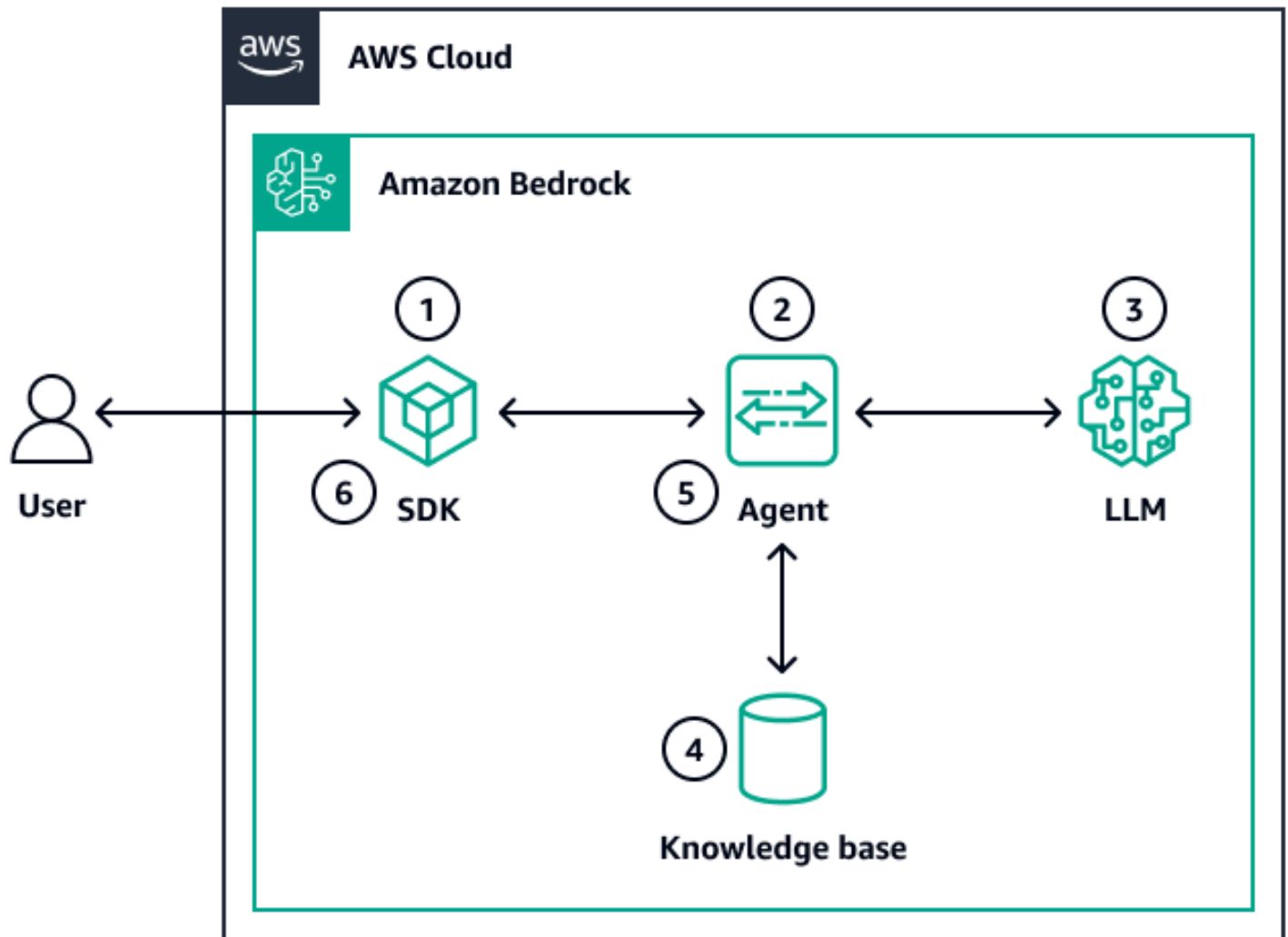


1. Un utilisateur envoie une demande à Amazon API Gateway.
2. Amazon API Gateway achemine la demande vers une AWS Lambda fonction.
3. AWS Lambda enrichit les données en interrogeant une base de données Amazon Aurora
4. Amazon API Gateway renvoie la charge utile enrichie à l'appelant.

Cette structure est à la fois fiable et évolutive, mais elle est fondamentalement statique. Les règles métier et les chemins logiques doivent être explicitement codés, et l'adaptation à des contextes changeants ou à des informations incomplètes est limitée.

Flux de travail augmentés par la cognition

Les architectures agentic ajoutent une augmentation cognitive à un système piloté par les événements. Le schéma suivant montre un équivalent agentique :



1. Un utilisateur soumet une requête via un SDK ou un appel d'API.
2. Un agent Amazon Bedrock reçoit la requête.
3. L'agent interprète la requête en invoquant un LLM
4. L'agent effectue un enrichissement sémantique en effectuant des recherches dans la base de connaissances Amazon Bedrock ou dans d'autres sources de données externes.
5. Le LLM synthétise une réponse riche en contexte et alignée sur les objectifs.
6. Le système renvoie une réponse synthétisée à l'utilisateur.

Dans ce flux, le LLM utilise la logique, comprend l'intention, récupère et combine le contexte pertinent, puis décide de la meilleure façon de répondre. Ce modèle reflète le modèle d'enrichissement traditionnel, dans lequel les messages sont complétés par des données externes avant d'être acheminés ultérieurement. Dans les systèmes agentiques, toutefois, cet enrichissement n'est pas une recherche statique. L'enrichissement est plutôt dynamique, guidé sémantiquement et motivé par un objectif.

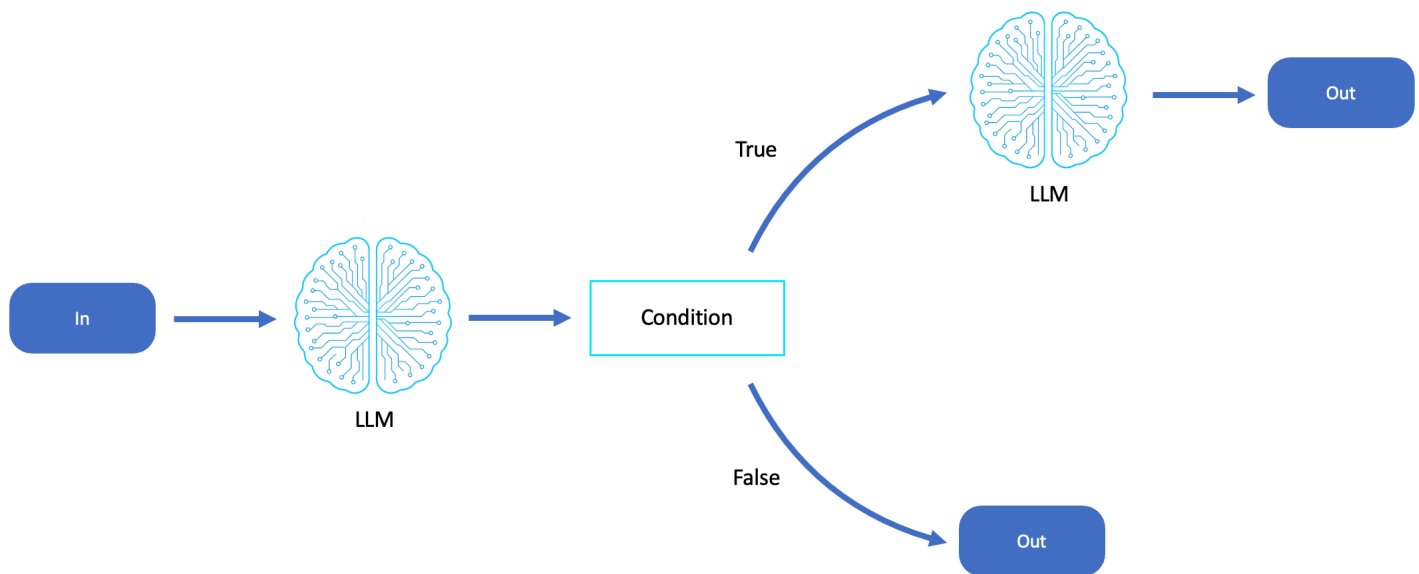
Informations de base

Chaque flux de travail LLM peut être mappé à un modèle de flux de travail agentic, qui reflète et fait évoluer les styles d'architecture traditionnels axés sur les événements. L'un des éléments de base des flux de travail agentiques est la capacité d'enrichir le contexte d'un LLM avec des données, des outils et de la mémoire. Cela crée une boucle de raisonnement informée, adaptative et alignée sur l'intention de l'utilisateur. Alors que les systèmes traditionnels enrichissent les messages avec des données de recherche, les systèmes agentic permettent aux logiciels d'agir moins comme des scripts que comme des collaborateurs intelligents.

Motifs de saga à enchaînement rapide

En réinventant le chaînage rapide LLM sous la forme d'une saga axée sur les événements, nous ouvrons la voie à un nouveau modèle opérationnel : les flux de travail deviennent distribués, récupérables et coordonnés sémantiquement entre les agents autonomes. Chaque étape de réponse rapide est redéfinie comme une tâche atomique, émise en tant qu'événement, consommée par un agent dédié et enrichie de métadonnées contextuelles.

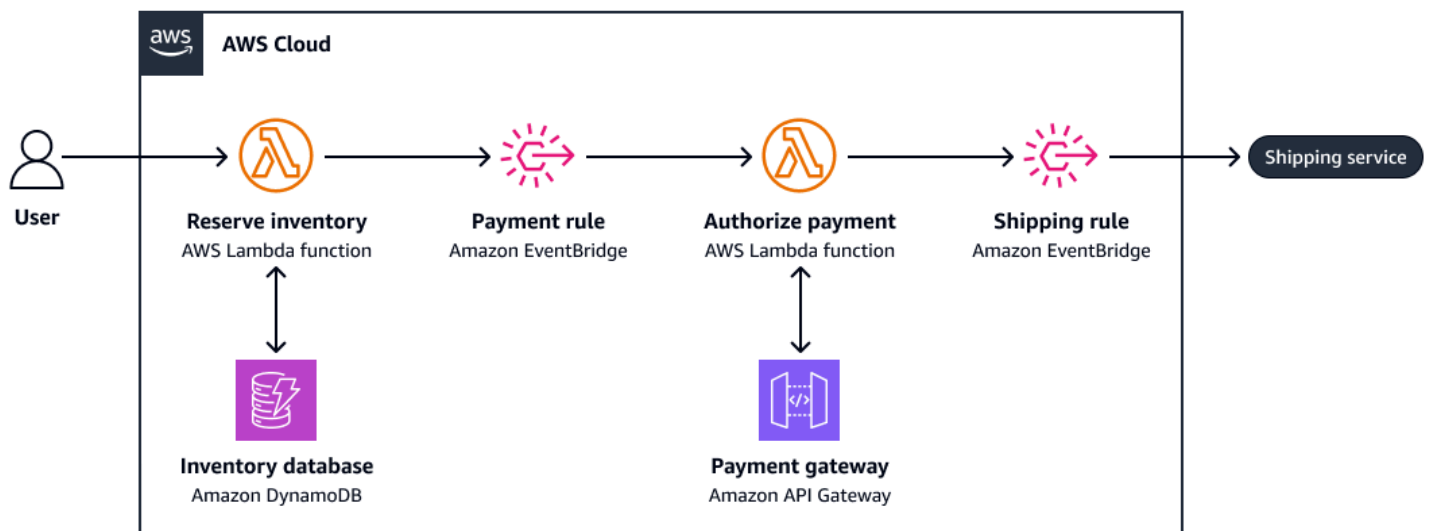
Le schéma suivant est un exemple de chaînage d'invite LLM :



Chorégraphie de saga

Le modèle de chorégraphie de la saga est une approche de mise en œuvre dans les systèmes distribués qui n'a pas de coordinateur central. Au lieu de cela, chaque service ou composant publie des événements qui déclenchent l'action suivante du flux de travail. Ce modèle est largement utilisé dans les systèmes distribués pour gérer les transactions entre plusieurs services. Dans une saga, le système exécute une série de transactions locales coordonnées. En cas d'échec, le système déclenche des actions de compensation pour maintenir la cohérence.

Le schéma suivant est un exemple de chorégraphie de saga :



1. Inventaire de réserve

2. Autoriser le paiement
3. Créer un bon de commande d'expédition

Si l'étape 3 échoue, le système invoque des actions compensatoires (par exemple, annuler un paiement ou libérer un inventaire).

Ce modèle est particulièrement utile dans les architectures axées sur les événements où les services sont faiblement couplés et où les états doivent être résolus de manière cohérente au fil du temps, même en cas de défaillance partielle.

Schéma de chaînage rapide

Le chaînage rapide ressemble au schéma de la saga, à la fois en termes de structure et d'objectif. Il exécute une série d'étapes de raisonnement qui se développent de manière séquentielle tout en préservant le contexte et en autorisant les annulations et les révisions.

Chorégraphie d'agent

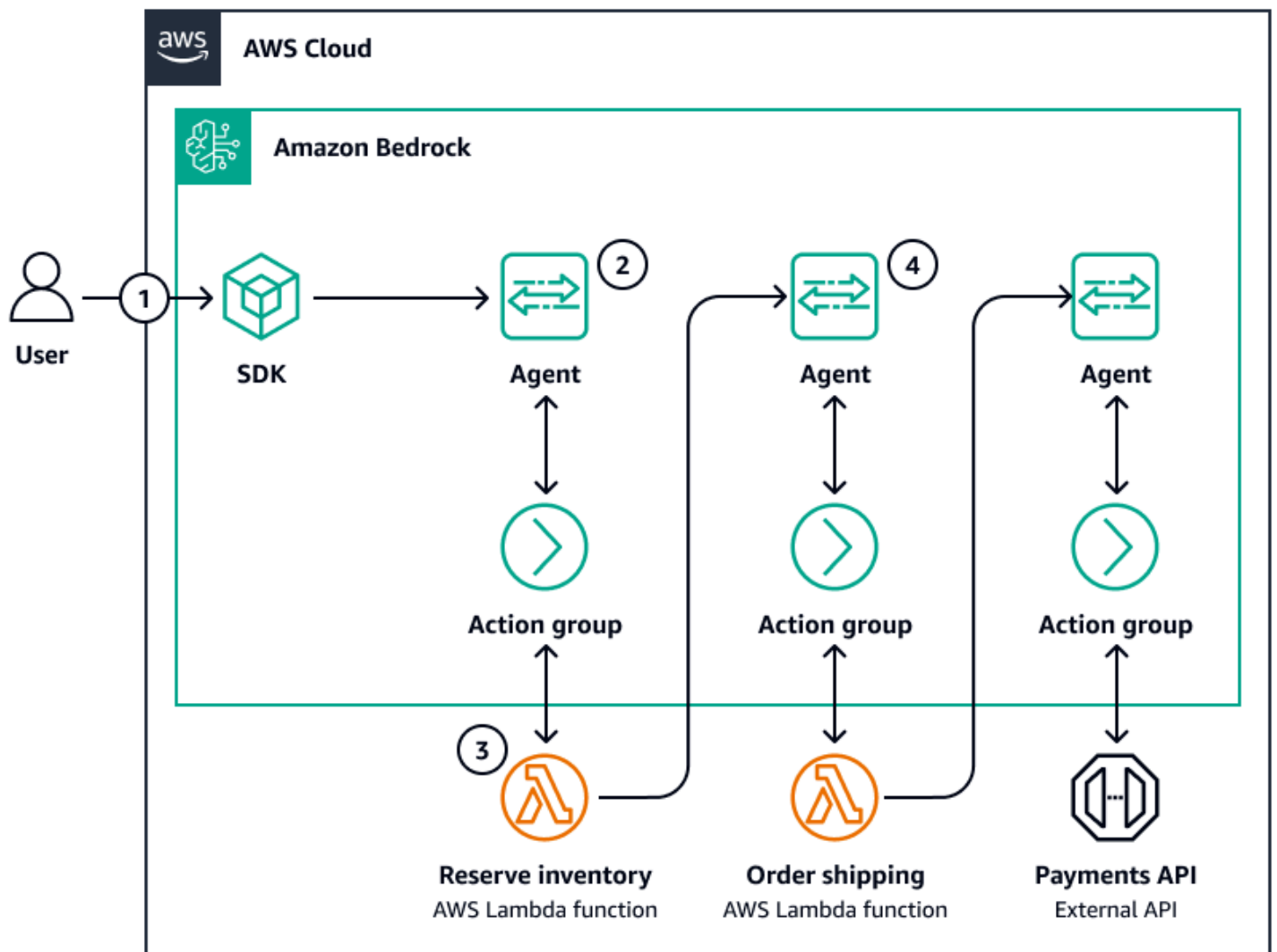
1. LLM interprète une requête utilisateur complexe et génère une hypothèse
2. Le LLM élabore un plan pour résoudre la tâche
3. LLM exécute une sous-tâche (par exemple, en utilisant un appel d'outil ou en récupérant des connaissances)
4. Le LLM affine le résultat ou revoit une étape précédente s'il juge le résultat insatisfaisant

Si un résultat intermédiaire est erroné, le système peut effectuer l'une des opérations suivantes :

- Réessayez les étapes en utilisant une approche différente
- Revenir à une demande précédente et replanifier
- Utiliser une boucle d'évaluateur (par exemple, à partir du modèle évaluateur-optimiseur) pour détecter et corriger les défaillances

À l'instar du modèle de saga, le chaînage rapide permet des mécanismes de progression partielle et de retour en arrière. Cela se fait par le biais d'un raffinement itératif et d'une correction dirigée par le LLM plutôt que par la compensation des transactions de base de données.

Le schéma suivant est un exemple de chorégraphie d'agent :



1. Un utilisateur soumet une requête par le biais d'un SDK.
2. Un agent Amazon Bedrock orchestre le raisonnement de la manière suivante :
 - Interprétation (LLM)
 - Planification (LLM)
 - Exécution par le biais d'un outil ou d'une base de connaissances
 - Construction de la réponse
3. Si un outil échoue ou ne renvoie pas suffisamment de données, l'agent peut replanifier ou reformuler la tâche de manière dynamique.
4. La mémoire (par exemple, un magasin de vecteurs à court terme) peut conserver son état d'une étape à l'autre

Plats à emporter

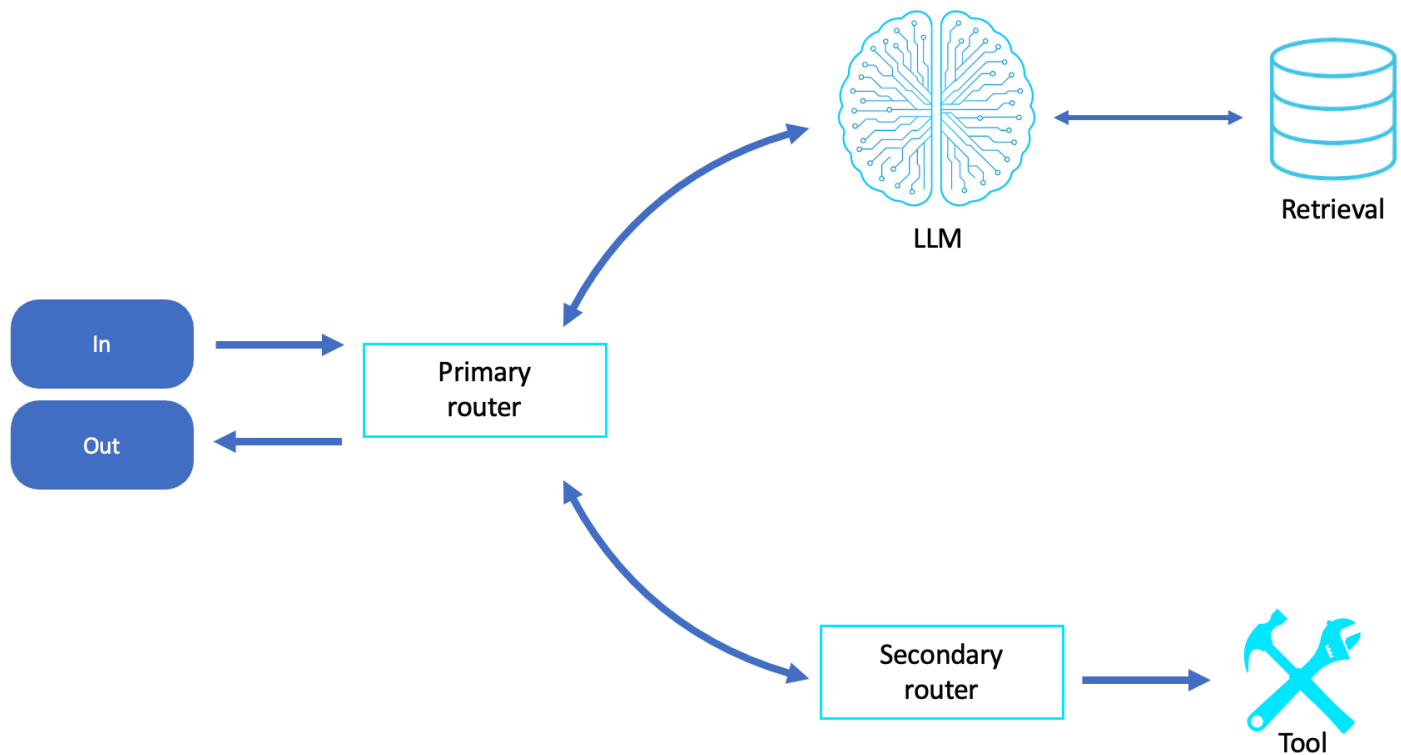
Alors que le modèle en saga gère les appels de service distribués selon une logique de compensation, le chaînage rapide gère les tâches de raisonnement grâce à un séquençage réflexif et à une replanification adaptative. Les deux systèmes permettent de réaliser des progrès progressifs, de décentraliser les points de décision et de remédier aux défaillances, et tout cela grâce à un raisonnement éclairé plutôt qu'à un retour en arrière rigide.

Le chaînage rapide introduit le raisonnement transactionnel, qui est l'équivalent cognitif des sagas. C'est-à-dire que chaque « pensée » est réévaluée, révisée ou abandonnée dans le cadre d'un dialogue plus large axé sur les objectifs.

Acheminement des modèles d'expédition dynamiques

Dans les systèmes agentiques modernes, où les tâches vont de l'analyse de documents à la génération autonome de logiciels, la capacité d'acheminer dynamiquement les demandes vers le modèle ou l'agent de grande langue (LLM) le plus performant devient essentielle. La logique de routage statique, souvent intégrée dans des scripts d'orchestration ou des couches d'API, ne possède pas l'adaptabilité requise pour les environnements multi-modèles et multi-capacités en temps réel. Pour y remédier, les flux de travail de routage LLM peuvent être transformés en une architecture axée sur les événements qui tire parti d'un modèle d'expédition dynamique, transformant les appels LLM en événements routés intelligemment et sensibles au contexte.

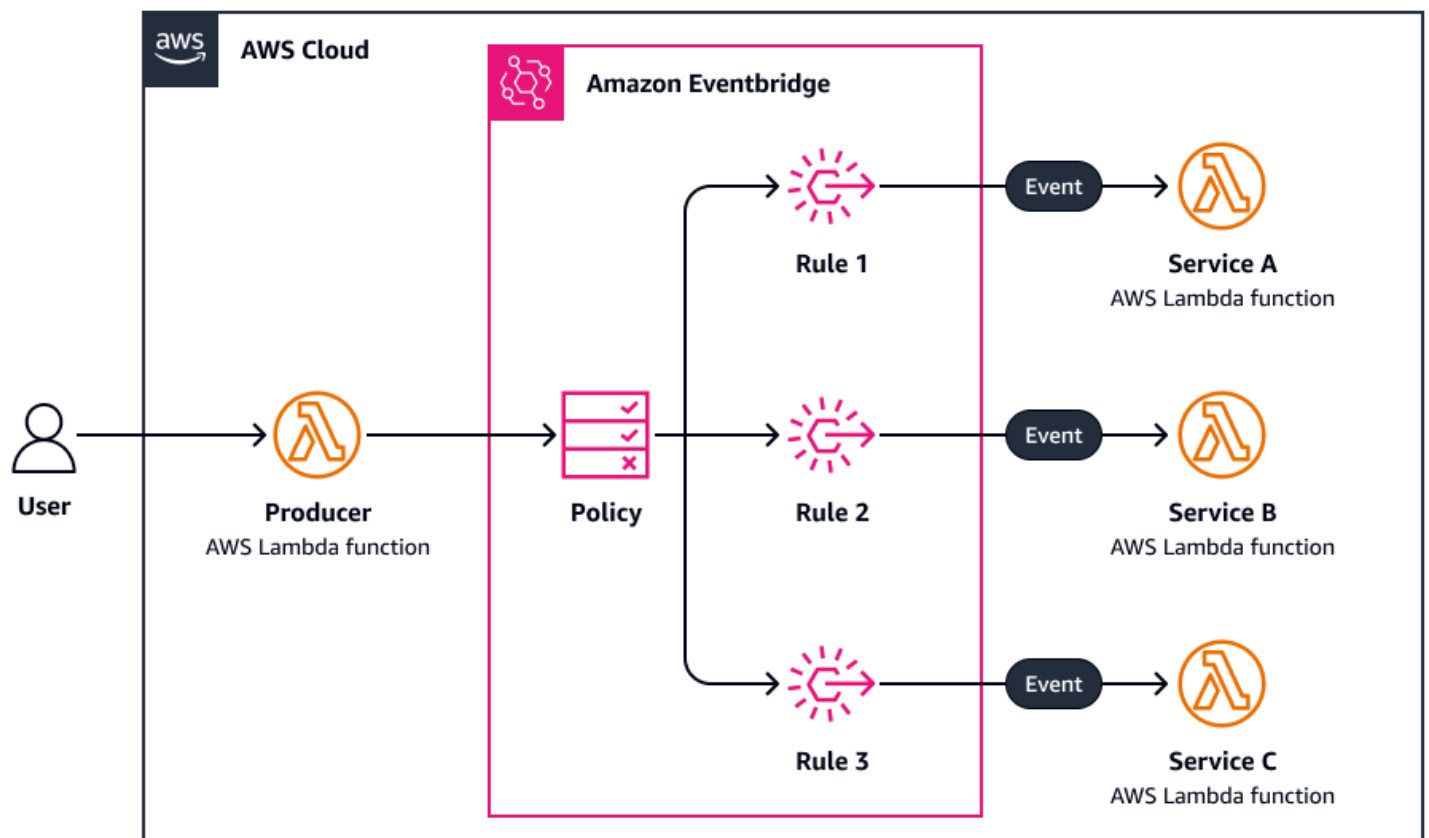
Le schéma suivant est un exemple de routage LLM :



Expédition dynamique

Dans les systèmes distribués traditionnels, le modèle de distribution dynamique sélectionne et invoque des services spécifiques au moment de l'exécution en fonction des attributs des événements entrants, tels que le type d'événement, la source et la charge utile. Ceci est généralement mis en œuvre à l'aide d'Amazon EventBridge, qui peut évaluer et acheminer les événements entrants vers des cibles appropriées (par exemple, des AWS Lambda fonctions AWS Step Functions ou des tâches Amazon Elastic Container Service).

Le schéma suivant est un exemple de répartition dynamique :



1. Une application émet un événement (par exemple, {"type" : « OrderCreated », « priority » : « high »}).
2. Amazon EventBridge évalue l'événement par rapport à ses règles de routage.
3. Sur la base des attributs d'un événement, le système effectue des envois dynamiques vers les sites suivants :
 - HighPriorityOrderProcessor(service A)
 - StandardOrderProcessor(service B)
 - UpdateOrderProcessor(service C)

Ce modèle prend en charge le couplage souple, la spécialisation basée sur le domaine et l'extensibilité du temps d'exécution. Cela permet aux systèmes de répondre intelligemment à l'évolution des exigences et de la sémantique des événements.

Routage basé sur le LLM

Dans les systèmes agentic, le routage effectue également une délégation dynamique des tâches, mais au lieu des EventBridge règles Amazon ou des filtres de métadonnées, le LLM classe et

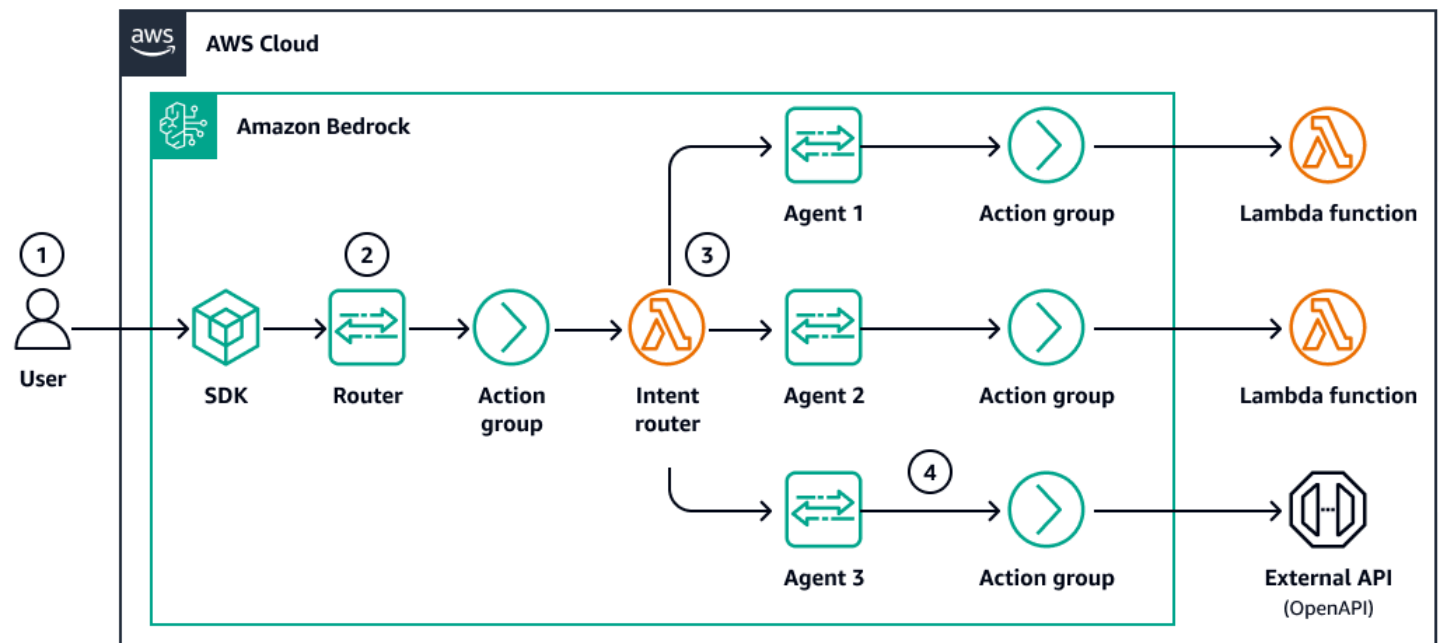
interprète l'intention de l'utilisateur en langage naturel. Le résultat est une forme de répartition flexible, sémantique et adaptative.

Routeur agent

Cette architecture permet une distribution riche basée sur l'intention sans schémas ni types d'événements prédéfinis, ce qui est idéal pour les entrées non structurées et les requêtes complexes.

1. Un utilisateur soumet la demande « Pouvez-vous m'aider à revoir les termes de mon contrat ? »
2. Le LLM interprète cela comme une tâche de documentation juridique.
3. L'agent achemine la tâche vers un ou plusieurs des sites suivants :
 - Modèle de demande de révision de contrat
 - Sous-agent de raisonnement juridique
 - outil d'analyse de documents

Le schéma suivant est un exemple d'agent-routeur :



1. Un utilisateur soumet une demande en langage naturel par le biais d'un SDK.
2. Un agent Amazon Bedrock utilise un LLM pour classer la tâche (par exemple, juridique, technique ou de planification).
3. L'agent achemine dynamiquement la tâche via un groupe d'action pour appeler l'agent requis :

- Agent spécifique au domaine
- Chaîne d'outils spécialisée
- Configuration rapide personnalisée

4. Le gestionnaire sélectionné traite la tâche et renvoie une réponse personnalisée.

Plats à emporter

Alors que l'expédition dynamique traditionnelle utilise EventBridge les règles d'Amazon pour le routage en fonction d'attributs d'événements structurés, le routage agentic permet de classer et d'LLMs acheminer sémantiquement les tâches en fonction de leur signification et de leur intention. Cela augmente la flexibilité du système en permettant ce qui suit :

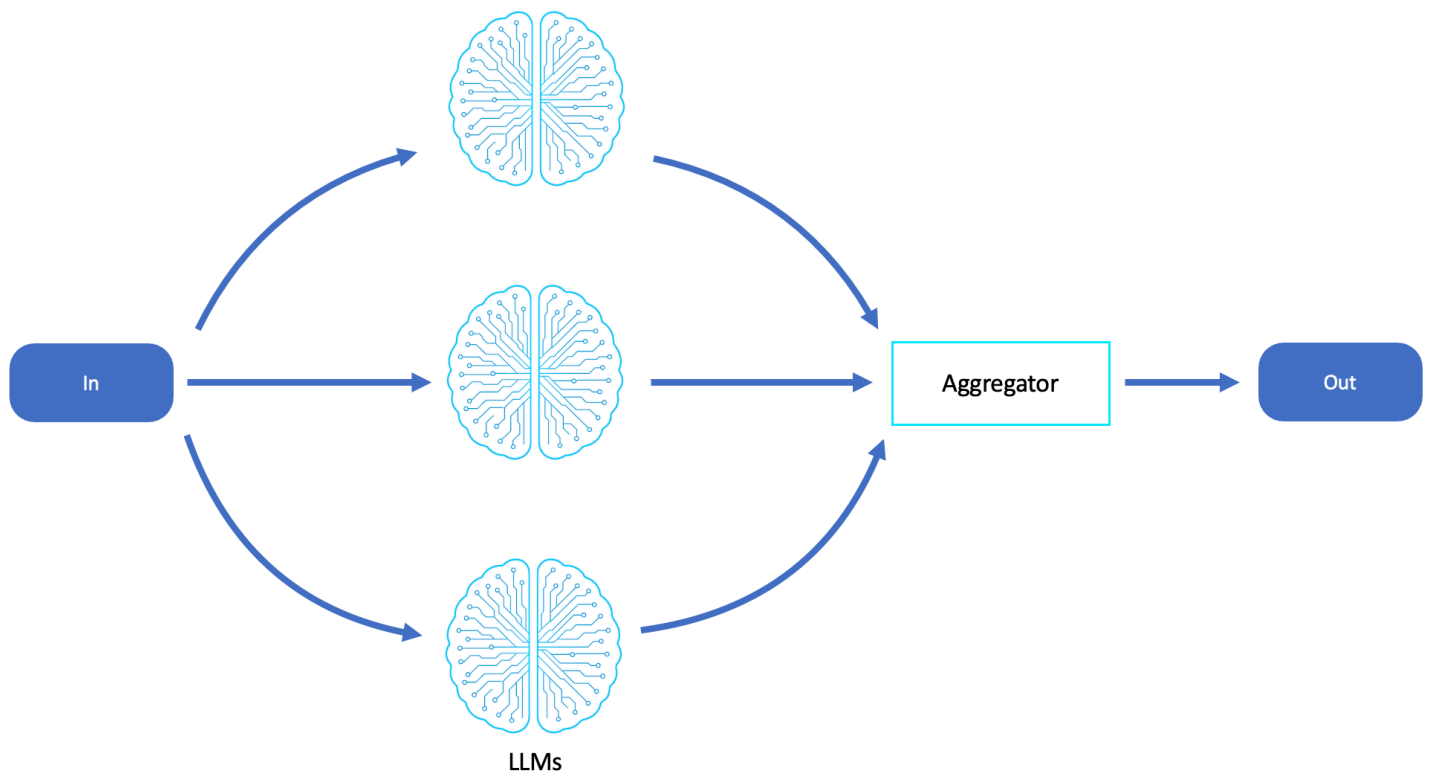
- Compréhension plus large des entrées
- Solution de repli intelligente et sélection d'outils
- Extensibilité naturelle grâce à de nouveaux rôles d'agent ou à des styles rapides

Le routage agentic remplace les règles rigides par un dispatching cognitif dynamique, qui permet aux systèmes d'évoluer en fonction du langage plutôt que du code.

Modèles de parallélisation et de diffusion

De nombreuses tâches avancées de raisonnement et de génération, telles que la synthèse de documents volumineux, l'évaluation de plusieurs solutions ou la comparaison de points de vue variés, tirent parti de l'exécution parallèle des instructions. Les flux de travail séquentiels traditionnels sont insuffisants lorsque l'évolutivité, la réactivité et la tolérance aux pannes sont requises. Pour surmonter ce problème, la parallélisation basée sur le LLM peut être réinventée à l'aide d'un modèle de collecte par dispersion piloté par des événements, dans lequel les tâches sont réparties dynamiquement vers des agents autonomes et les résultats synthétisés de manière intelligente.

Le schéma suivant est un exemple de flux de travail de parallélisation LLM :



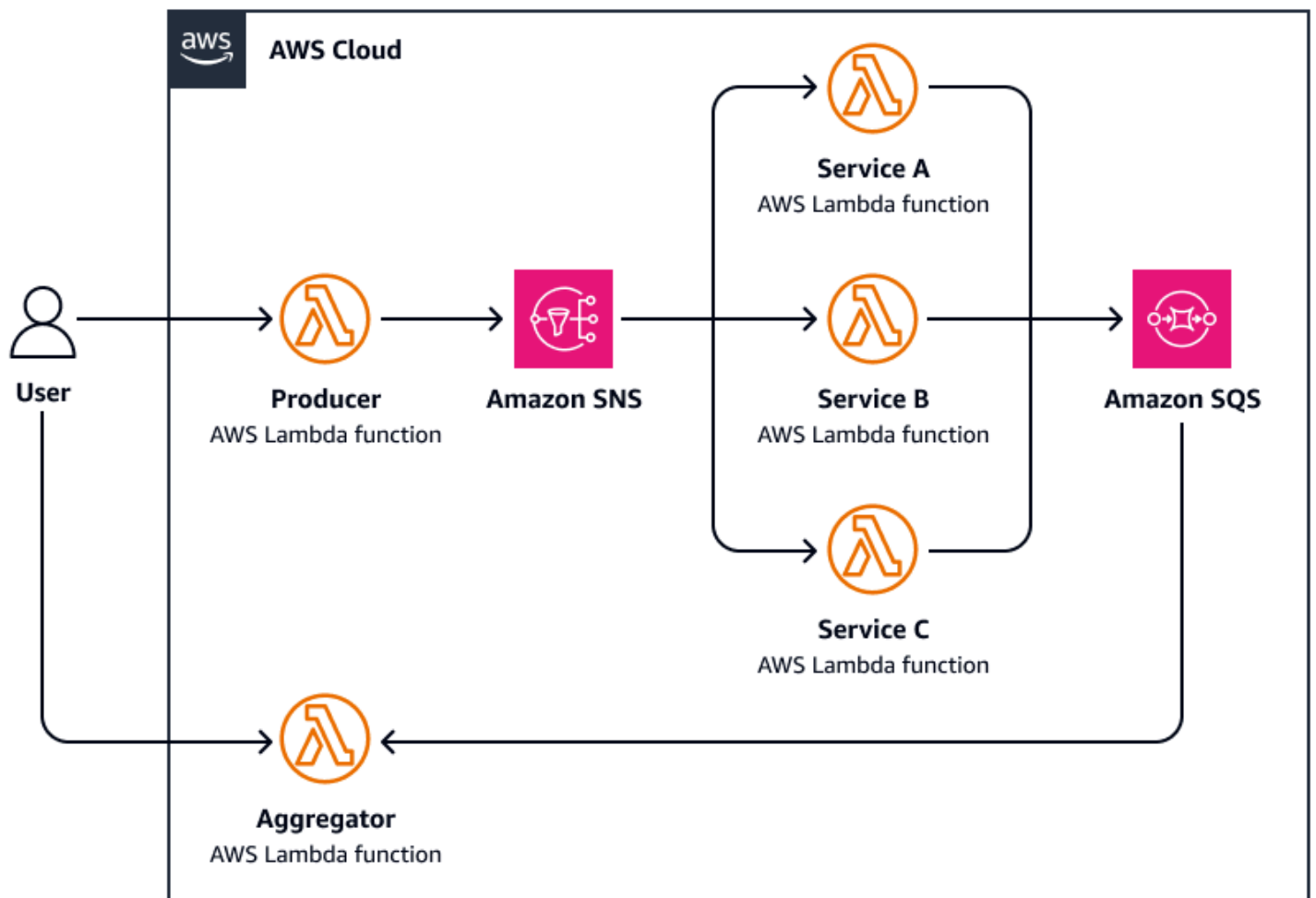
Dispersez et collectez

Dans les systèmes distribués, un modèle de collecte par dispersion envoie des tâches à plusieurs services ou unités de traitement en parallèle, attend leurs réponses, puis agrège les résultats dans une sortie consolidée. Contrairement au fan-out, le scatter-gather est coordonné car il attend des réponses et applique généralement une logique pour combiner, comparer et sélectionner les résultats.

Les implémentations courantes pour la parallélisation et le scatter-gather sont les suivantes :

- AWS Step Functions mapper un état pour l'exécution de tâches en parallèle
- AWS Lambda avec simultanée, coordination des résultats de plusieurs fonctions invoquées
- Amazon EventBridge avec des flux de travail IDs de corrélation et d'agrégation
- Modèle de contrôleur personnalisé pour gérer le fan-out et recueillir des résultats à l'aide d'Amazon Simple Storage Service (Amazon S3), d'Amazon DynamoDB ou de files d'attente

Le schéma suivant est un exemple de scatter-gather :



1. Un utilisateur envoie une demande à une fonction de coordination centrale qui répartit la tâche en publiant des messages parallèles sur une rubrique Amazon Simple Notification Service (Amazon SNS).
2. Chaque message inclut les métadonnées des tâches et est acheminé vers un travailleur AWS Lambda spécialisé.
3. Chaque collaborateur traite AWS Lambda indépendamment la sous-tâche qui lui est assignée (par exemple, interroger une API externe, traiter un document et analyser des données).
4. Les résultats sont écrits sur une couche de stockage commune, telle qu'Amazon Simple Queue Service (Amazon SQS).
5. La fonction d'agrégation attend que toutes les réponses soient terminées, puis elle effectue les opérations suivantes :
 - Rassemble et agrège les résultats (par exemple, fusionne les résumés, sélectionne les meilleures correspondances)

- Envoie une réponse finale ou déclenche un flux de travail en aval

Les cas d'utilisation courants des modèles de dispersion sont les suivants :

- Recherche fédérée
- Moteurs de comparaison de prix
- Analyse des données agrégées
- Inférence multimodèle

Parallélisation basée sur le LLM (cognition par diffusion)

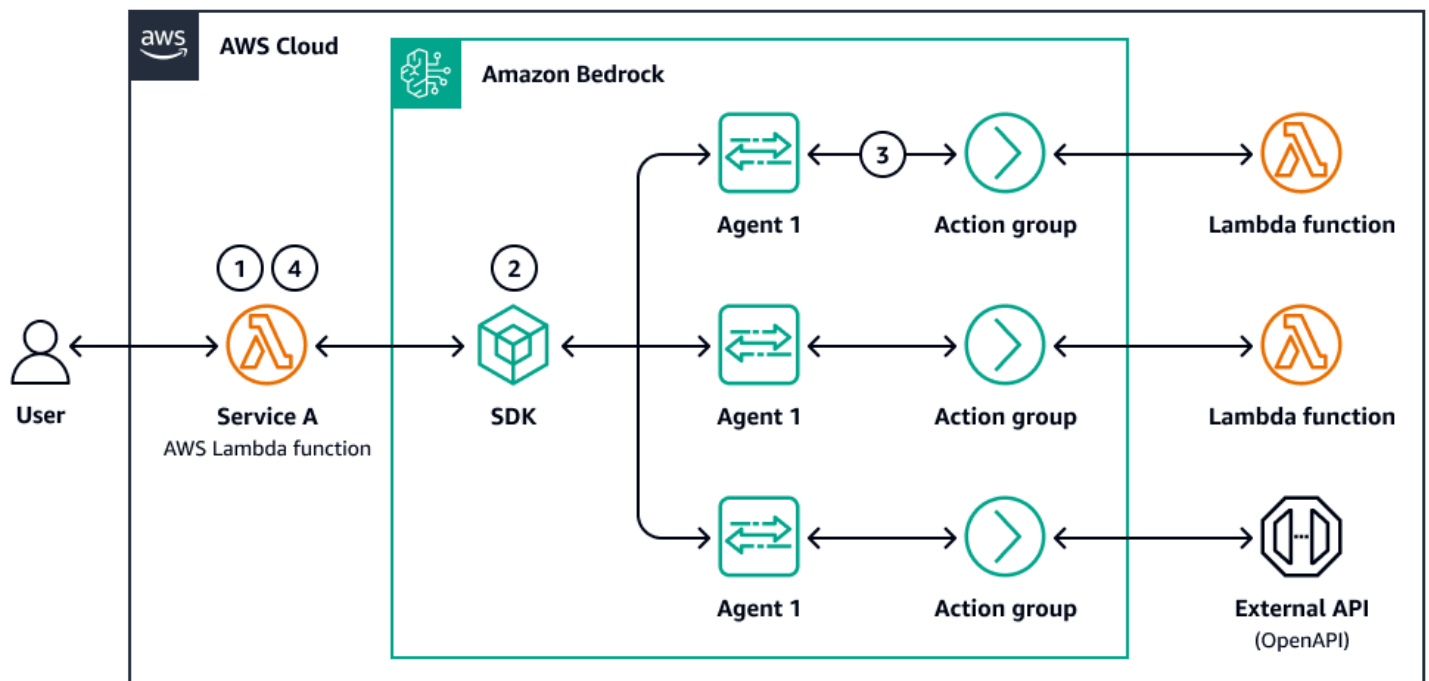
Dans les systèmes agentic, la parallélisation reflète étroitement la diffusion en répartissant les sous-tâches entre plusieurs appels ou agents LLM, chacun abordant indépendamment une partie du problème. Les résultats renvoyés sont collectés et synthétisés par un processus d'agrégation, qui est souvent un autre LLM ou agent de contrôle.

Parallélisation des agents

1. Un agent soumet une demande « Résumez les informations issues de ces 10 rapports ».
2. Il répartit les rapports en 10 tâches de synthèse LLM parallèles.
3. Lorsqu'il renvoie tous les résumés, l'agent effectue les opérations suivantes :
 - Regroupe les résumés dans un briefing unifié
 - Identifie les thèmes ou les contradictions
 - Envoie la sortie synthétisée à l'utilisateur

Ce flux de travail agentic permet un raisonnement parallèle évolutif, modulaire et adaptatif. C'est idéal pour les cas d'utilisation nécessitant un débit cognitif élevé.

Le schéma suivant est un exemple de parallélisation d'agents :



1. Un utilisateur soumet une requête ou un ensemble de documents en plusieurs parties.
2. Un contrôleur AWS Lambda ou une fonction d'étape distribue les sous-tâches. Chaque tâche appelle un appel ou un sous-agent Amazon Bedrock LLM avec sa propre invite.
3. Lorsque les appels et les sous-tâches sont terminés, les résultats sont stockés (par exemple, dans Amazon S3 ou dans une mémoire), et une étape d'agrégation fusionne, compare ou filtre les sorties.
4. Le système renvoie la réponse finale à l'utilisateur ou à l'agent en aval.

Ce système comporte une boucle de raisonnement distribuée avec traçabilité, tolérance aux pannes et logique optionnelle de pondération ou de sélection des résultats.

Plats à emporter

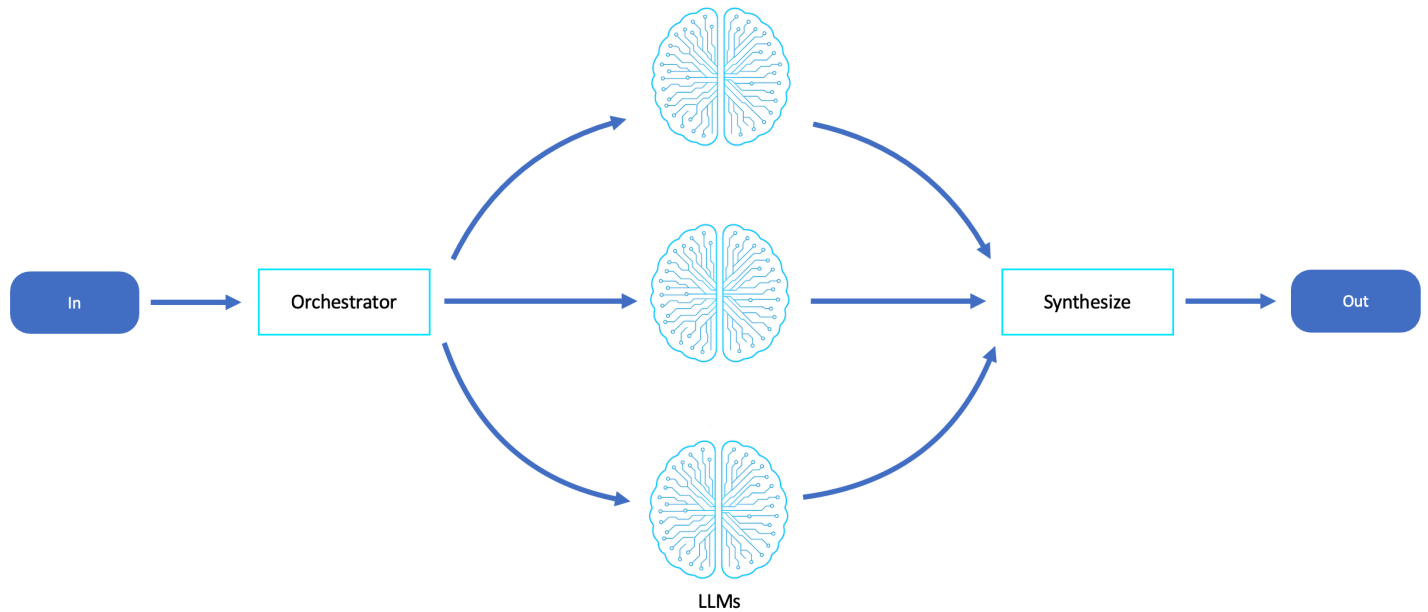
La parallélisation agentique utilise des modèles de dispersion pour distribuer les tâches LLM, permettant ainsi un traitement parallèle et une synthèse intelligente des résultats.

Modèles d'orchestration de Saga

Les flux de travail pilotés par LLMs deviennent de plus en plus complexes, impliquant des chaînes rapides, des étapes de traitement des données, des invocations d'outils et la collaboration entre

agents, le besoin d'une orchestration intelligente devient essentiel. Plutôt que de s'appuyer sur des scripts étroitement couplés ou des flux d'exécution statiques prédéterminés, ces flux de travail peuvent être mis en œuvre sous forme de modèles d'orchestration pilotés par des événements, permettant aux systèmes basés sur le LLM de coordonner, de surveiller et d'adapter de manière dynamique des tâches en plusieurs étapes entre des agents autonomes.

Le schéma suivant est un exemple d'orchestrateur :



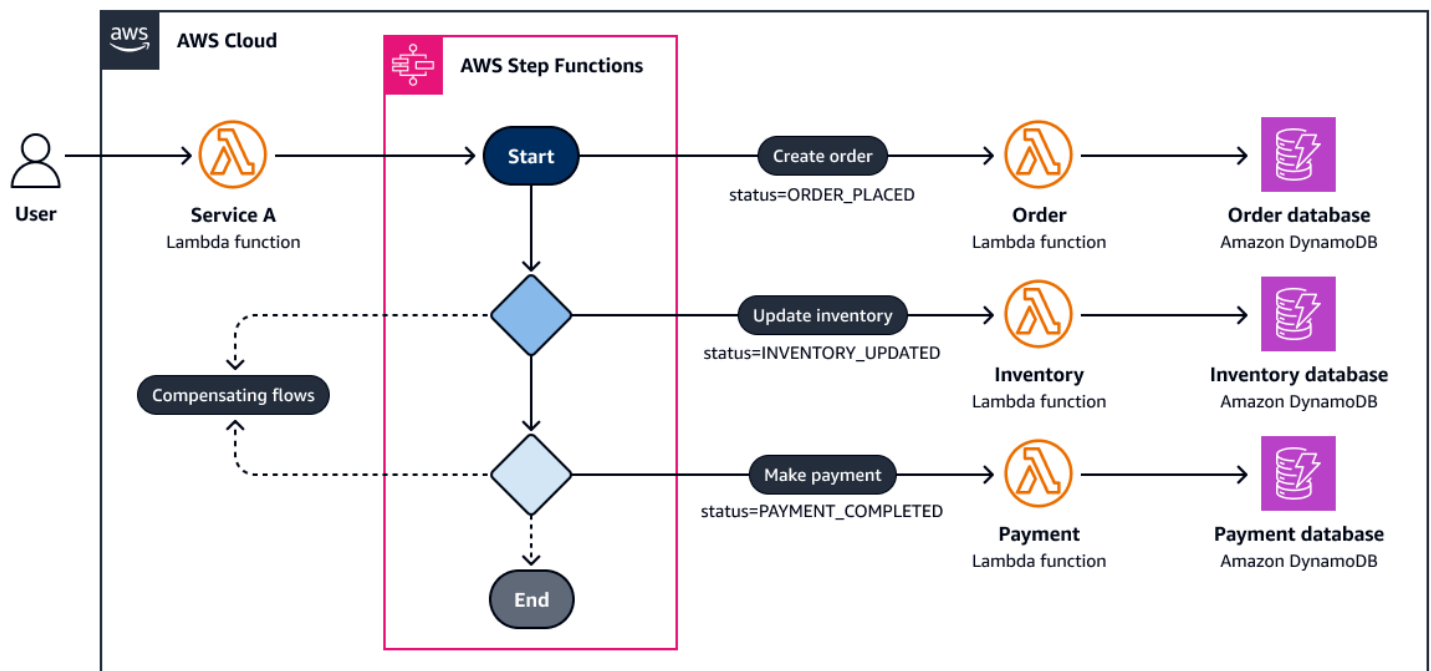
Orchestration d'événements

Dans les systèmes distribués traditionnels, l'orchestration d'événements fait référence à un modèle dans lequel un coordinateur central gère un flux de travail complexe en dirigeant explicitement le flux de contrôle entre plusieurs services ou tâches. Contrairement à la chorégraphie événementielle (où chaque service réagit indépendamment), l'orchestration fournit une logique, une visibilité et un contrôle centralisés sur l'ensemble du processus.

Ceci est généralement mis en œuvre à l'aide des outils suivants :

- AWS Step Functions— Définissez et exécutez des flux de travail dynamiques
- AWS Lambda— Effectuez des tâches discrètes dans le cadre du flux orchestré
- Amazon SQS ou Amazon EventBridge — Déclenche des étapes ou des réponses asynchrones

Le schéma suivant est un exemple d'orchestration de saga :



Un AWS Step Functions flux de travail gère le processus de commande d'un client :

1. Créer une commande (AWS Lambda)
2. Mettre à jour l'inventaire (AWS Lambda)
3. Effectuer le paiement (AWS Lambda)

L'orchestrateur coordonne chaque étape en gérant les nouvelles tentatives, les branches parallèles, les délais d'attente et les échecs.

Système d'agent basé sur les rôles (orchestrateur)

Dans les systèmes agenciques, le modèle d'orchestrateur reflète l'orchestration des événements mais répartit la logique entre plusieurs agents de raisonnement, chacun ayant un rôle ou une spécialisation définis. Un agent orchestrateur central interprète l'ensemble de la tâche, la décompose en sous-tâches et les délègue aux agents de travail, chacune étant optimisée pour un domaine particulier (par exemple, recherche, codage, synthèse, révision).

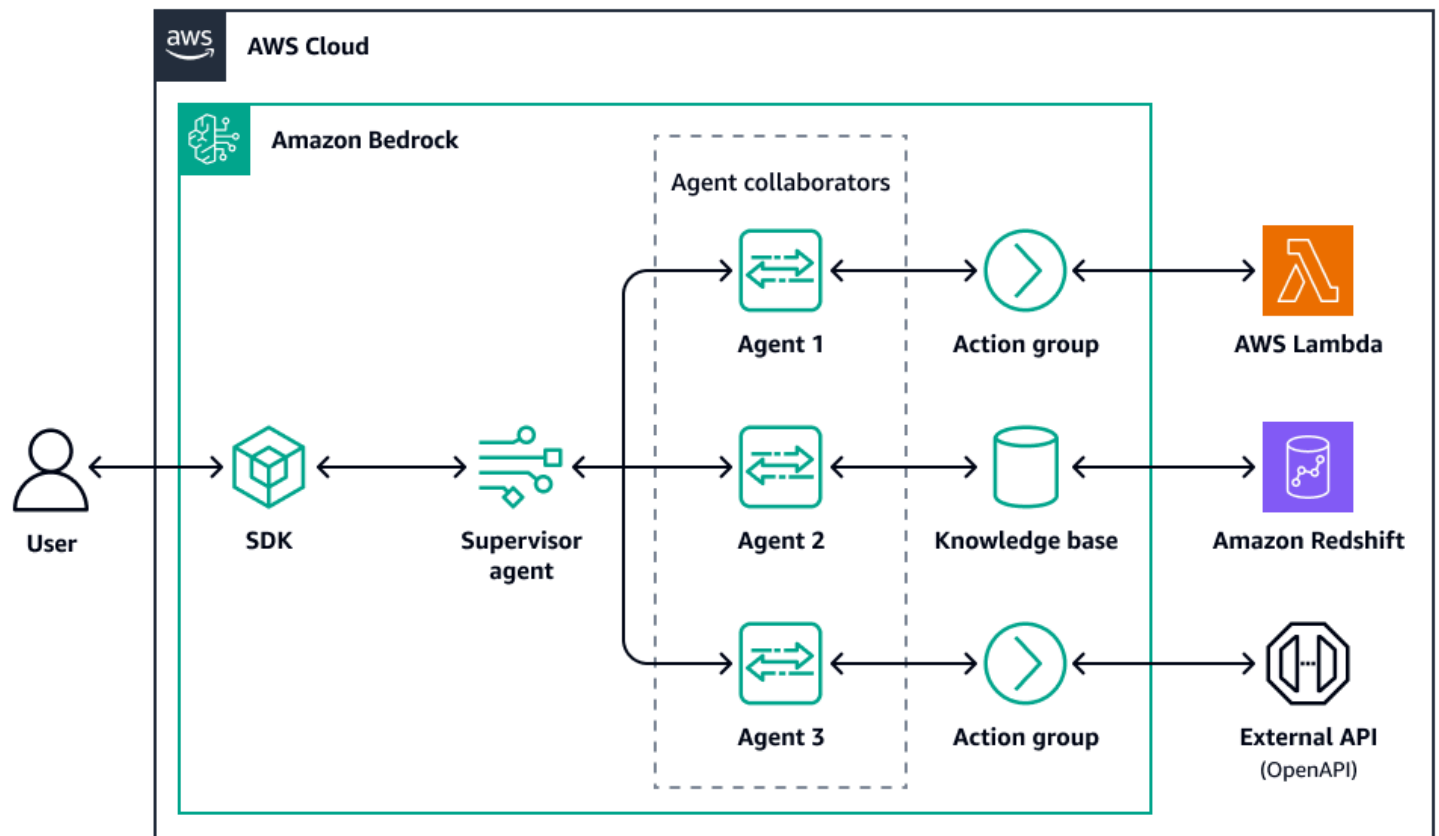
Superviseur

1. Un utilisateur soumet la requête « Créez un brief de projet et résumez les 5 principaux concurrents ».
2. L'agent Orchestrator effectue les opérations suivantes :

- Désigne un agent de recherche pour trouver les données des concurrents
- Envoie les résultats bruts à un agent de synthèse
- Transmet les résultats à un agent rédacteur
- Compile le résultat final pour l'utilisateur

Chaque agent fonctionne indépendamment, mais l'orchestrateur coordonne les tâches. Cela ressemble à une fonction Lambda qui gère les tâches du flux de travail.

Le schéma suivant est un exemple de superviseur :



1. Un utilisateur soumet une tâche à un agent superviseur Amazon Bedrock.
2. L'agent superviseur analyse la demande en sous-tâches pour chaque agent collaborateur.
3. Chaque sous-tâche est attribuée à un agent collaborateur avec des instructions ou des chaînes d'outils spécifiques au rôle.
4. Les agents ouvriers appellent des outils externes APIs ou des outils via un groupe d'action.
5. Chaque agent de travail renvoie le résultat dans un format structuré.

6. Lorsque tous les travailleurs renvoient leurs résultats, le superviseur évalue, synthétise et renvoie la réponse finale.

Cette structure permet la modularité, l'adaptabilité et l'introspection dans les flux de travail d'agents complexes en plusieurs étapes.

Plats à emporter

Lorsque l'orchestration d'événements utilise un contrôle centralisé (par exemple AWS Step Functions) pour diriger l'exécution des services, les systèmes d'agents basés sur les rôles utilisent un agent orchestrateur basé sur le LLM pour définir l'objectif, déléguer les sous-tâches aux agents ouvriers et synthétiser le résultat final.

Dans les deux paradigmes, l'orchestrateur effectue les opérations suivantes :

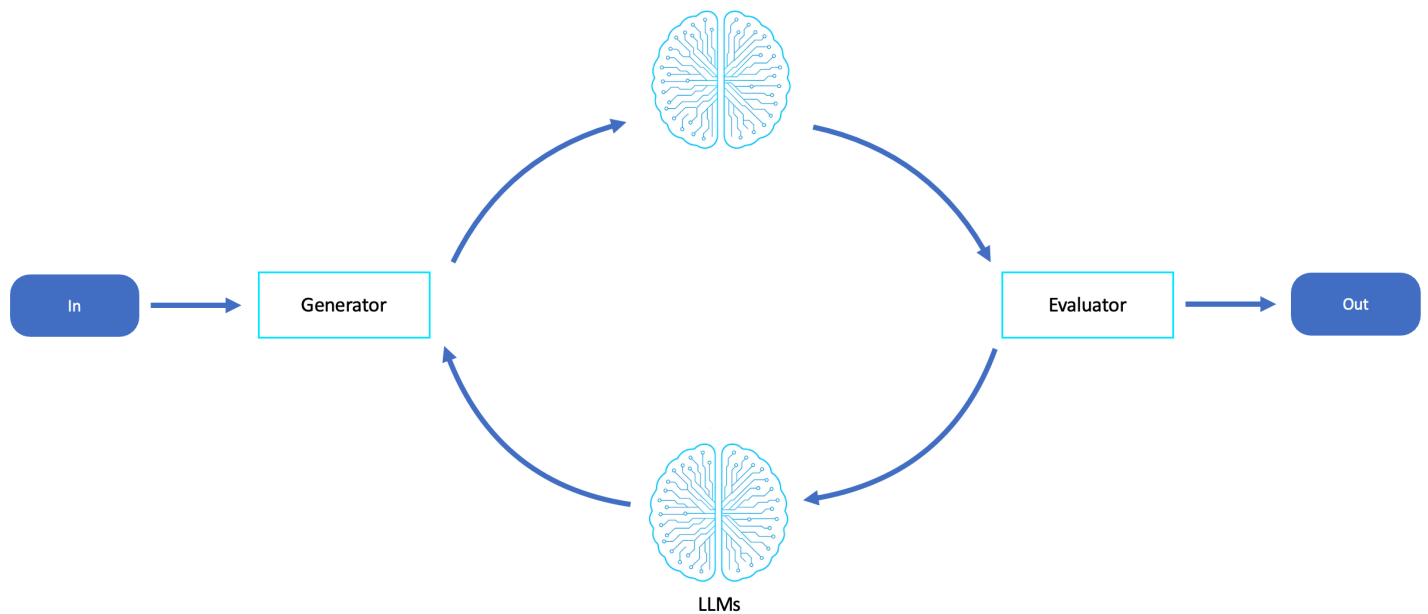
- Maintient le contexte et le flux d'exécution
- Gère le branchement, le séquençage et la gestion des erreurs
- Produit un résultat unifié à partir de composants distribués

L'orchestration agentique ajoute toutefois du raisonnement, de l'adaptabilité et de la délégation sémantique. Cela le rend parfaitement adapté aux tâches ouvertes, ambiguës et évolutives.

L'évaluateur reflète et affine les modèles de boucles

Les tâches telles que la génération de code, la synthèse ou la prise de décision autonome bénéficient grandement du feedback d'exécution, permettant au système d'évoluer grâce à l'observation et au raffinement. Pour rendre cela opérationnel, le cycle reflète-affiner peut être mis en œuvre sous la forme d'une boucle de contrôle de rétroaction pilotée par les événements, un modèle inspiré de l'ingénierie des systèmes, adapté aux flux de travail autonomes et intelligents.

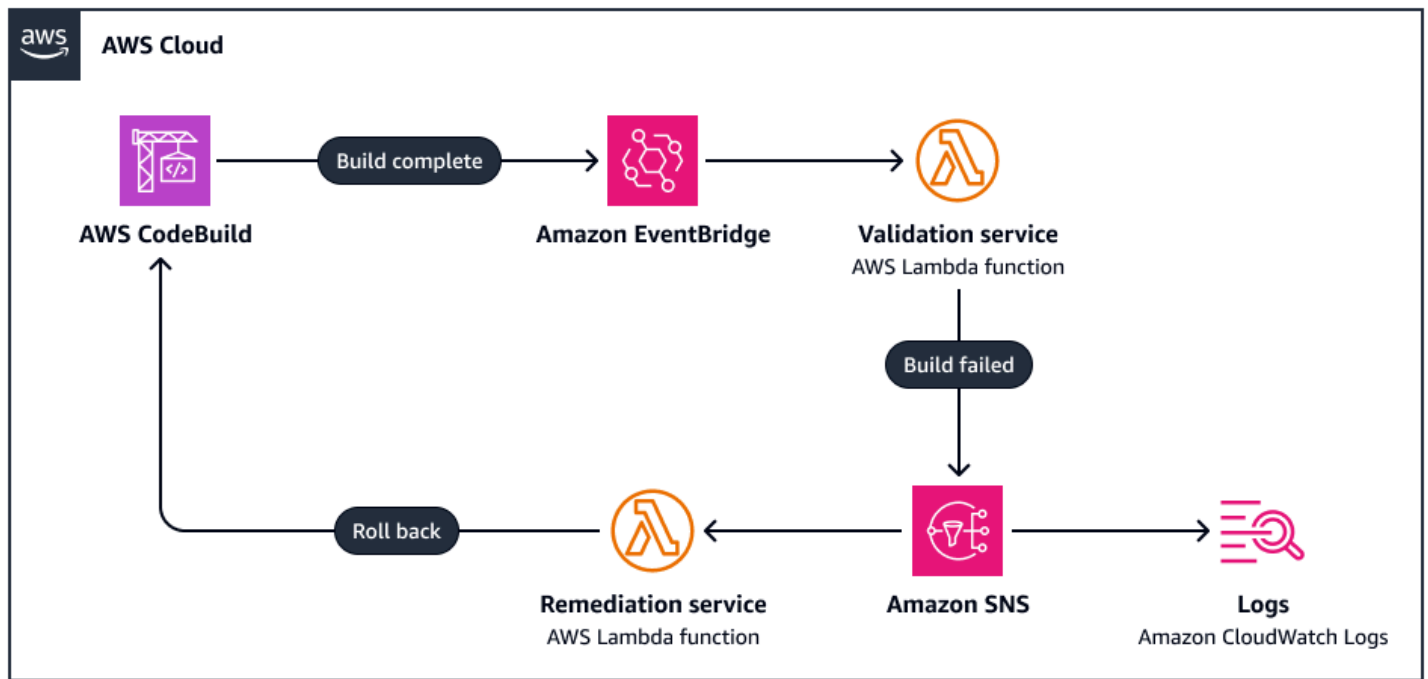
Le schéma suivant est un exemple de boucle de rétroaction basée sur l'affinement du reflet par un évaluateur :



Boucle de contrôle du feedback

Une boucle de contrôle par rétroaction est un modèle qui surveille ses propres sorties et comportements, les évalue par rapport à des critères définis ou à un état souhaité, puis ajuste ses actions en conséquence. Cette architecture s'inspire de la théorie du contrôle et est fondamentale dans des domaines tels que l'automatisation, les pipelines d'intégration continue et de livraison continue (CI/CD), ainsi que les opérations d'apprentissage automatique.

Le schéma suivant est un exemple de boucle de régulation de rétroaction :



1. Un pipeline de déploiement émet un événement BuildComplete.
2. L'événement déclenche un test ou une tâche d'évaluation automatique qui valide le build.
3. En cas d'échec de la validation (par exemple, en raison de l'échec des tests, de problèmes de sécurité ou d'une violation des règles), le système :
 - Émet un événement BuildComplete
 - Enregistre le problème ou envoie une notification
 - Déclenche une correction ou une action corrective, telle que l'annulation, l'application de correctifs ou une nouvelle tentative

La boucle continue jusqu'à ce qu'elle produise un résultat acceptable ou une escalade, ou jusqu'à ce qu'un délai d'attente se produise. Ce modèle est couramment utilisé dans les cas suivants :

- EventBridge Règles Amazon pour acheminer les événements vers des tâches d'évaluation ou de correction
- AWS Step Functions pour la logique itérative des nouvelles tentatives et le branchement sur les résultats de l'évaluation
- Amazon Simple Notification Service (Amazon SNS) ou alarmes Amazon pour les CloudWatch déclencheurs de commentaires et les alertes
- AWS Lambda fonctions ou travailleurs conteneurisés pour appliquer des mesures correctives

Boucle de régulation du feedback (évaluateur)

Un flux de travail d'évaluateur est une boucle de rétroaction cognitive alimentée par LLMs ou par des agents de raisonnement. Le processus comprend les étapes suivantes :

1. Un agent générateur ou LLM produit un résultat (par exemple, un plan, une réponse ou un brouillon).
2. Un agent évaluateur examine le résultat à l'aide d'une invite de critique ou d'une rubrique d'évaluation.
3. Sur la base des commentaires, l'agent d'origine ou un nouvel agent d'optimisation révise le résultat.

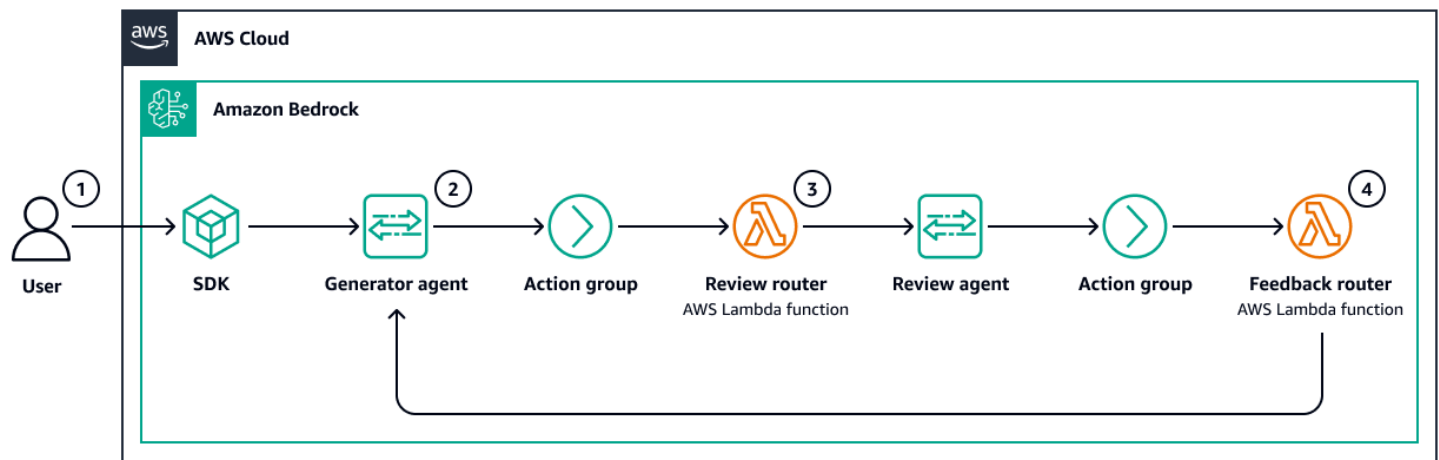
La boucle se répète jusqu'à ce que le résultat réponde à un ensemble de critères, soit approuvé ou atteigne une limite de nouvelles tentatives.

Évaluateur

1. Un utilisateur demande à un agent de rédiger un résumé de la politique.
2. L'agent générateur le rédige.
3. Un agent évaluateur vérifie la couverture, le ton et l'exactitude juridique.
4. Si la réponse est inadéquate, elle est affinée et soumise à nouveau jusqu'à ce que la boucle de rétroaction converge.

Cela permet une auto-évaluation, un raffinement itératif et un contrôle adaptatif des sorties, le tout sans intervention humaine.

Le schéma suivant est un exemple de boucle de régulation de rétroaction (évaluateur) :



1. Un utilisateur émet une tâche (par exemple, rédiger une stratégie commerciale).
2. Un agent Amazon Bedrock génère un brouillon initial à l'aide d'un LLM.
3. Un deuxième agent (ou une invite de suivi) effectue une évaluation structurée (par exemple, « évaluez ce résultat en fonction de sa clarté, de son exhaustivité et de son ton »).
4. Si la note tombe en dessous d'un seuil, la réponse est révisée comme suit :
 - Réinvoquer le générateur avec une critique intégrée
 - Envoi du feedback à un agent de raffinement spécialisé
 - Itérer jusqu'à ce qu'une réponse acceptable soit atteinte

Les composants optionnels tels que AWS Lambda les contrôleurs AWS Step Functions peuvent gérer les seuils de feedback, les nouvelles tentatives et les stratégies de repli.

Plats à emporter

Alors que les boucles de contrôle par rétroaction traditionnelles utilisent des événements, des métriques et une logique de correction pour valider et ajuster le comportement du système, les boucles d'évaluation agentic utilisent des agents de raisonnement pour évaluer, refléter et réviser les résultats de manière dynamique.

Dans les deux paradigmes :

- La sortie est évaluée une fois qu'elle a été générée
- Des actions correctives ou d'affinage sont déclenchées en fonction des commentaires
- Le système s'adapte en permanence à une qualité ou à un objectif cible

La version agentic transforme la validation statique en réflexion sémantique, permettant aux agents de s'améliorer eux-mêmes d'évaluer leur propre efficacité.

Conception de flux de travail agentiques sur AWS

Chaque modèle de ce guide peut être créé à l'aide de Services AWS. Les agents Amazon Bedrock fournissent des canaux d'orchestration, d'accès aux données et d'interaction.

Composant	Service AWS	Objectif
Raisonnement du LLM	Amazon Bedrock	Logique des agents, planification, utilisation des outils
Exécution de l'outil	AWS Lambda, Amazon ECS, Amazon SageMaker	Héberger des outils externes pour les agents
Mémoire et RAG	Base de connaissances Amazon Bedrock, Amazon S3, OpenSearch	Mémoire persistante et sémantique
Orchestration	AWS Step Functions	Coordination des tâches et des agents en plusieurs étapes
Routage des événements	Amazon EventBridge, Amazon SQS	Messagerie interagent découplée
Interface utilisateur	Amazon API Gateway AWS AppSync, SDK	Points d'entrée pour les applications ou les systèmes
Contrôle	Amazon CloudWatch AWS X-Ray, AWS Distro pour OpenTelemetry	Observabilité et introspection des agents

Conclusion

Les modèles de flux de travail agentiques constituent la prochaine étape évolutive des architectures axées sur les événements, dans lesquelles la logique métier n'est pas définie de manière statique

mais raisonnée dynamiquement grâce à l'utilisation d'une cognition améliorée par le biais d'un modèle de langage étendu (LLM). En combinant des primitives natives du cloud traditionnelles avec des flux de travail LLM et des modèles de conception d'agents, les entreprises peuvent créer des systèmes adaptatifs, intelligents et modulaires qui répondent à des objectifs précis et tirent les leçons de l'expérience.

Dans ces modèles, Amazon Bedrock est la passerelle vers la cognition agentique, permettant aux agents basés sur le LLM d'accéder aux flux de travail liés aux événements, d'interagir avec les outils et la mémoire, et de fournir des résultats structurés, traçables et alignés.

Lorsque vous concevez et déployez des systèmes agentiques, ces modèles de flux de travail fournissent des modèles pour la création d'architectures d'IA autonomes et composables. Ces systèmes sont basés sur les meilleures pratiques sans serveur et complétés par des modèles de base intelligents.

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
Publication initiale	—	14 juillet 2025

AWS Glossaire des directives prescriptives

Les termes suivants sont couramment utilisés dans les stratégies, les guides et les modèles fournis par les directives AWS prescriptives. Pour suggérer des entrées, veuillez utiliser le lien [Faire un commentaire](#) à la fin du glossaire.

Nombres

7 R

Sept politiques de migration courantes pour transférer des applications vers le cloud. Ces politiques s'appuient sur les 5 R identifiés par Gartner en 2011 et sont les suivantes :

- **Refactor/re-architect** — Déplacez une application et modifiez son architecture en tirant pleinement parti des fonctionnalités natives du cloud pour améliorer l'agilité, les performances et l'évolutivité. Cela implique généralement le transfert du système d'exploitation et de la base de données. Exemple : migrez votre base de données Oracle sur site vers l' PostgreSQL-Compatible édition Amazon Aurora.
- **Replatformer** (déplacer et remodeler) : transférez une application vers le cloud et introduisez un certain niveau d'optimisation pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle sur site vers Amazon Relational Database Service (Amazon RDS) pour Oracle dans le. AWS Cloud
- **Racheter** (rachat) : optez pour un autre produit, généralement en passant d'une licence traditionnelle à un modèle SaaS. Exemple : migrez votre système de gestion de la relation client (CRM) vers Salesforce.com.
- **Réhéberger** (lift and shift) : transférez une application vers le cloud sans apporter de modifications pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle sur site vers Oracle sur une instance EC2 dans le. AWS Cloud
- **Relocaliser** (lift and shift au niveau de l'hyperviseur) : transférez l'infrastructure vers le cloud sans acheter de nouveau matériel, réécrire des applications ou modifier vos opérations existantes. Vous migrez des serveurs d'une plateforme sur site vers un service cloud pour la même plateforme. Exemple : migrer une Microsoft Hyper-V application vers AWS.
- **Retenir** : conservez les applications dans votre environnement source. Il peut s'agir d'applications nécessitant une refactorisation majeure, que vous souhaitez retarder, et d'applications existantes que vous souhaitez retenir, car rien ne justifie leur migration sur le plan commercial.

- Retirer : mettez hors service ou supprimez les applications dont vous n'avez plus besoin dans votre environnement source.

A

A2 (1) Agent-to-Agent

Protocole dynamique pour la collaboration agent-agent prenant en charge la délégation de tâches et le transfert d'état.

ABAC

Voir contrôle [d'accès basé sur les attributs](#).

services abstraits

Consultez la section [Services gérés](#).

ACIDE

Voir [atomicité, consistance, isolation, durabilité](#).

migration active-active

Méthode de migration de base de données dans laquelle la synchronisation des bases de données source et cible est maintenue (à l'aide d'un outil de réplication bidirectionnelle ou d'opérations d'écriture double), tandis que les deux bases de données gèrent les transactions provenant de la connexion d'applications pendant la migration. Cette méthode prend en charge la migration par petits lots contrôlés au lieu d'exiger un basculement ponctuel. Elle est plus flexible mais demande plus de travail qu'une migration [active-passive](#).

migration active-passive

Méthode de migration de base de données dans laquelle les bases de données source et cible sont synchronisées, mais seule la base de données source gère les transactions liées à la connexion des applications pendant que les données sont répliquées vers la base de données cible. La base de données cible n'accepte aucune transaction pendant la migration.

Agent

Un système d'IA capable de raisonner, de planifier et de prendre des mesures de manière autonome à l'aide d'outils pour atteindre des objectifs.

Agent Ops

Pratiques opérationnelles pour la création, le test, le déploiement et l'exécution d'agents d'IA en production à grande échelle.

fonction d'agrégation

Fonction SQL qui agit sur un groupe de lignes et calcule une valeur de retour unique pour le groupe. Des exemples de fonctions d'agrégation incluent SUM et MAX.

AI

Voir [intelligence artificielle](#).

AIOps

Voir les [opérations d'intelligence artificielle](#).

anonymisation

Processus de suppression définitive d'informations personnelles dans un ensemble de données. L'anonymisation peut contribuer à protéger la vie privée. Les données anonymisées ne sont plus considérées comme des données personnelles.

anti-motif

Solution fréquemment utilisée pour un problème récurrent lorsque la solution est contre-productive, inefficace ou moins efficace qu'une solution alternative.

contrôle des applications

Une approche de sécurité qui permet d'utiliser uniquement des applications approuvées afin de protéger un système contre les logiciels malveillants.

portefeuille d'applications

Ensemble d'informations détaillées sur chaque application utilisée par une organisation, y compris le coût de génération et de maintenance de l'application, ainsi que sa valeur métier. Ces informations sont essentielles pour [le processus de découverte et d'analyse du portefeuille](#) et permettent d'identifier et de prioriser les applications à migrer, à moderniser et à optimiser.

intelligence artificielle (IA)

Domaine de l'informatique consacré à l'utilisation des technologies de calcul pour exécuter des fonctions cognitives généralement associées aux humains, telles que l'apprentissage, la résolution de problèmes et la reconnaissance de modèles. Pour plus d'informations, veuillez consulter [Qu'est-ce que l'intelligence artificielle ?](#)

opérations d'intelligence artificielle (AIOps)

Processus consistant à utiliser des techniques de machine learning pour résoudre les problèmes opérationnels, réduire les incidents opérationnels et les interventions humaines, mais aussi améliorer la qualité du service. Pour plus d'informations sur la façon dont les AIOps sont utilisées dans la stratégie de migration AWS, veuillez consulter le [guide d'intégration des opérations](#).

chiffrement asymétrique

Algorithme de chiffrement qui utilise une paire de clés, une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Vous pouvez partager la clé publique, car elle n'est pas utilisée pour le déchiffrement, mais l'accès à la clé privée doit être très restreint.

atomicité, cohérence, isolement, durabilité (ACID)

Ensemble de propriétés logicielles garantissant la validité des données et la fiabilité opérationnelle d'une base de données, même en cas d'erreur, de panne de courant ou d'autres problèmes.

contrôle d'accès par attributs (ABAC)

Pratique qui consiste à créer des autorisations détaillées en fonction des attributs de l'utilisateur, tels que le service, le poste et le nom de l'équipe. Pour plus d'informations, consultez [ABAC pour AWS](#) dans la documentation Gestion des identités et des accès AWS (IAM).

source de données faisant autorité

Emplacement où vous stockez la version principale des données, considérée comme la source d'information la plus fiable. Vous pouvez copier les données de la source de données officielle vers d'autres emplacements à des fins de traitement ou de modification des données, par exemple en les anonymisant, en les expurgant ou en les pseudonymisant.

Zone de disponibilité

Un emplacement distinct au sein d'une Région AWS réseau isolé des défaillances dans d'autres zones de disponibilité et fournissant une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même région.

AWS Cadre d'adoption du cloud (AWS CAF)

Un cadre de directives et de meilleures pratiques visant AWS à aider les entreprises à élaborer un plan efficace pour réussir leur migration vers le cloud. AWS La CAF organise ses conseils en six domaines prioritaires appelés perspectives : les affaires, les personnes, la gouvernance, les plateformes, la sécurité et les opérations. Les perspectives d'entreprise, de personnes et de gouvernance mettent l'accent sur les compétences et les processus métier, tandis que les

perspectives relatives à la plateforme, à la sécurité et aux opérations se concentrent sur les compétences et les processus techniques. Par exemple, la perspective liée aux personnes cible les parties prenantes qui s'occupent des ressources humaines (RH), des fonctions de dotation en personnel et de la gestion des personnes. Dans cette perspective, la AWS CAF fournit des conseils pour le développement du personnel, la formation et les communications afin de préparer l'organisation à une adoption réussie du cloud. Pour plus d'informations, veuillez consulter le [site Web AWS CAF](#) et le [livre blanc AWS CAF](#).

AWS Cadre de qualification de la charge de travail (AWS WQF)

Outil qui évalue les charges de travail liées à la migration des bases de données, recommande des stratégies de migration et fournit des estimations de travail. AWS Le WQF est inclus avec AWS Schema Conversion Tool (AWS SCT). Il analyse les schémas de base de données et les objets de code, le code d'application, les dépendances et les caractéristiques de performance, et fournit des rapports d'évaluation.

B

mauvais bot

Un [bot](#) destiné à perturber ou à nuire à des individus ou à des organisations.

BCP

Consultez la section [Planification de la continuité des activités](#).

graphique de comportement

Vue unifiée et interactive des comportements des ressources et des interactions au fil du temps. Vous pouvez utiliser un graphique de comportement avec Amazon Detective pour examiner les tentatives de connexion infructueuses, les appels d'API suspects et les actions similaires. Pour plus d'informations, veuillez consulter [Data in a behavior graph](#) dans la documentation Detective.

système de poids fort

Système qui stocke d'abord l'octet le plus significatif. Voir aussi [endianité](#).

classification binaire

Processus qui prédit un résultat binaire (l'une des deux classes possibles). Par exemple, votre modèle de machine learning peut avoir besoin de prévoir des problèmes tels que « Cet e-mail est-il du spam ou non ? » ou « Ce produit est-il un livre ou une voiture ? ».

filtre de Bloom

Structure de données probabiliste et efficace en termes de mémoire qui est utilisée pour tester si un élément fait partie d'un ensemble.

blue/green déploiement

Stratégie de déploiement dans laquelle vous créez deux environnements distincts mais identiques. Vous exécutez la version actuelle de l'application dans un environnement (bleu) et la nouvelle version de l'application dans l'autre environnement (vert). Cette stratégie vous permet de revenir rapidement en arrière avec un impact minimal.

bot

Application logicielle qui exécute des tâches automatisées sur Internet et simule l'activité ou l'interaction humaine. Certains robots sont utiles ou bénéfiques, comme les robots d'exploration Web qui indexent des informations sur Internet. D'autres robots, appelés « bots malveillants », sont destinés à perturber ou à nuire à des individus ou à des organisations.

botnet

Réseaux de [robots](#) infectés par des [logiciels malveillants](#) et contrôlés par une seule entité, connue sous le nom d'herder ou d'opérateur de bots. Les botnets sont le mécanisme le plus connu pour faire évoluer les bots et leur impact.

branche

Zone contenue d'un référentiel de code. La première branche créée dans un référentiel est la branche principale. Vous pouvez créer une branche à partir d'une branche existante, puis développer des fonctionnalités ou corriger des bogues dans la nouvelle branche. Une branche que vous créez pour générer une fonctionnalité est communément appelée branche de fonctionnalités. Lorsque la fonctionnalité est prête à être publiée, vous fusionnez à nouveau la branche de fonctionnalités dans la branche principale. Pour plus d'informations, consultez [À propos des branches](#) (GitHub documentation).

accès par brise-vitre

Dans des circonstances exceptionnelles et par le biais d'un processus approuvé, c'est un moyen rapide pour un utilisateur d'accéder à un accès auquel Compte AWS il n'est généralement pas autorisé. Pour plus d'informations, consultez l'indicateur [Mettre en œuvre des procédures permettant de briser le verre](#) dans le AWS Well-Architected guide.

stratégie existante (brownfield)

L'infrastructure existante de votre environnement. Lorsque vous adoptez une stratégie existante pour une architecture système, vous concevez l'architecture en fonction des contraintes des systèmes et de l'infrastructure actuels. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et [greenfield](#) (inédites).

cache de tampon

Zone de mémoire dans laquelle sont stockées les données les plus fréquemment consultées.

capacité métier

Ce que fait une entreprise pour générer de la valeur (par exemple, les ventes, le service client ou le marketing). Les architectures de microservices et les décisions de développement peuvent être dictées par les capacités métier. Pour plus d'informations, veuillez consulter la section [Organisation en fonction des capacités métier](#) du livre blanc [Exécution de microservices conteneurisés sur AWS](#).

planification de la continuité des activités (BCP)

Plan qui tient compte de l'impact potentiel d'un événement perturbateur, tel qu'une migration à grande échelle, sur les opérations, et qui permet à une entreprise de reprendre ses activités rapidement.

C

CAF

Voir le [cadre d'adoption du AWS cloud](#).

déploiement de Canary

Diffusion lente et progressive d'une version pour les utilisateurs finaux. Lorsque vous êtes sûr, vous déployez la nouvelle version et remplacez la version actuelle dans son intégralité.

CCoE

Voir [le Centre d'excellence du cloud](#).

CDC

Consultez la section [Capture des données de modification](#).

capture des données de modification (CDC)

Processus de suivi des modifications apportées à une source de données, telle qu'une table de base de données, et d'enregistrement des métadonnées relatives à ces modifications. Vous pouvez utiliser la CDC à diverses fins, telles que l'audit ou la réplication des modifications dans un système cible afin de maintenir la synchronisation.

ingénierie du chaos

Introduire intentionnellement des défaillances ou des événements perturbateurs pour tester la résilience d'un système. Vous pouvez utiliser [AWS Fault Injection Service \(AWS FIS\)](#) pour effectuer des expériences qui stressent vos AWS charges de travail et évaluer leur réponse.

CI/CD

Découvrez [l'intégration continue et la livraison continue](#).

classification

Processus de catégorisation qui permet de générer des prédictions. Les modèles de ML pour les problèmes de classification prédisent une valeur discrète. Les valeurs discrètes se distinguent toujours les unes des autres. Par exemple, un modèle peut avoir besoin d'évaluer la présence ou non d'une voiture sur une image.

Développeur citoyen

Un utilisateur professionnel qui crée des applications d'intelligence artificielle à l'aide de plateformes sans code/low code sans compétences techniques spécialisées.

chiffrement côté client

Chiffrement des données localement, avant que la cible ne les Service AWS reçoive.

Centre d'excellence cloud (CCoE)

Une équipe multidisciplinaire qui dirige les efforts d'adoption du cloud au sein d'une organisation, notamment en développant les bonnes pratiques en matière de cloud, en mobilisant des ressources, en établissant des délais de migration et en guidant l'organisation dans le cadre de transformations à grande échelle. Pour plus d'informations, consultez les [articles du CCoE](#) sur le blog de stratégie AWS Cloud d'entreprise.

cloud computing

Technologie cloud généralement utilisée pour le stockage de données à distance et la gestion des appareils IoT. Le cloud computing est généralement associé à la technologie [informatique de pointe](#).

modèle d'exploitation du cloud

Dans une organisation informatique, modèle d'exploitation utilisé pour créer, faire évoluer et optimiser un ou plusieurs environnements cloud. Pour plus d'informations, consultez la section [Création de votre modèle d'exploitation cloud](#).

étapes d'adoption du cloud

Les quatre phases que les entreprises traversent généralement lorsqu'elles migrent vers AWS Cloud :

- **Projet** : exécution de quelques projets liés au cloud à des fins de preuve de concept et d'apprentissage
- **Base** : réaliser des investissements fondamentaux pour mettre à l'échelle l'adoption du cloud (par exemple, en créant une zone de destination, en définissant un CCoE ou en établissant un modèle opérationnel)
- **Migration** : migration d'applications individuelles
- **Re-invention** — Optimisation des produits et services et innovation dans le cloud

Ces étapes ont été définies par Stephen Orban dans le billet de blog The [Journey Toward Cloud-First & the Stages of Adoption](#) publié sur le blog AWS Cloud Enterprise Strategy. Pour plus d'informations sur leur lien avec la stratégie de AWS migration, consultez le [guide de préparation à la migration](#).

CMDB

Consultez la base de [données de gestion des configurations](#).

référentiel de code

Emplacement où le code source et d'autres ressources, comme la documentation, les exemples et les scripts, sont stockés et mis à jour par le biais de processus de contrôle de version. Les référentiels cloud courants incluent GitHub ou Bitbucket Cloud. Chaque version du code est appelée branche. Dans une structure de microservice, chaque référentiel est consacré à une seule fonctionnalité. Un CI/CD pipeline unique peut utiliser plusieurs référentiels.

cache passif

Cache tampon vide, mal rempli ou contenant des données obsolètes ou non pertinentes. Cela affecte les performances, car l'instance de base de données doit lire à partir de la mémoire principale ou du disque, ce qui est plus lent que la lecture à partir du cache tampon.

données gelées

Données rarement consultées et généralement historiques. Lorsque vous interrogez ce type de données, les requêtes lentes sont généralement acceptables. Le transfert de ces données vers des niveaux ou classes de stockage moins performants et moins coûteux peut réduire les coûts.

vision par ordinateur (CV)

Domaine de l'[IA](#) qui utilise l'apprentissage automatique pour analyser et extraire des informations à partir de formats visuels tels que des images numériques et des vidéos. Par exemple, Amazon SageMaker AI fournit des algorithmes de traitement d'image pour les CV.

dérive de configuration

Pour une charge de travail, une modification de configuration par rapport à l'état attendu. Cela peut entraîner une non-conformité de la charge de travail, et cela est généralement progressif et involontaire.

base de données de gestion des configurations (CMDB)

Référentiel qui stocke et gère les informations relatives à une base de données et à son environnement informatique, y compris les composants matériels et logiciels ainsi que leurs configurations. Vous utilisez généralement les données d'une CMDB lors de la phase de découverte et d'analyse du portefeuille de la migration.

pack de conformité

Ensemble de AWS Config règles et d'actions correctives que vous pouvez assembler pour personnaliser vos contrôles de conformité et de sécurité. Vous pouvez déployer un pack de conformité en tant qu'entité unique dans une région Compte AWS et, ou au sein d'une organisation, à l'aide d'un modèle YAML. Pour plus d'informations, consultez la section [Packs de conformité](#) dans la AWS Config documentation.

intégration continue et livraison continue (CI/CD)

Processus d'automatisation des étapes de source, de construction, de test, de préparation et de production du processus de publication du logiciel. CI/CD est communément décrit comme un pipeline. CI/CD peut vous aider à automatiser les processus, à améliorer la productivité, à améliorer la qualité du code et à accélérer les livraisons. Pour plus d'informations, veuillez consulter [Avantages de la livraison continue](#). CD peut également signifier déploiement continu. Pour plus d'informations, veuillez consulter [Livraison continue et déploiement continu](#).

CV

Voir [vision par ordinateur](#).

D

données au repos

Données stationnaires dans votre réseau, telles que les données stockées.

classification des données

Processus permettant d'identifier et de catégoriser les données de votre réseau en fonction de leur sévérité et de leur sensibilité. Il s'agit d'un élément essentiel de toute stratégie de gestion des risques de cybersécurité, car il vous aide à déterminer les contrôles de protection et de conservation appropriés pour les données. La classification des données est une composante du pilier de sécurité du AWS Well-Architected cadre. Pour plus d'informations, veuillez consulter [Classification des données](#).

dérive des données

Une variation significative entre les données de production et les données utilisées pour entraîner un modèle ML, ou une modification significative des données d'entrée au fil du temps. La dérive des données peut réduire la qualité, la précision et l'équité globales des prédictions des modèles ML.

données en transit

Données qui circulent activement sur votre réseau, par exemple entre les ressources du réseau.

maillage de données

Un cadre architectural qui fournit une propriété des données distribuée et décentralisée avec une gestion et une gouvernance centralisées.

minimisation des données

Le principe de collecte et de traitement des seules données strictement nécessaires. La pratique de la minimisation des données AWS Cloud peut réduire les risques liés à la confidentialité, les coûts et l'empreinte carbone de vos analyses.

périmètre de données

Ensemble de garde-fous préventifs dans votre AWS environnement qui permettent de garantir que seules les identités fiables accèdent aux ressources fiables des réseaux attendus. Pour plus d'informations, voir [Création d'un périmètre de données sur AWS](#).

prétraitement des données

Pour transformer les données brutes en un format facile à analyser par votre modèle de ML. Le prétraitement des données peut impliquer la suppression de certaines colonnes ou lignes et le traitement des valeurs manquantes, incohérentes ou en double.

provenance des données

Le processus de suivi de l'origine et de l'historique des données tout au long de leur cycle de vie, par exemple la manière dont les données ont été générées, transmises et stockées.

sujet des données

Personne dont les données sont collectées et traitées.

entrepôt des données

Un système de gestion des données qui prend en charge les informations commerciales, telles que les analyses. Les entrepôts de données contiennent généralement de grandes quantités de données historiques et sont généralement utilisés pour les requêtes et les analyses.

langage de définition de base de données (DDL)

Instructions ou commandes permettant de créer ou de modifier la structure des tables et des objets dans une base de données.

langage de manipulation de base de données (DML)

Instructions ou commandes permettant de modifier (insérer, mettre à jour et supprimer) des informations dans une base de données.

DDL

Voir [langage de définition de base](#) de données.

ensemble profond

Sert à combiner plusieurs modèles de deep learning à des fins de prédiction. Vous pouvez utiliser des ensembles profonds pour obtenir une prévision plus précise ou pour estimer l'incertitude des prédictions.

deep learning

Un sous-champ de ML qui utilise plusieurs couches de réseaux neuronaux artificiels pour identifier le mappage entre les données d'entrée et les variables cibles d'intérêt.

défense en profondeur

Approche de la sécurité de l'information dans laquelle une série de mécanismes et de contrôles de sécurité sont judicieusement répartis sur l'ensemble d'un réseau informatique afin de protéger la confidentialité, l'intégrité et la disponibilité du réseau et des données qu'il contient. Lorsque vous adoptez cette stratégie AWS, vous ajoutez plusieurs contrôles à différentes couches de la AWS Organizations structure afin de sécuriser les ressources. Par exemple, une approche de défense approfondie peut combiner l'authentification multifactorielle, la segmentation du réseau et le chiffrement.

administrateur délégué

Dans AWS Organizations, un service compatible peut enregistrer un compte AWS membre pour administrer les comptes de l'organisation et gérer les autorisations pour ce service. Ce compte est appelé administrateur délégué pour ce service. Pour plus d'informations et une liste des services compatibles, veuillez consulter la rubrique [Services qui fonctionnent avec AWS Organizations](#) dans la documentation AWS Organizations .

déploiement

Processus de mise à disposition d'une application, de nouvelles fonctionnalités ou de corrections de code dans l'environnement cible. Le déploiement implique la mise en œuvre de modifications dans une base de code, puis la génération et l'exécution de cette base de code dans les environnements de l'application.

environnement de développement

Voir [environnement](#).

contrôle de détection

Contrôle de sécurité conçu pour détecter, journaliser et alerter après la survenue d'un événement. Ces contrôles constituent une deuxième ligne de défense et vous alertent en cas d'événements de sécurité qui ont contourné les contrôles préventifs en place. Pour plus d'informations, veuillez consulter la rubrique [Contrôles de détection](#) dans Implementing security controls on AWS.

cartographie de la chaîne de valeur du développement (DVSM)

Processus utilisé pour identifier et hiérarchiser les contraintes qui nuisent à la rapidité et à la qualité du cycle de vie du développement logiciel. DVSM étend le processus de cartographie de la chaîne de valeur initialement conçu pour les pratiques de production allégée. Il met l'accent sur les étapes et les équipes nécessaires pour créer et transférer de la valeur tout au long du processus de développement logiciel.

jumeau numérique

Représentation virtuelle d'un système réel, tel qu'un bâtiment, une usine, un équipement industriel ou une ligne de production. Les jumeaux numériques prennent en charge la maintenance prédictive, la surveillance à distance et l'optimisation de la production.

tableau des dimensions

Dans un [schéma en étoile](#), table plus petite contenant les attributs de données relatifs aux données quantitatives d'une table de faits. Les attributs des tables de dimensions sont généralement des champs de texte ou des nombres discrets qui se comportent comme du texte. Ces attributs sont couramment utilisés pour la contrainte des requêtes, le filtrage et l'étiquetage des ensembles de résultats.

catastrophe

Un événement qui empêche une charge de travail ou un système d'atteindre ses objectifs commerciaux sur son site de déploiement principal. Ces événements peuvent être des catastrophes naturelles, des défaillances techniques ou le résultat d'actions humaines, telles qu'une mauvaise configuration involontaire ou une attaque de logiciel malveillant.

reprise après sinistre (DR)

La stratégie et le processus que vous utilisez pour minimiser les temps d'arrêt et les pertes de données causés par un [sinistre](#). Pour plus d'informations, consultez la section [Reprise après sinistre des charges de travail sur AWS : Restauration dans le cloud](#) dans le AWS Well-Architected Framework.

DML

Voir [langage de manipulation de base](#) de données.

conception axée sur le domaine

Approche visant à développer un système logiciel complexe en connectant ses composants à des domaines évolutifs, ou objectifs métier essentiels, que sert chaque composant. Ce concept a été introduit par Eric Evans dans son livre, Domain-Driven Design : Tackling Complexity in the Heart of Software (Boston : Addison-Wesley Professional, 2003). Pour plus d'informations sur la manière dont vous pouvez utiliser la conception axée sur le domaine avec le modèle Strangler Fig, consultez la section [Modernisation incrémentielle des anciens services Web ASP.NET Microsoft \(ASMX\) à l'aide de conteneurs et d'Amazon API Gateway](#).

DR

Consultez la section [Reprise après sinistre](#).

détection de dérive

Suivi des écarts par rapport à une configuration de référence. Par exemple, vous pouvez l'utiliser AWS CloudFormation pour [détecter la dérive des ressources du système](#) ou AWS Control Tower pour [détecter les modifications de votre zone d'atterrissage](#) susceptibles d'affecter le respect des exigences de gouvernance.

DVSM

Voir la [cartographie de la chaîne de valeur du développement](#).

E

EDA

Voir [analyse exploratoire des données](#).

EDI

Voir échange [de données informatisé](#).

informatique de périphérie

Technologie qui augmente la puissance de calcul des appareils intelligents en périphérie d'un réseau IoT. Comparé au [cloud computing, l'informatique](#) de pointe peut réduire la latence des communications et améliorer le temps de réponse.

échange de données informatisé (EDI)

L'échange automatique de documents commerciaux entre les organisations. Pour plus d'informations, voir [Qu'est-ce que l'échange de données informatisé ?](#)

chiffrement

Processus informatique qui transforme des données en texte clair, lisibles par l'homme, en texte chiffré.

clé de chiffrement

Chaîne cryptographique de bits aléatoires générée par un algorithme cryptographique. La longueur des clés peut varier, et chaque clé est conçue pour être imprévisible et unique.

endianisme

Ordre dans lequel les octets sont stockés dans la mémoire de l'ordinateur. Big-endian les systèmes stockent d'abord l'octet le plus significatif. Little-endian les systèmes stockent d'abord l'octet le moins significatif.

point de terminaison

Voir [point de terminaison de service](#).

service de point de terminaison

Service que vous pouvez héberger sur un cloud privé virtuel (VPC) pour le partager avec d'autres utilisateurs. Vous pouvez créer un service de point de terminaison avec AWS PrivateLink et accorder des autorisations à d'autres principaux Comptes AWS ou à Gestion des identités et des accès AWS (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de terminaison de manière privée en créant des points de terminaison d'un VPC d'interface. Pour plus d'informations, veuillez consulter [Création d'un service de point de terminaison](#) dans la documentation Amazon Virtual Private Cloud (Amazon VPC).

planification des ressources d'entreprise (ERP)

Système qui automatise et gère les principaux processus métier (tels que la comptabilité, le [MES](#) et la gestion de projet) pour une entreprise.

chiffrement d'enveloppe

Processus de chiffrement d'une clé de chiffrement à l'aide d'une autre clé de chiffrement. Pour plus d'informations, consultez la section [Chiffrement des enveloppes](#) dans la documentation AWS Key Management Service (AWS KMS).

environnement

Instance d'une application en cours d'exécution. Les types d'environnement les plus courants dans le cloud computing sont les suivants :

- Environnement de développement : instance d'une application en cours d'exécution à laquelle seule l'équipe principale chargée de la maintenance de l'application peut accéder. Les environnements de développement sont utilisés pour tester les modifications avant de les promouvoir dans les environnements supérieurs. Ce type d'environnement est parfois appelé environnement de test.
- Environnements inférieurs : tous les environnements de développement d'une application, tels que ceux utilisés pour les générations et les tests initiaux.

- Environnement de production : instance d'une application en cours d'exécution à laquelle les utilisateurs finaux peuvent accéder. Dans un CI/CD pipeline, l'environnement de production est le dernier environnement de déploiement.
- Environnements supérieurs : tous les environnements accessibles aux utilisateurs autres que l'équipe de développement principale. Ils peuvent inclure un environnement de production, des environnements de préproduction et des environnements pour les tests d'acceptation par les utilisateurs.

épopée

Dans les méthodologies agiles, catégories fonctionnelles qui aident à organiser et à prioriser votre travail. Les épopées fournissent une description détaillée des exigences et des tâches d'implémentation. Par exemple, les points forts de la AWS CAF en matière de sécurité incluent la gestion des identités et des accès, les contrôles de détection, la sécurité des infrastructures, la protection des données et la réponse aux incidents. Pour plus d'informations sur les épopées dans la stratégie de migration AWS , veuillez consulter le [guide d'implémentation du programme](#).

ERP

Voir [Planification des ressources d'entreprise](#).

analyse exploratoire des données (EDA)

Processus d'analyse d'un jeu de données pour comprendre ses principales caractéristiques. Vous collectez ou agrégez des données, puis vous effectuez des enquêtes initiales pour trouver des modèles, détecter des anomalies et vérifier les hypothèses. L'EDA est réalisée en calculant des statistiques récapitulatives et en créant des visualisations de données.

F

tableau des faits

La table centrale dans un [schéma en étoile](#). Il stocke des données quantitatives sur les opérations commerciales. Généralement, une table de faits contient deux types de colonnes : celles qui contiennent des mesures et celles qui contiennent une clé étrangère pour une table de dimensions.

échouer rapidement

Une philosophie qui utilise des tests fréquents et progressifs pour réduire le cycle de vie du développement. C'est un élément essentiel d'une approche agile.

limite d'isolation des défauts

Dans le AWS Cloud, une limite telle qu'une zone de disponibilité Région AWS, un plan de contrôle ou un plan de données qui limite l'effet d'une panne et contribue à améliorer la résilience des charges de travail. Pour plus d'informations, consultez la section [Limites d'isolation des AWS pannes](#).

branche de fonctionnalités

Voir [la succursale](#).

fonctionnalités

Les données d'entrée que vous utilisez pour faire une prédiction. Par exemple, dans un contexte de fabrication, les fonctionnalités peuvent être des images capturées périodiquement à partir de la ligne de fabrication.

importance des fonctionnalités

Le niveau d'importance d'une fonctionnalité pour les prédictions d'un modèle. Il s'exprime généralement sous la forme d'un score numérique qui peut être calculé à l'aide de différentes techniques, telles que la méthode Shapley Additive Explanations (SHAP) et les gradients intégrés. Pour plus d'informations, voir [Interprétabilité du modèle d'apprentissage automatique avec AWS](#).

transformation de fonctionnalité

Optimiser les données pour le processus de ML, notamment en enrichissant les données avec des sources supplémentaires, en mettant à l'échelle les valeurs ou en extrayant plusieurs ensembles d'informations à partir d'un seul champ de données. Cela permet au modèle de ML de tirer parti des données. Par exemple, si vous décomposez la date « 2021-05-27 00:15:37 » en « 2021 », « mai », « jeudi » et « 15 », vous pouvez aider l'algorithme d'apprentissage à apprendre des modèles nuancés associés à différents composants de données.

invitation en quelques coups

Fournir à un [LLM](#) un petit nombre d'exemples illustrant la tâche et le résultat souhaité avant de lui demander d'effectuer une tâche similaire. Cette technique est une application de l'apprentissage contextuel, dans le cadre de laquelle les modèles apprennent à partir d'exemples (prises de vue) intégrés dans des instructions. Few-shot l'envoi d'instructions peut être efficace pour les tâches qui nécessitent un formatage, un raisonnement ou une connaissance du domaine spécifiques. Voir également l'[invite Zero-Shot](#).

FGAC

Découvrez le [contrôle d'accès détaillé](#).

contrôle d'accès détaillé (FGAC)

Utilisation de plusieurs conditions pour autoriser ou refuser une demande d'accès.

migration instantanée (flash-cut)

Méthode de migration de base de données qui utilise la réplication continue des données par [le biais de la capture des données de modification](#) afin de migrer les données dans les plus brefs délais, au lieu d'utiliser une approche progressive. L'objectif est de réduire au maximum les temps d'arrêt.

FM

Voir le [modèle de fondation](#).

modèle de fondation (FM)

Un vaste réseau neuronal d'apprentissage profond qui s'entraîne sur des ensembles de données massifs de données généralisées et non étiquetées. Les FM sont capables d'effectuer une grande variété de tâches générales, telles que la compréhension du langage, la génération de texte et d'images et la conversation en langage naturel. Pour plus d'informations, voir [Que sont les modèles de base ?](#)

Passerelle FM

Un intermédiaire centralisé qui contrôle et normalise l'accès aux [modèles de base](#). Également connue sous le nom de passerelle LLM.

G

IA générative

Sous-ensemble de modèles d'[IA](#) qui ont été entraînés sur de grandes quantités de données et qui peuvent utiliser une simple invite textuelle pour créer de nouveaux contenus et artefacts, tels que des images, des vidéos, du texte et du son. Pour plus d'informations, consultez [Qu'est-ce que l'IA générative](#).

blocage géographique

Voir les [restrictions géographiques](#).

restrictions géographiques (blocage géographique)

Sur Amazon CloudFront, option permettant d'empêcher les utilisateurs de certains pays d'accéder aux distributions de contenu. Vous pouvez utiliser une liste d'autorisation ou une liste de blocage

pour spécifier les pays approuvés et interdits. Pour plus d'informations, consultez [la section Restreindre la distribution géographique de votre contenu](#) dans la CloudFront documentation.

Flux de travail Gitflow

Approche dans laquelle les environnements inférieurs et supérieurs utilisent différentes branches dans un référentiel de code source. Le flux de travail Gitflow est considéré comme existant, et le [flux de travail basé sur les troncs](#) est l'approche moderne préférée.

image dorée

Un instantané d'un système ou d'un logiciel utilisé comme modèle pour déployer de nouvelles instances de ce système ou logiciel. Par exemple, dans le secteur de la fabrication, une image dorée peut être utilisée pour fournir des logiciels sur plusieurs appareils et contribue à améliorer la vitesse, l'évolutivité et la productivité des opérations de fabrication des appareils.

stratégie inédite

L'absence d'infrastructures existantes dans un nouvel environnement. Lorsque vous adoptez une stratégie inédite pour une architecture système, vous pouvez sélectionner toutes les nouvelles technologies sans restriction de compatibilité avec l'infrastructure existante, également appelée [brownfield](#). Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

barrière de protection

Règle de haut niveau qui permet de régir les ressources, les politiques et la conformité au sein des unités d'organisation (UO). Les barrières de protection préventives appliquent des politiques pour garantir l'alignement sur les normes de conformité. Elles sont mises en œuvre à l'aide de politiques de contrôle des services et de limites des autorisations IAM. Les barrières de protection de détection détectent les violations des politiques et les problèmes de conformité, et génèrent des alertes pour y remédier. Ils sont implémentés à l'aide d'Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, d'Amazon Inspector et de AWS Lambda contrôles personnalisés.

rambardes (AI)

Des mécanismes de sécurité qui filtrent, valident et limitent les entrées et sorties des [agents](#) afin de garantir un comportement responsable et sûr de l'IA.

H

HA

Découvrez [la haute disponibilité](#).

migration de base de données hétérogène

Migration de votre base de données source vers une base de données cible qui utilise un moteur de base de données différent (par exemple, Oracle vers Amazon Aurora). La migration hétérogène fait généralement partie d'un effort de réarchitecture, et la conversion du schéma peut s'avérer une tâche complexe. [AWS propose AWS SCT](#) qui facilite les conversions de schémas.

haute disponibilité (HA)

Capacité d'une charge de travail à fonctionner en continu, sans intervention, en cas de difficultés ou de catastrophes. Les systèmes HA sont conçus pour basculer automatiquement, fournir constamment des performances de haute qualité et gérer différentes charges et défaillances avec un impact minimal sur les performances.

modernisation des historiens

Approche utilisée pour moderniser et mettre à niveau les systèmes de technologie opérationnelle (OT) afin de mieux répondre aux besoins de l'industrie manufacturière. Un historien est un type de base de données utilisé pour collecter et stocker des données provenant de diverses sources dans une usine.

données de rétention

Partie de données historiques étiquetées qui n'est pas divulguée dans un ensemble de données utilisé pour entraîner un modèle d'[apprentissage automatique](#). Vous pouvez utiliser les données de blocage pour évaluer les performances du modèle en comparant les prévisions du modèle aux données de blocage.

humain dans la boucle (HiTL)

Un modèle de flux de travail dans lequel l'exécution des [agents](#) s'arrête pour examen et approbation par l'homme aux points de décision critiques.

migration de base de données homogène

Migration de votre base de données source vers une base de données cible qui partage le même moteur de base de données (par exemple, Microsoft SQL Server vers Amazon RDS for SQL Server). La migration homogène s'inscrit généralement dans le cadre d'un effort de

réhébergement ou de replateforme. Vous pouvez utiliser les utilitaires de base de données natifs pour migrer le schéma.

données chaudes

Données fréquemment consultées, telles que les données en temps réel ou les données translationnelles récentes. Ces données nécessitent généralement un niveau ou une classe de stockage à hautes performances pour fournir des réponses rapides aux requêtes.

correctif

Solution d'urgence à un problème critique dans un environnement de production. En raison de son urgence, un correctif est généralement créé en dehors du flux de travail de DevOps publication habituel.

période de soins intensifs

Immédiatement après le basculement, période pendant laquelle une équipe de migration gère et surveille les applications migrées dans le cloud afin de résoudre les problèmes éventuels. En règle générale, cette période dure de 1 à 4 jours. À la fin de la période de soins intensifs, l'équipe de migration transfère généralement la responsabilité des applications à l'équipe des opérations cloud.

I

IaC

Considérez [l'infrastructure comme un code](#).

politique basée sur l'identité

Politique attachée à un ou plusieurs principaux IAM qui définit leurs autorisations au sein de l'AWS Cloud environnement.

application inactive

Application dont l'utilisation moyenne du processeur et de la mémoire se situe entre 5 et 20 % sur une période de 90 jours. Dans un projet de migration, il est courant de retirer ces applications ou de les retenir sur site.

IIoT

Voir [Internet industriel des objets](#).

infrastructure immuable

Modèle qui déploie une nouvelle infrastructure pour les charges de travail de production au lieu de mettre à jour, d'appliquer des correctifs ou de modifier l'infrastructure existante. Les infrastructures immuables sont intrinsèquement plus cohérentes, fiables et prévisibles que les infrastructures [mutables](#). Pour plus d'informations, consultez les meilleures pratiques de [déploiement à l'aide d'une infrastructure immuable](#) dans le AWS Well-Architected Framework.

VPC entrant (d'entrée)

Dans une architecture AWS multi-comptes, un VPC qui accepte, inspecte et achemine les connexions réseau depuis l'extérieur d'une application. L'[architecture de référence de sécurité AWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et Internet en général.

migration incrémentielle

Stratégie de basculement dans le cadre de laquelle vous migrez votre application par petites parties au lieu d'effectuer un basculement complet unique. Par exemple, il se peut que vous ne transfériez que quelques microservices ou utilisateurs vers le nouveau système dans un premier temps. Après avoir vérifié que tout fonctionne correctement, vous pouvez transférer progressivement des microservices ou des utilisateurs supplémentaires jusqu'à ce que vous puissiez mettre hors service votre système hérité. Cette stratégie réduit les risques associés aux migrations de grande ampleur.

Industry 4.0

Terme introduit par [Klaus Schwab](#) en 2016 pour désigner la modernisation des processus de fabrication grâce aux avancées en matière de connectivité, de données en temps réel, d'automatisation, d'analyse et. AI/ML

infrastructure

Ensemble des ressources et des actifs contenus dans l'environnement d'une application.

infrastructure en tant que code (IaC)

Processus de mise en service et de gestion de l'infrastructure d'une application via un ensemble de fichiers de configuration. IaC est conçue pour vous aider à centraliser la gestion de l'infrastructure, à normaliser les ressources et à mettre à l'échelle rapidement afin que les nouveaux environnements soient reproductibles, fiables et cohérents.

internet industriel des objets (IIoT)

L'utilisation de capteurs et d'appareils connectés à Internet dans les secteurs industriels tels que la fabrication, l'énergie, l'automobile, les soins de santé, les sciences de la vie et l'agriculture.

Pour plus d'informations, veuillez consulter [Building an industrial Internet of Things \(IIoT\) digital transformation strategy](#).

VPC d'inspection

Dans une architecture AWS multi-comptes, un VPC centralisé qui gère les inspections du trafic réseau entre les VPC (identiques ou Régions AWS différents), Internet et les réseaux sur site.

L'[architecture de référence de sécurité AWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et Internet en général.

Internet des objets (IoT)

Réseau d'objets physiques connectés dotés de capteurs ou de processeurs intégrés qui communiquent avec d'autres appareils et systèmes via Internet ou via un réseau de communication local. Pour plus d'informations, veuillez consulter la section [Qu'est-ce que l'IoT ?](#).

interprétabilité

Caractéristique d'un modèle de machine learning qui décrit dans quelle mesure un être humain peut comprendre comment les prédictions du modèle dépendent de ses entrées. Pour plus d'informations, voir [Interprétabilité du modèle d'apprentissage automatique avec AWS](#).

IoT

Voir [Internet des objets](#).

Bibliothèque d'informations informatiques (ITIL)

Ensemble de bonnes pratiques pour proposer des services informatiques et les aligner sur les exigences métier. L'ITIL constitue la base de l'ITSM.

gestion des services informatiques (ITSM)

Activités associées à la conception, à la mise en œuvre, à la gestion et à la prise en charge de services informatiques d'une organisation. Pour plus d'informations sur l'intégration des opérations cloud aux outils ITSM, veuillez consulter le [guide d'intégration des opérations](#).

ITIL

Consultez la [bibliothèque d'informations informatiques](#).

ITSM

Voir [Gestion des services informatiques](#).

L

contrôle d'accès basé sur des étiquettes (LBAC)

Une implémentation du contrôle d'accès obligatoire (MAC) dans laquelle une valeur d'étiquette de sécurité est explicitement attribuée aux utilisateurs et aux données elles-mêmes. L'intersection entre l'étiquette de sécurité utilisateur et l'étiquette de sécurité des données détermine les lignes et les colonnes visibles par l'utilisateur.

zone de destination

Une zone d'atterrissage est un AWS environnement multi-comptes bien conçu, évolutif et sécurisé. Il s'agit d'un point de départ à partir duquel vos entreprises peuvent rapidement lancer et déployer des charges de travail et des applications en toute confiance dans leur environnement de sécurité et d'infrastructure. Pour plus d'informations sur les zones de destination, veuillez consulter [Setting up a secure and scalable multi-account AWS environment](#).

grand modèle de langage (LLM)

Un modèle d'[intelligence artificielle basé](#) sur le deep learning qui est préentraîné sur une grande quantité de données. Un LLM peut effectuer plusieurs tâches, telles que répondre à des questions, résumer des documents, traduire du texte dans d'autres langues et compléter des phrases. Pour plus d'informations, voir [Que sont les LLM](#).

migration de grande envergure

Migration de 300 serveurs ou plus.

LBAC

Voir contrôle d'[accès basé sur des étiquettes](#).

principe de moindre privilège

Bonne pratique de sécurité qui consiste à accorder les autorisations minimales nécessaires à l'exécution d'une tâche. Pour plus d'informations, veuillez consulter la rubrique [Accorder les autorisations de moindre privilège](#) dans la documentation IAM.

lift and shift

Voir [7 Rs](#).

système de poids faible

Système qui stocke d'abord l'octet le moins significatif. Voir aussi [endianité](#).

LLM

Voir le [grand modèle de langage](#).

environnements inférieurs

Voir [environnement](#).

M

machine learning (ML)

Type d'intelligence artificielle qui utilise des algorithmes et des techniques pour la reconnaissance et l'apprentissage de modèles. Le ML analyse et apprend à partir de données enregistrées, telles que les données de l'Internet des objets (IoT), pour générer un modèle statistique basé sur des modèles. Pour plus d'informations, veuillez consulter [Machine Learning](#).

branche principale

Voir [la succursale](#).

malware

Logiciel conçu pour compromettre la sécurité ou la confidentialité de l'ordinateur. Les logiciels malveillants peuvent perturber les systèmes informatiques, divulguer des informations sensibles ou obtenir un accès non autorisé. Parmi les malwares, on peut citer les virus, les vers, les rançongiciels, les chevaux de Troie, les logiciels espions et les enregistreurs de frappe.

services gérés

Services AWS pour lequel AWS fonctionnent la couche d'infrastructure, le système d'exploitation et les plateformes, et vous accédez aux points de terminaison pour stocker et récupérer des données. Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB sont des exemples de services gérés. Ils sont également connus sous le nom de services abstraits.

système d'exécution de la fabrication (MES)

Un système logiciel pour le suivi, la surveillance, la documentation et le contrôle des processus de production qui convertissent les matières premières en produits finis dans l'atelier.

MAP

Voir [Migration Acceleration Program](#).

MCP

Voir [Model Context Protocol](#).

Protocole de contexte du modèle (MCP)

Protocole sans état pour la communication entre [un agent](#) et un [outil](#).

serveur MCP

Service qui expose un ou plusieurs [outils](#) via le [protocole Model Context](#).

mécanisme

Processus complet au cours duquel vous créez un outil, favorisez son adoption, puis inspectez les résultats afin de procéder aux ajustements nécessaires. Un mécanisme est un cycle qui se renforce et s'améliore au fur et à mesure de son fonctionnement. Pour plus d'informations, voir [Création de mécanismes](#) dans le AWS Well-Architected cadre.

compte membre

Tous, à l'exception des Comptes AWS exception du compte de gestion, qui font partie d'une organisation dans AWS Organizations. Un compte ne peut être membre que d'une seule organisation à la fois.

MAILLES

Voir le [système d'exécution de la fabrication](#).

Transport téléométrique en file d'attente de messages (MQTT)

[Un protocole de communication léger de machine à machine \(M2M\), basé sur le publish/subscribe modèle, pour les appareils IoT aux ressources limitées.](#)

microservice

Petit service indépendant qui communique via des API bien définies et qui est généralement détenu par de petites équipes autonomes. Par exemple, un système d'assurance peut inclure des microservices qui mappent à des capacités métier, telles que les ventes ou le marketing, ou à des sous-domaines, tels que les achats, les réclamations ou l'analytique. Les avantages des microservices incluent l'agilité, la flexibilité de la mise à l'échelle, la facilité de déploiement, la réutilisation du code et la résilience. Pour plus d'informations, consultez la section [Intégration de microservices à l'aide de services AWS sans serveur](#).

architecture de microservices

Approche de création d'une application avec des composants indépendants qui exécutent chaque processus d'application en tant que microservice. Ces microservices communiquent via une interface bien définie à l'aide d'API légères. Chaque microservice de cette architecture peut être mis à jour, déployé et mis à l'échelle pour répondre à la demande de fonctions spécifiques d'une application. Pour plus d'informations, consultez la section [Implémentation de microservices sur AWS](#).

Programme d'accélération des migrations (MAP)

Un AWS programme qui fournit un support de conseil, des formations et des services pour aider les entreprises à établir une base opérationnelle solide pour passer au cloud, et pour aider à compenser le coût initial des migrations. MAP inclut une méthodologie de migration pour exécuter les migrations héritées de manière méthodique, ainsi qu'un ensemble d'outils pour automatiser et accélérer les scénarios de migration courants.

migration à grande échelle

Processus consistant à transférer la majeure partie du portefeuille d'applications vers le cloud par vagues, un plus grand nombre d'applications étant déplacées plus rapidement à chaque vague. Cette phase utilise les bonnes pratiques et les enseignements tirés des phases précédentes pour implémenter une usine de migration d'équipes, d'outils et de processus en vue de rationaliser la migration des charges de travail grâce à l'automatisation et à la livraison agile. Il s'agit de la troisième phase de la [stratégie de migration AWS](#).

usine de migration

Cross-functional des équipes qui rationalisent la migration des charges de travail grâce à des approches automatisées et agiles. Les équipes de Migration Factory comprennent généralement des responsables des opérations, des analystes commerciaux et des propriétaires, des ingénieurs de migration, des développeurs et DevOps des professionnels travaillant dans le cadre de sprints. Entre 20 et 50 % du portefeuille d'applications d'entreprise est constitué de modèles répétés qui peuvent être optimisés par une approche d'usine. Pour plus d'informations, veuillez consulter la rubrique [discussion of migration factories](#) et le [guide Cloud Migration Factory](#) dans cet ensemble de contenus.

métadonnées de migration

Informations relatives à l'application et au serveur nécessaires pour finaliser la migration. Chaque modèle de migration nécessite un ensemble de métadonnées de migration différent. Les

exemples de métadonnées de migration incluent le sous-réseau cible, le groupe de sécurité et le AWS compte.

modèle de migration

Tâche de migration reproductible qui détaille la stratégie de migration, la destination de la migration et l'application ou le service de migration utilisé. Exemple : réorganisez la migration vers Amazon EC2 AWS avec le service de migration d'applications.

Évaluation du portefeuille de migration (MPA)

Outil en ligne qui fournit des informations pour valider l'analyse de rentabilisation en faveur de la migration vers le. AWS Cloud La MPA propose une évaluation détaillée du portefeuille (dimensionnement approprié des serveurs, tarification, comparaison du coût total de possession, analyse des coûts de migration), ainsi que la planification de la migration (analyse et collecte des données d'applications, regroupement des applications, priorisation des migrations et planification des vagues). L'[outil MPA](#) (connexion requise) est disponible gratuitement pour tous les AWS consultants et consultants APN Partner.

Évaluation de la préparation à la migration (MRA)

Processus qui consiste à obtenir des informations sur l'état de préparation d'une organisation au cloud, à identifier les forces et les faiblesses et à élaborer un plan d'action pour combler les lacunes identifiées, à l'aide du AWS CAF. Pour plus d'informations, veuillez consulter le [guide de préparation à la migration](#). La MRA est la première phase de la [stratégie de migration AWS](#).

stratégie de migration

L'approche utilisée pour migrer une charge de travail vers le AWS Cloud. Pour plus d'informations, reportez-vous aux [7 R](#) de ce glossaire et à [Mobiliser votre organisation pour accélérer les migrations à grande échelle](#).

ML

Voir [apprentissage automatique](#).

modernisation

Transformation d'une application obsolète (héritée ou monolithique) et de son infrastructure en un système agile, élastique et hautement disponible dans le cloud afin de réduire les coûts, de gagner en efficacité et de tirer parti des innovations. Pour plus d'informations, consultez [la section Stratégie de modernisation des applications dans le AWS Cloud](#).

évaluation de la préparation à la modernisation

Évaluation qui permet de déterminer si les applications d'une organisation sont prêtes à être modernisées, d'identifier les avantages, les risques et les dépendances, et qui détermine dans quelle mesure l'organisation peut prendre en charge l'état futur de ces applications. Le résultat de l'évaluation est un plan de l'architecture cible, une feuille de route détaillant les phases de développement et les étapes du processus de modernisation, ainsi qu'un plan d'action pour combler les lacunes identifiées. Pour plus d'informations, consultez la section [Évaluation de l'état de préparation à la modernisation des applications dans le AWS Cloud](#).

applications monolithiques (monolithes)

Applications qui s'exécutent en tant que service unique avec des processus étroitement couplés. Les applications monolithiques ont plusieurs inconvénients. Si une fonctionnalité de l'application connaît un pic de demande, l'architecture entière doit être mise à l'échelle. L'ajout ou l'amélioration des fonctionnalités d'une application monolithique devient également plus complexe lorsque la base de code s'élargit. Pour résoudre ces problèmes, vous pouvez utiliser une architecture de microservices. Pour plus d'informations, veuillez consulter [Decomposing monoliths into microservices](#).

MPA

Voir [Évaluation du portefeuille de migration](#).

MQTT

Voir [Message Queuing Telemetry](#) Transport.

classification multi-classes

Processus qui permet de générer des prédictions pour plusieurs classes (prédiction d'un résultat parmi plus de deux). Par exemple, un modèle de ML peut demander « Ce produit est-il un livre, une voiture ou un téléphone ? » ou « Quelle catégorie de produits intéresse le plus ce client ? ».

infrastructure mutable

Modèle qui met à jour et modifie l'infrastructure existante pour les charges de travail de production. Pour améliorer la cohérence, la fiabilité et la prévisibilité, le AWS Well-Architected Framework recommande l'utilisation d'une [infrastructure immuable](#) comme meilleure pratique.

O

OAC

Voir [Contrôle d'accès à l'origine](#).

OAI

Voir [l'identité d'accès à l'origine](#).

OCM

Voir [gestion du changement organisationnel](#).

migration hors ligne

Méthode de migration dans laquelle la charge de travail source est supprimée au cours du processus de migration. Cette méthode implique un temps d'arrêt prolongé et est généralement utilisée pour de petites charges de travail non critiques.

OI

Consultez la section [Intégration des opérations](#).

OLA

Voir l'accord [au niveau opérationnel](#).

migration en ligne

Méthode de migration dans laquelle la charge de travail source est copiée sur le système cible sans être mise hors ligne. Les applications connectées à la charge de travail peuvent continuer à fonctionner pendant la migration. Cette méthode implique un temps d'arrêt nul ou minimal et est généralement utilisée pour les charges de travail de production critiques.

OPC-UA

Voir [Open Process Communications - Architecture unifiée](#).

Communications par processus ouvert - Architecture unifiée (OPC-UA)

Protocole de communication machine à machine (M2M) pour l'automatisation industrielle. OPC-UA fournit une norme d'interopérabilité avec des schémas de chiffrement, d'authentification et d'autorisation des données.

accord au niveau opérationnel (OLA)

Accord qui précise ce que les groupes informatiques fonctionnels s'engagent à fournir les uns aux autres, afin de prendre en charge un contrat de niveau de service (SLA).

examen de l'état de préparation opérationnelle (ORR)

Une liste de questions et de bonnes pratiques associées qui vous aident à comprendre, à évaluer, à prévenir ou à réduire l'ampleur des incidents et des défaillances possibles. Pour plus d'informations, voir [Examens de l'état de préparation opérationnelle \(ORR\)](#) dans le AWS Well-Architected cadre.

technologie opérationnelle (OT)

Systèmes matériels et logiciels qui fonctionnent avec l'environnement physique pour contrôler les opérations, les équipements et les infrastructures industriels. Dans le secteur manufacturier, l'intégration des systèmes OT et des technologies de l'information (IT) est au cœur des transformations de [l'industrie 4.0](#).

intégration des opérations (OI)

Processus de modernisation des opérations dans le cloud, qui implique la planification de la préparation, l'automatisation et l'intégration. Pour en savoir plus, veuillez consulter le [guide d'intégration des opérations](#).

journal de suivi d'organisation

Un parcours créé par AWS CloudTrail qui enregistre tous les événements pour tous les membres Comptes AWS d'une organisation dans AWS Organizations. Ce journal de suivi est créé dans chaque Compte AWS qui fait partie de l'organisation et suit l'activité de chaque compte. Pour plus d'informations, consultez [la section Création d'un suivi pour une organisation](#) dans la CloudTrail documentation.

gestion du changement organisationnel (OCM)

Cadre pour gérer les transformations métier majeures et perturbatrices du point de vue des personnes, de la culture et du leadership. L'OCM aide les organisations à se préparer et à effectuer la transition vers de nouveaux systèmes et de nouvelles politiques en accélérant l'adoption des changements, en abordant les problèmes de transition et en favorisant des changements culturels et organisationnels. Dans la stratégie de AWS migration, ce cadre est appelé accélération du personnel, en raison de la rapidité du changement requise dans les projets d'adoption du cloud. Pour plus d'informations, veuillez consulter le [guide OCM](#).

contrôle d'accès d'origine (OAC)

Dans CloudFront, une option améliorée pour restreindre l'accès afin de sécuriser votre contenu Amazon Simple Storage Service (Amazon S3). OAC prend en charge tous les compartiments S3 dans leur ensemble Régions AWS, le chiffrement côté serveur avec AWS KMS (SSE-KMS) et les DELETE requêtes dynamiques PUT adressées au compartiment S3.

identité d'accès d'origine (OAI)

Dans CloudFront, une option permettant de restreindre l'accès afin de sécuriser votre contenu Amazon S3. Lorsque vous utilisez OAI, il CloudFront crée un principal auprès duquel Amazon S3 peut s'authentifier. Les principaux authentifiés ne peuvent accéder au contenu d'un compartiment S3 que par le biais d'une distribution spécifique CloudFront . Voir également [OAC](#), qui fournit un contrôle d'accès plus précis et amélioré.

ORR

Voir l'[examen de l'état de préparation opérationnelle](#).

DE

Voir [technologie opérationnelle](#).

VPC sortant (de sortie)

Dans une architecture AWS multi-comptes, un VPC qui gère les connexions réseau initiées depuis une application. L'[architecture de référence de sécuritéAWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et Internet en général.

P

limite des autorisations

Politique de gestion IAM attachée aux principaux IAM pour définir les autorisations maximales que peut avoir l'utilisateur ou le rôle. Pour plus d'informations, veuillez consulter la rubrique [Limites des autorisations](#) dans la documentation IAM.

informations personnelles identifiables (PII)

Informations qui, lorsqu'elles sont consultées directement ou associées à d'autres données connexes, peuvent être utilisées pour déduire raisonnablement l'identité d'une personne. Les

exemples d'informations personnelles incluent les noms, les adresses et les informations de contact.

PII

Voir les [informations personnelles identifiables](#).

manuel stratégique

Ensemble d'étapes prédéfinies qui capturent le travail associé aux migrations, comme la fourniture de fonctions d'opérations de base dans le cloud. Un manuel stratégique peut revêtir la forme de scripts, de runbooks automatisés ou d'un résumé des processus ou des étapes nécessaires au fonctionnement de votre environnement modernisé.

PLC

Voir [contrôleur logique programmable](#).

PLM

Consultez la section [Gestion du cycle de vie des](#) produits.

policy

Objet capable de définir les autorisations (voir la [politique basée sur l'identité](#)), de spécifier les conditions d'accès (voir la [politique basée sur les ressources](#)) ou de définir les autorisations maximales pour tous les comptes d'une organisation dans AWS Organizations (voir la politique de contrôle des [services](#)).

persistance polyglotte

Choix indépendant de la technologie de stockage de données d'un microservice en fonction des modèles d'accès aux données et d'autres exigences. Si vos microservices utilisent la même technologie de stockage de données, ils peuvent rencontrer des difficultés d'implémentation ou présenter des performances médiocres. Les microservices sont plus faciles à mettre en œuvre, atteignent de meilleures performances, ainsi qu'une meilleure capacité de mise à l'échelle s'ils utilisent l'entrepôt de données le mieux adapté à leurs besoins.

évaluation du portefeuille

Processus de découverte, d'analyse et de priorisation du portefeuille d'applications afin de planifier la migration. Pour plus d'informations, veuillez consulter [Evaluating migration readiness](#).

predicate

Une condition de requête qui renvoie `true` ou `false`, généralement située dans une `WHERE` clause.

prédicat pushdown

Technique d'optimisation des requêtes de base de données qui filtre les données de la requête avant le transfert. Cela réduit la quantité de données qui doivent être extraites et traitées à partir de la base de données relationnelle et améliore les performances des requêtes.

contrôle préventif

Contrôle de sécurité conçu pour empêcher qu'un événement ne se produise. Ces contrôles constituent une première ligne de défense pour empêcher tout accès non autorisé ou toute modification indésirable de votre réseau. Pour plus d'informations, veuillez consulter [Preventative controls](#) dans Implementing security controls on AWS.

principal

Entité capable d'effectuer AWS des actions et d'accéder à des ressources. Cette entité est généralement un utilisateur root pour un Compte AWS rôle IAM ou un utilisateur. Pour plus d'informations, veuillez consulter la rubrique Principal dans [Termes et concepts relatifs aux rôles](#), dans la documentation IAM.

confidentialité dès la conception

Une approche d'ingénierie système qui prend en compte la confidentialité tout au long du processus de développement.

zones hébergées privées

Conteneur qui contient des informations concernant la façon dont vous souhaitez qu'Amazon Route 53 réponde aux requêtes DNS pour un domaine et ses sous-domaines dans un ou plusieurs VPC. Pour plus d'informations, veuillez consulter [Working with private hosted zones](#) dans la documentation Route 53.

contrôle proactif

[Contrôle de sécurité](#) conçu pour empêcher le déploiement de ressources non conformes. Ces contrôles analysent les ressources avant qu'elles ne soient provisionnées. Si la ressource n'est pas conforme au contrôle, elle n'est pas provisionnée. Pour plus d'informations, consultez le [guide de référence sur les contrôles](#) dans la AWS Control Tower documentation et consultez la section [Contrôles proactifs dans Implémentation](#) des contrôles de sécurité sur AWS.

gestion du cycle de vie des produits (PLM)

Gestion des données et des processus d'un produit tout au long de son cycle de vie, depuis la conception, le développement et le lancement, en passant par la croissance et la maturité, jusqu'au déclin et au retrait.

environnement de production

Voir [environnement](#).

contrôleur logique programmable (PLC)

Dans le secteur manufacturier, un ordinateur hautement fiable et adaptable qui surveille les machines et automatise les processus de fabrication.

chaînage rapide

Utiliser le résultat d'une invite [LLM](#) comme entrée pour l'invite suivante afin de générer de meilleures réponses. Cette technique est utilisée pour décomposer une tâche complexe en sous-tâches ou pour affiner ou développer de manière itérative une réponse préliminaire. Cela permet d'améliorer la précision et la pertinence des réponses d'un modèle et permet d'obtenir des résultats plus précis et personnalisés.

pseudonymisation

Processus de remplacement des identifiants personnels dans un ensemble de données par des valeurs fictives. La pseudonymisation peut contribuer à protéger la vie privée. Les données pseudonymisées sont toujours considérées comme des données personnelles.

publish/subscribe (pub/sub)

Modèle qui permet des communications asynchrones entre les microservices afin d'améliorer l'évolutivité et la réactivité. Par exemple, dans un [MES](#) basé sur des microservices, un microservice peut publier des messages d'événements sur un canal auquel d'autres microservices peuvent s'abonner. Le système peut ajouter de nouveaux microservices sans modifier le service de publication.

Q

plan de requête

Série d'étapes, telles que des instructions, utilisées pour accéder aux données d'un système de base de données relationnelle SQL.

régression du plan de requêtes

Le cas où un optimiseur de service de base de données choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données. Cela peut être dû à des

changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à jour du moteur de base de données.

R

Matrice RACI

Voir [responsable, responsable, consulté, informé \(RACI\)](#).

RAG

Voir [Retrieval Augmented Generation](#).

rançongiciel

Logiciel malveillant conçu pour bloquer l'accès à un système informatique ou à des données jusqu'à ce qu'un paiement soit effectué.

Matrice RASCI

Voir [responsable, responsable, consulté, informé \(RACI\)](#).

RCAC

Voir [contrôle d'accès aux lignes et aux colonnes](#).

réplica en lecture

Copie d'une base de données utilisée en lecture seule. Vous pouvez acheminer les requêtes vers le réplica de lecture pour réduire la charge sur votre base de données principale.

réarchitecte

Voir [7 Rs](#).

objectif de point de récupération (RPO)

Durée maximale acceptable depuis le dernier point de récupération des données. Il détermine ce qui est considéré comme étant une perte de données acceptable entre le dernier point de reprise et l'interruption du service.

objectif de temps de récupération (RTO)

Le délai maximum acceptable entre l'interruption du service et le rétablissement du service.

refactoriser

Voir [7 Rs](#).

Région

Un ensemble de AWS ressources dans une zone géographique. Chacune Région AWS est isolée et indépendante des autres pour garantir tolérance aux pannes, stabilité et résilience. Pour plus d'informations, voir [Spécifier ce que Régions AWS votre compte peut utiliser](#).

régression

Technique de ML qui prédit une valeur numérique. Par exemple, pour résoudre le problème « Quel sera le prix de vente de cette maison ? », un modèle de ML pourrait utiliser un modèle de régression linéaire pour prédire le prix de vente d'une maison sur la base de faits connus à son sujet (par exemple, la superficie en mètres carrés).

réhéberger

Voir [7 Rs](#).

version

Dans un processus de déploiement, action visant à promouvoir les modifications apportées à un environnement de production.

déplacer

Voir [7 Rs](#).

replateforme

Voir [7 Rs](#).

rachat

Voir [7 Rs](#).

résilience

La capacité d'une application à résister aux perturbations ou à s'en remettre. [La haute disponibilité et la reprise après sinistre](#) sont des considérations courantes lors de la planification de la résilience dans le AWS Cloud. Pour plus d'informations, consultez la section [AWS Cloud Résilience](#).

politique basée sur les ressources

Politique attachée à une ressource, comme un compartiment Amazon S3, un point de terminaison ou une clé de chiffrement. Ce type de politique précise les principaux auxquels l'accès est autorisé, les actions prises en charge et toutes les autres conditions qui doivent être remplies.

matrice responsable, redevable, consulté et informé (RACI)

Une matrice qui définit les rôles et les responsabilités de toutes les parties impliquées dans les activités de migration et les opérations cloud. Le nom de la matrice est dérivé des types de responsabilité définis dans la matrice : responsable (R), responsable (A), consulté (C) et informé (I). Le type de support (S) est facultatif. Si vous incluez le support, la matrice est appelée matrice RASCI, et si vous l'excluez, elle est appelée matrice RACI.

contrôle réactif

Contrôle de sécurité conçu pour permettre de remédier aux événements indésirables ou aux écarts par rapport à votre référence de sécurité. Pour plus d'informations, veuillez consulter la rubrique [Responsive controls](#) dans Implementing security controls on AWS.

retain

Voir [7 Rs](#).

se retirer

Voir [7 Rs](#).

Génération augmentée de récupération (RAG)

Technologie d'[IA générative](#) dans laquelle un [LLM](#) fait référence à une source de données faisant autorité qui se trouve en dehors de ses sources de données de formation avant de générer une réponse. Par exemple, un modèle RAG peut effectuer une recherche sémantique dans la base de connaissances ou dans les données personnalisées d'une organisation. Pour plus d'informations, voir [Qu'est-ce que RAG ?](#)

rotation

Processus de mise à jour périodique d'un [secret](#) pour empêcher un attaquant d'accéder aux informations d'identification.

contrôle d'accès aux lignes et aux colonnes (RCAC)

Utilisation d'expressions SQL simples et flexibles dotées de règles d'accès définies. Le RCAC comprend des autorisations de ligne et des masques de colonnes.

RPO

Voir l'[objectif du point de récupération](#).

RTO

Voir l'[objectif en matière de temps de rétablissement](#).

runbook

Ensemble de procédures manuelles ou automatisées nécessaires à l'exécution d'une tâche spécifique. Elles visent généralement à rationaliser les opérations ou les procédures répétitives présentant des taux d'erreur élevés.

S

SAML 2.0

Un standard ouvert utilisé par de nombreux fournisseurs d'identité (IdPs). Cette fonctionnalité permet l'authentification unique fédérée (SSO), afin que les utilisateurs puissent se connecter AWS Management Console ou appeler les opérations de l' AWS API sans que vous ayez à créer un utilisateur dans IAM pour tous les membres de votre organisation. Pour plus d'informations sur la fédération SAML 2.0, veuillez consulter [À propos de la fédération SAML 2.0](#) dans la documentation IAM.

SCADA

Voir [Contrôle de supervision et acquisition de données](#).

SCP

Voir la [politique de contrôle des services](#).

secret

Dans AWS Secrets Manager des informations confidentielles ou restreintes, telles qu'un mot de passe ou des informations d'identification utilisateur, que vous stockez sous forme cryptée. Il comprend la valeur secrète et ses métadonnées. La valeur secrète peut être binaire, une chaîne unique ou plusieurs chaînes. Pour plus d'informations, voir [Que contient le secret d'un Secrets Manager ?](#) dans la documentation de Secrets Manager.

sécurité dès la conception

Une approche d'ingénierie système qui prend en compte la sécurité tout au long du processus de développement.

contrôle de sécurité

Barrière de protection technique ou administrative qui empêche, détecte ou réduit la capacité d'un assaillant d'exploiter une vulnérabilité de sécurité. Il existe quatre principaux types de contrôles de sécurité : [préventifs](#), [détectifs](#), [réactifs](#) et [proactifs](#).

renforcement de la sécurité

Processus qui consiste à réduire la surface d'attaque pour la rendre plus résistante aux attaques. Cela peut inclure des actions telles que la suppression de ressources qui ne sont plus requises, la mise en œuvre des bonnes pratiques de sécurité consistant à accorder le moindre privilège ou la désactivation de fonctionnalités inutiles dans les fichiers de configuration.

système de gestion des informations et des événements de sécurité (SIEM)

Outils et services qui associent les systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un système SIEM collecte, surveille et analyse les données provenant de serveurs, de réseaux, d'appareils et d'autres sources afin de détecter les menaces et les failles de sécurité, mais aussi de générer des alertes.

automatisation des réponses de sécurité

Action prédéfinie et programmée conçue pour répondre automatiquement à un événement de sécurité ou y remédier. Ces automatisations servent de contrôles de sécurité [détectifs ou réactifs](#) qui vous aident à mettre en œuvre les meilleures pratiques en matière AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une instance Amazon EC2 ou la rotation des informations d'identification.

chiffrement côté serveur

Chiffrement des données à destination, par celui Service AWS qui les reçoit.

Politique de contrôle des services (SCP)

Politique qui propose un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. Les SCP définissent des barrières de protection ou des limites aux actions qu'un administrateur peut déléguer à des utilisateurs ou à des rôles. Vous pouvez utiliser les SCP comme listes d'autorisation ou de refus, pour indiquer les services ou les actions autorisés ou interdits. Pour plus d'informations, consultez la section [Politiques de contrôle des services](#) dans la AWS Organizations documentation.

point de terminaison du service

URL du point d'entrée pour un Service AWS. Pour vous connecter par programmation au service cible, vous pouvez utiliser un point de terminaison. Pour plus d'informations, veuillez consulter la rubrique [Service AWS endpoints](#) dans Références générales AWS.

contrat de niveau de service (SLA)

Accord qui précise ce qu'une équipe informatique promet de fournir à ses clients, comme le temps de disponibilité et les performances des services.

indicateur de niveau de service (SLI)

Mesure d'un aspect des performances d'un service, tel que son taux d'erreur, sa disponibilité ou son débit.

objectif de niveau de service (SLO)

Mesure cible qui représente l'état d'un service, tel que mesuré par un indicateur de [niveau de service](#).

modèle de responsabilité partagée

Un modèle décrivant la responsabilité que vous partagez en matière AWS de sécurité et de conformité dans le cloud. AWS est responsable de la sécurité du cloud, alors que vous êtes responsable de la sécurité dans le cloud. Pour de plus amples informations, veuillez consulter [Modèle de responsabilité partagée](#).

IA de l'ombre

Applications d'[IA](#) non autorisées créées ou utilisées en dehors des canaux régis au sein d'une organisation.

SIEM

Consultez les [informations de sécurité et le système de gestion des événements](#).

point de défaillance unique (SPOF)

Défaillance d'un seul composant critique d'une application susceptible de perturber le système.

SLA

Voir le contrat [de niveau de service](#).

SLI

Voir l'indicateur de [niveau de service](#).

SLO

Voir l'objectif de [niveau de service](#).

modèle split-and-seed

Modèle permettant de mettre à l'échelle et d'accélérer les projets de modernisation. Au fur et à mesure que les nouvelles fonctionnalités et les nouvelles versions de produits sont définies, l'équipe principale se divise pour créer des équipes de produit. Cela permet de mettre à l'échelle

les capacités et les services de votre organisation, d'améliorer la productivité des développeurs et de favoriser une innovation rapide. Pour plus d'informations, consultez la section [Approche progressive de la modernisation des applications dans](#) le. AWS Cloud

SPOF

Voir [point de défaillance unique](#).

schéma en étoile

Structure organisationnelle de base de données qui utilise une grande table de faits pour stocker les données transactionnelles ou mesurées et utilise une ou plusieurs tables dimensionnelles plus petites pour stocker les attributs des données. Cette structure est conçue pour être utilisée dans un [entrepôt de données](#) ou à des fins de business intelligence.

modèle de figuier étrangleur

Approche de modernisation des systèmes monolithiques en réécrivant et en remplaçant progressivement les fonctionnalités du système jusqu'à ce que le système hérité puisse être mis hors service. Ce modèle utilise l'analogie d'un figuier de vigne qui se développe dans un arbre existant et qui finit par supplanter son hôte. Le schéma a été [présenté par Martin Fowler](#) comme un moyen de gérer les risques lors de la réécriture de systèmes monolithiques. Pour un exemple d'application de ce modèle, consultez la section [Modernisation progressive des anciens services Web Microsoft ASP.NET \(ASMX\) à l'aide de conteneurs et d'Amazon API Gateway](#).

sous-réseau

Plage d'adresses IP dans votre VPC. Un sous-réseau doit se trouver dans une seule zone de disponibilité.

contrôle de supervision et acquisition de données (SCADA)

Dans le secteur manufacturier, un système qui utilise du matériel et des logiciels pour surveiller les actifs physiques et les opérations de production.

chiffrement symétrique

Algorithme de chiffrement qui utilise la même clé pour chiffrer et déchiffrer les données.

tests synthétiques

Tester un système de manière à simuler les interactions des utilisateurs afin de détecter les problèmes potentiels ou de surveiller les performances. Vous pouvez utiliser [Amazon CloudWatch Synthetics](#) pour créer ces tests.

invite du système

Technique permettant de fournir un contexte, des instructions ou des directives à un [LLM](#) afin d'orienter son comportement. Les instructions du système aident à définir le contexte et à établir des règles pour les interactions avec les utilisateurs.

T

tags

Key-value des paires qui agissent comme des métadonnées pour organiser vos AWS ressources. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources. Pour plus d'informations, veuillez consulter la rubrique [Balisage de vos AWS ressources](#).

variable cible

La valeur que vous essayez de prédire dans le cadre du ML supervisé. Elle est également qualifiée de variable de résultat. Par exemple, dans un environnement de fabrication, la variable cible peut être un défaut du produit.

liste de tâches

Outil utilisé pour suivre les progrès dans un runbook. Liste de tâches qui contient une vue d'ensemble du runbook et une liste des tâches générales à effectuer. Pour chaque tâche générale, elle inclut le temps estimé nécessaire, le propriétaire et l'avancement.

environnement de test

Voir [environnement](#).

entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

outil

Fonction ou API qu'un [agent](#) peut invoquer pour effectuer des opérations dans des systèmes externes.

passerelle de transit

Hub de transit de réseau que vous pouvez utiliser pour relier vos VPC et vos réseaux sur site. Pour plus d'informations, voir [Qu'est-ce qu'une passerelle de transit](#) dans la AWS Transit Gateway documentation.

flux de travail basé sur jonction

Approche selon laquelle les développeurs génèrent et testent des fonctionnalités localement dans une branche de fonctionnalités, puis fusionnent ces modifications dans la branche principale. La branche principale est ensuite intégrée aux environnements de développement, de préproduction et de production, de manière séquentielle.

accès sécurisé

Accorder des autorisations à un service que vous spécifiez pour effectuer des tâches au sein de votre organisation AWS Organizations et dans ses comptes en votre nom. Le service de confiance crée un rôle lié au service dans chaque compte, lorsque ce rôle est nécessaire, pour effectuer des tâches de gestion à votre place. Pour plus d'informations, consultez la section [Utilisation AWS Organizations avec d'autres AWS services](#) dans la AWS Organizations documentation.

réglage

Pour modifier certains aspects de votre processus d'entraînement afin d'améliorer la précision du modèle de ML. Par exemple, vous pouvez entraîner le modèle de ML en générant un ensemble d'étiquetage, en ajoutant des étiquettes, puis en répétant ces étapes plusieurs fois avec différents paramètres pour optimiser le modèle.

équipe de deux pizzas

Une petite DevOps équipe que vous pouvez nourrir avec deux pizzas. Une équipe de deux pizzas garantit les meilleures opportunités de collaboration possible dans le développement de logiciels.

U

incertitude

Un concept qui fait référence à des informations imprécises, incomplètes ou inconnues susceptibles de compromettre la fiabilité des modèles de ML prédictifs. Il existe deux types d'incertitude : l'incertitude épistémique est causée par des données limitées et incomplètes, alors que l'incertitude aléatoire est causée par le bruit et le caractère aléatoire inhérents aux données.

tâches indifférenciées

Également connu sous le nom de « levage de charges lourdes », ce travail est nécessaire pour créer et exploiter une application, mais qui n'apporte pas de valeur directe à l'utilisateur final ni d'avantage concurrentiel. Les exemples de tâches indifférenciées incluent l'approvisionnement, la maintenance et la planification des capacités.

environnements supérieurs

Voir [environnement](#).

V

mise à vide

Opération de maintenance de base de données qui implique un nettoyage après des mises à jour incrémentielles afin de récupérer de l'espace de stockage et d'améliorer les performances.

contrôle de version

Processus et outils permettant de suivre les modifications, telles que les modifications apportées au code source dans un référentiel.

Appairage de VPC

Connexion entre deux VPC qui vous permet d'acheminer le trafic à l'aide d'adresses IP privées. Pour plus d'informations, veuillez consulter la rubrique [Qu'est-ce que l'appairage de VPC ?](#) dans la documentation Amazon VPC.

vulnérabilités

Défaut logiciel ou matériel qui compromet la sécurité du système.

W

cache actif

Cache tampon qui contient les données actuelles et pertinentes fréquemment consultées. L'instance de base de données peut lire à partir du cache tampon, ce qui est plus rapide que la lecture à partir de la mémoire principale ou du disque.

données chaudes

Données rarement consultées. Lorsque vous interrogez ce type de données, des requêtes modérément lentes sont généralement acceptables.

fonction de fenêtre

Fonction SQL qui effectue un calcul sur un groupe de lignes liées d'une manière ou d'une autre à l'enregistrement en cours. Les fonctions de fenêtre sont utiles pour traiter des tâches, telles que le calcul d'une moyenne mobile ou l'accès à la valeur des lignes en fonction de la position relative de la ligne en cours.

charge de travail

Ensemble de ressources et de code qui fournit une valeur métier, par exemple une application destinée au client ou un processus de backend.

flux de travail

Groupes fonctionnels d'un projet de migration chargés d'un ensemble de tâches spécifique. Chaque flux de travail est indépendant, mais prend en charge les autres flux de travail du projet. Par exemple, le flux de travail du portefeuille est chargé de prioriser les applications, de planifier les vagues et de collecter les métadonnées de migration. Le flux de travail du portefeuille fournit ces actifs au flux de travail de migration, qui migre ensuite les serveurs et les applications.

VER

Voir [écrire une fois, lire plusieurs](#).

WQF

Voir le [cadre AWS de qualification de la charge](#) de travail.

écrire une fois, lire plusieurs (WORM)

Modèle de stockage qui écrit les données une seule fois et empêche leur suppression ou leur modification. Les utilisateurs autorisés peuvent lire les données autant de fois que nécessaire, mais ils ne peuvent pas les modifier. Cette infrastructure de stockage de données est considérée comme [immuable](#).

Z

exploit Zero-Day

Une attaque, généralement un logiciel malveillant, qui tire parti d'une [vulnérabilité de type « jour zéro »](#).

vulnérabilité de type « jour zéro »

Une faille ou une vulnérabilité non atténuée dans un système de production. Les acteurs malveillants peuvent utiliser ce type de vulnérabilité pour attaquer le système. Les développeurs prennent souvent conscience de la vulnérabilité à la suite de l'attaque.

invite Zero-Shot

Fournir à un [LLM](#) des instructions pour effectuer une tâche, mais aucun exemple (plans) pouvant aider à la guider. Le LLM doit utiliser ses connaissances pré-entraînées pour gérer la tâche. L'efficacité de l'invite zéro dépend de la complexité de la tâche et de la qualité de l'invite. Voir également les instructions [en quelques clics](#).

application zombie

Application dont l'utilisation moyenne du processeur et de la mémoire est inférieure à 5 %. Dans un projet de migration, il est courant de retirer ces applications.

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.