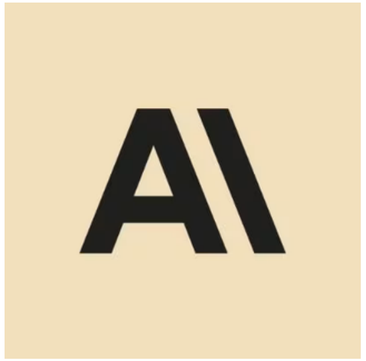

MCP
(Model Context Protocol)

SOMMAIRE

1/Brève histoire du MCP



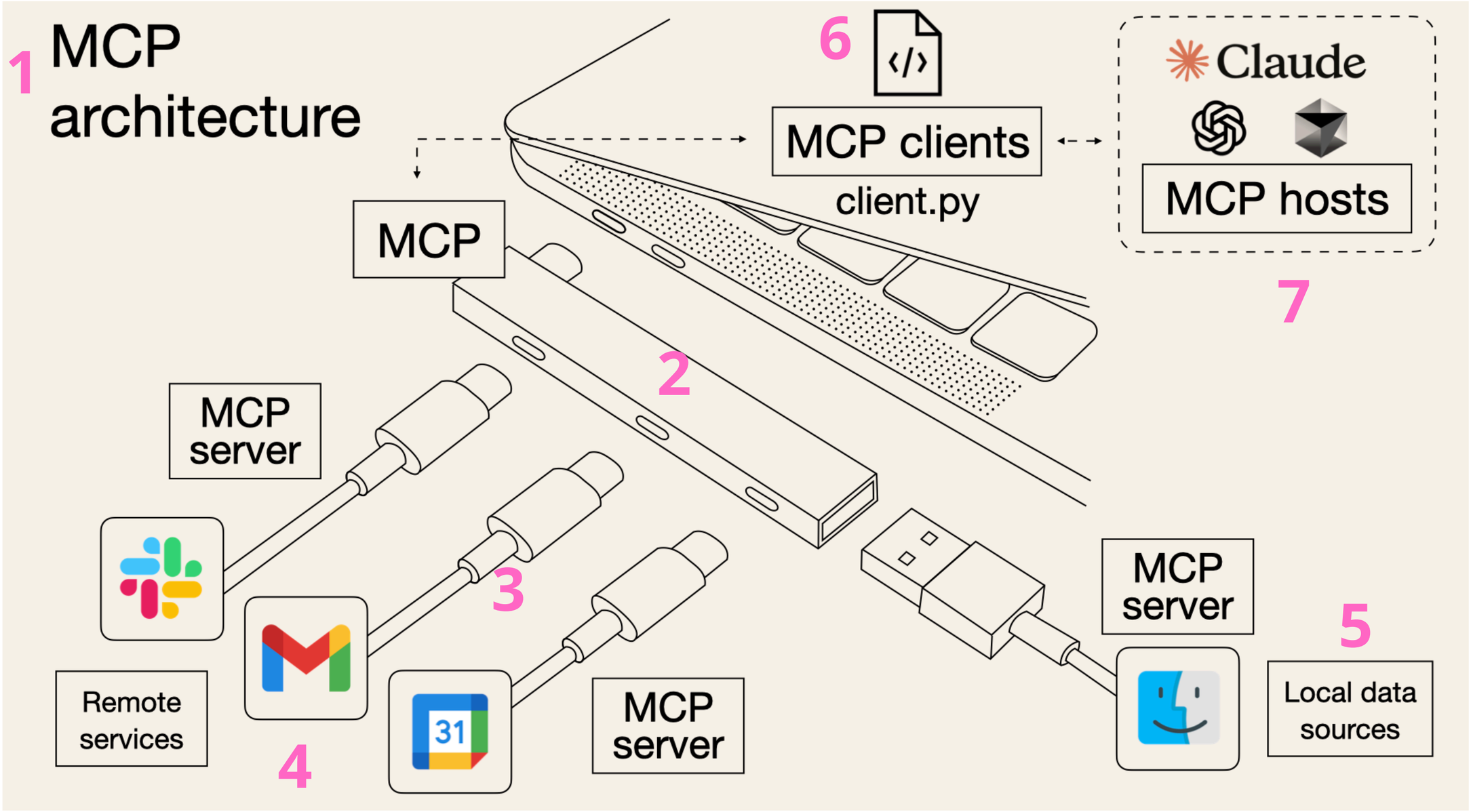
- En novembre 2024, Anthropic, le créateur du modèle de langage (LLM) Claude, a publié en open source le **Model Context Protocol** (MCP).

2/Qu’est ce qu’un server MCP ?

- MCP (Model Context Protocol) est un protocole ouvert et universel qui standardise la manière dont les applications fournissent des informations contextuelles aux modèles de langage de grande taille (LLMs)

- Tout comme le protocole HTTP permet à différents sites web et navigateurs d'échanger des informations selon les mêmes règles, **le MCP est comme le protocole HTTP du monde de l'IA**.

3/Architecture hybride



1.MCP Architecture:
C'est le concept général présenté dans l'image, montrant une manière d'organiser l'accès à différentes sources de contenu

2.MCP (Hub Central):
Représenté par la barre au milieu, c'est le cœur du système. Il fonctionne comme un hub central auquel se connectent les sources de données et les applications, standardisant l'échange de données

3.MCP Server:
Ce sont des adaptateurs spécifiques à chaque source de données (Slack, Gmail, calendrier, fichiers locaux). Chaque serveur communique avec sa source de données et traduit les informations pour le hub MCP / Plusieurs sont connectables au hub central de façon simultanée

4.Remote Services (Services Distants):
Ce sont les sources de données externes comme Slack, Gmail, ou Google Calendar.
Chacune utilise un MCP Server dédié pour se connecter au hub MCP

5.Local Data Sources:
Ce sont des sources de données présentes sur la machine locale de l'utilisateur, comme le système de fichiers (représenté par l'icône du Finder macOS).

6.MCP Clients (client.py):
Ce sont les applications où elles ont besoin d'accéder aux données.
Au lieu de se connecter directement à chaque source (Slack, Gmail, fichiers locaux, etc.), le client se connecte uniquement au hub MCP central.

7.MCP Hosts:
Ce sont les LLM, comme Claude. Ces LLM récupèrent les données par les MCP Clients pour donner des réponses intelligentes

4/ Rapport d’analyse des MCP leaders adaptées aux environnements professionnels + API/SECURITE/ISOLATION

Plateformes de Graphes de Connaissances d'Entreprise (EKG)

> *Ces plateformes connectent des données diverses et permettent une interrogation sémantique via API, alignées avec l'aspect "cognitif" et "multi-sources".*

Stardog

Pertinence : Plateforme EKG mature, focus sur sémantique, inférence, intégration de données virtuelles.

Intégration : API (SPARQL, GraphQL), connecteurs variés, orientée entreprise.

Neo4j

Pertinence : Leader des bases de données graphe, souvent utilisée comme fondation EKG.

Intégration : API (Cypher via Bolt, GraphQL), outils d'intégration, large écosystème.

Plateformes d'Intégration / Virtualisation / Data Fabric

> *Excellent pour connecter des sources hétérogènes via une couche unifiée avec gouvernance, mais potentiellement moins "cognitif" nativement.*

Denodo

Pertinence : Leader en virtualisation de données et data fabric.

Intégration : Accès unifié à de nombreuses sources via API (SQL, REST, OData), forte gouvernance.

Informatica (IDMC)

Pertinence : Plateforme complète (intégration, qualité, gouvernance).

Intégration : Connecteurs très étendus, APIs, focus entreprise.

Plateformes d'Intelligence et Recherche Cognitive

> *Conçues pour la recherche intelligente sur de grands corpus multi-sources, intégrant des capacités IA.*

Sinequa / Coveo

Pertinence : Leaders en recherche d'entreprise et pertinence IA.

Intégration : API de recherche avancée, connecteurs, peuvent agir comme un RAG externe ou une source pour l'orchestrateur.

AWS Kendra / Azure Cognitive Search / Google Vertex AI Search

Pertinence : Services cloud managés intégrant recherche sémantique et IA.

Intégration : API spécifiques à chaque cloud, bonne intégration avec leurs écosystèmes respectifs, options de sécurité cloud.

1. Stardog (Base de Graphe Sémantique)

- **API**

Fournit une API web standard (HTTP) pour interroger les données via les langages de graphe (SPARQL, GraphQL) et des bibliothèques pour intégration facile dans diverses applications.

- **Contrôles de sécurité**

Gère l'accès via authentification multiple (dont Kerberos) et contrôle précisément les droits utilisateurs (RBAC) sur les bases et les éléments du graphe; sécurise les connexions (TLS).

- **Mécanismes d'isolation**

Permet de séparer les accès logiques entre différentes bases de données ou graphes via les rôles, et sécurise l'accès aux données externes via les graphes virtuels.

2. Neo4j (Base de Graphe Native)

- **API**

Offre un protocole binaire optimisé (Bolt) pour les requêtes Cypher via drivers, une API web (HTTP) et une interface GraphQL pour la flexibilité d'intégration.

- **Contrôles de sécurité**

Propose authentification variée, contrôle d'accès par rôles (RBAC) extensible à des règles très fines (propriétés) en version Enterprise, et chiffre les données en transit (TLS) et au repos (Enterprise).

- **Mécanismes d'isolation**

Isole les données via les rôles utilisateurs et les permissions très fines sur les éléments du graphe (Enterprise), ainsi que par base de données distincte.

3. Denodo Platform (Virtualisation de Données)

- **API**

Permet l'accès unifié aux données via SQL standard (JDBC/ODBC) et expose facilement ces données comme des services web modernes (REST, OData, GraphQL).

- **Contrôles de sécurité**

S'intègre aux systèmes d'authentification d'entreprise (SAML, OAuth2...), offre un contrôle d'accès très précis (jusqu'à la ligne/colonne), masque les données sensibles et peut déléguer l'authentification aux sources. Sécurise toutes les connexions (TLS).

- **Mécanismes d'isolation**

Cloisonne l'accès aux données de manière très fine grâce aux rôles, aux règles sur les lignes/colonnes, au masquage, et peut appliquer des politiques de sécurité globales basées sur les métadonnées.

4. Sinequa (Recherche & Analyse Intelligente)

- **API**

Expose principalement une API REST pour lancer des recherches et récupérer des résultats structurés, complétée par des outils pour construire rapidement des interfaces de recherche.

- **Contrôles de sécurité**

S'appuie sur l'authentification unique de l'entreprise (SSO) et garantit que les résultats de recherche respectent les permissions des systèmes sources en filtrant selon les ACLs indexées. Sécurise les flux via TLS.

- **Mécanismes d'isolation**

Isole les informations en s'assurant que chaque utilisateur ne voit que les documents autorisés dans les sources, permet de créer des index séparés (collections) et gère les droits d'administration via RBAC.

5/Technique RAG

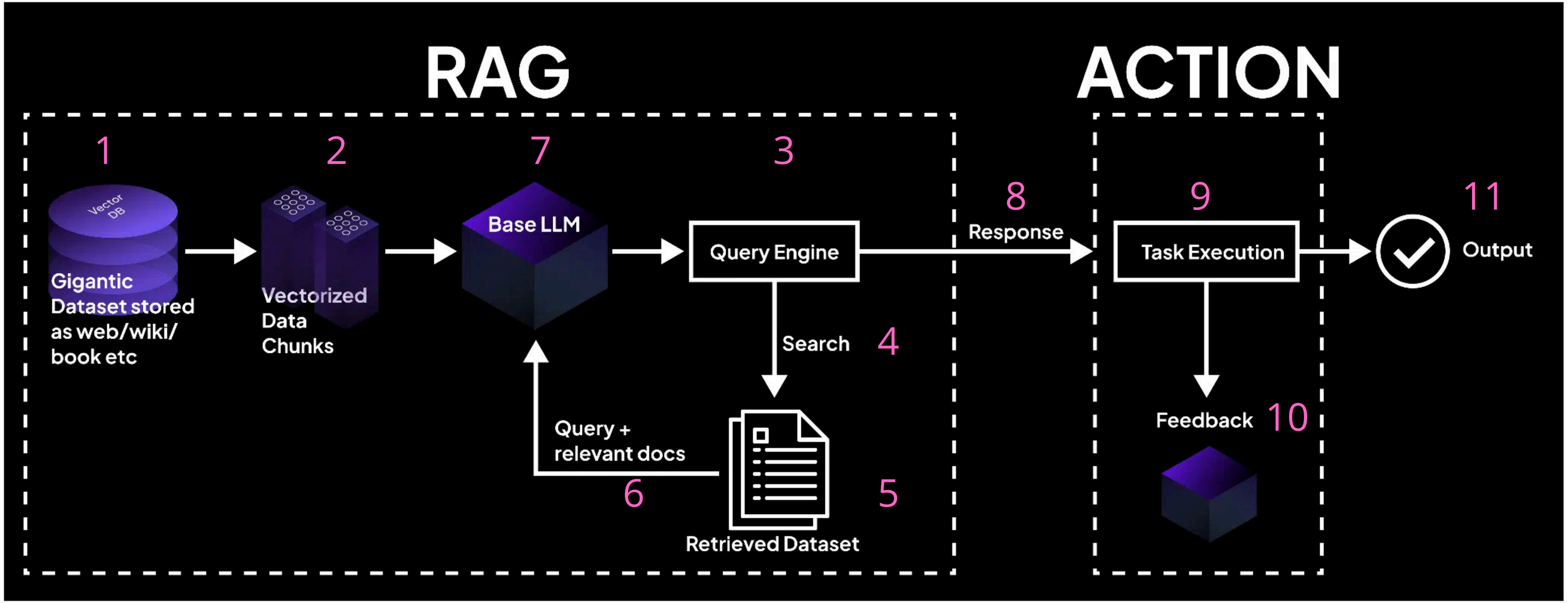
La **RAG (Retrieval-Augmented Generation)** est une technique pour améliorer les réponses des modèles d'IA (LLM). Au lieu de répondre uniquement "de mémoire", le système RAG :

1. Cherche (Retrieval)**:** Trouve d'abord les informations les plus pertinentes concernant la question dans une base de connaissances externe (base de données vectorielle)

2. Augmente (Augmented)**:** Fournit ces informations spécifiques au LLM en même temps que la question originale

3. Génère (Generation): Le LLM utilise ces informations supplémentaires pour produire une réponse beaucoup plus précise, factuelle et adaptée au contexte.

.


RAG
ACTION
1
2
7
3
8
9
11
Vector DB
Gigantic Dataset stored as web/wiki/ book etc
Vectorized Data Chunks
Base LLM
Query Engine
Response
Task Execution
Output
Search
4
Query + relevant docs
6
5
Retrieved Dataset
Feedback
10

RAG

1.Contient toutes les informations en base de données par exemple
2.Ces informations sont découpées en petits bouts et transformées en vecteurs pour être faciles à chercher.
3.Reçoit la question et organise la recherche
4.Cherche dans les fiches codées celles qui correspondent à la question.
5.Récupère les morceaux d'informations pertinents qui ont été trouvés lors de la recherche
6.On donne à l’IA la question originale et les documents trouvés.
7.Réception du LLM qui comprend et génère le texte.

ACTION

8.La réponse textuelle qui vient du RAG.
9.Le composant lit la réponse (ex: envoie un email, met à jour un fichier).
10.Informations “succès ou échec” qui reviennent à l'éxécuteur.
11.Ce qui est produit à la toute fin, après que l'action a été réalisée.

6/ Démo MCP hybride