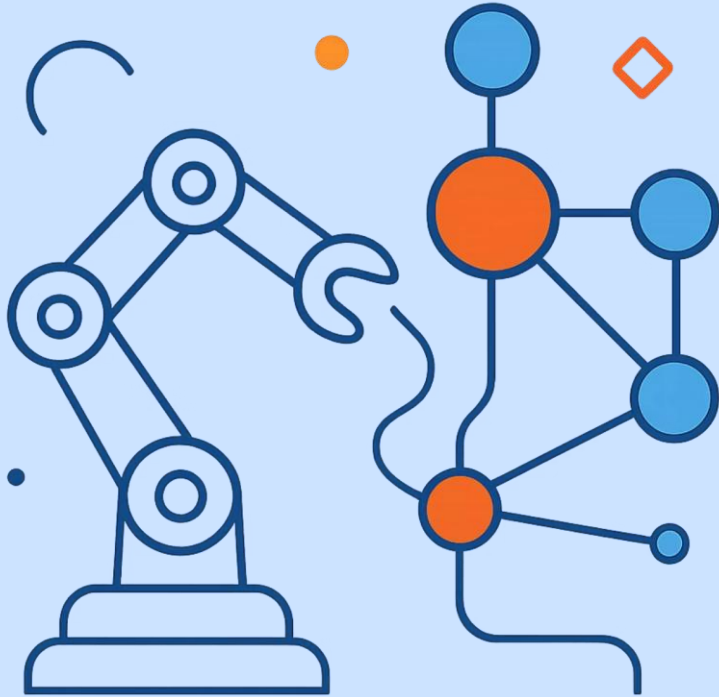




L'Agentic AI

Un changement de paradigme dans l'intelligence artificielle : des systèmes capables de **planifier, agir et itérer** de manière **autonome** pour accomplir des missions complexes sans supervision constante :

LES AGENTS

Qu'est-ce que l'Agentic IA ?

Définition

L'Agentic IA désigne des systèmes IA capables d'agir de manière autonome pour atteindre des objectifs définis. Le terme "agentic" vient du concept d'**agent** : une entité qui perçoit son environnement, prend des décisions et exécute des actions.

Trois caractéristiques clés

Autonomie

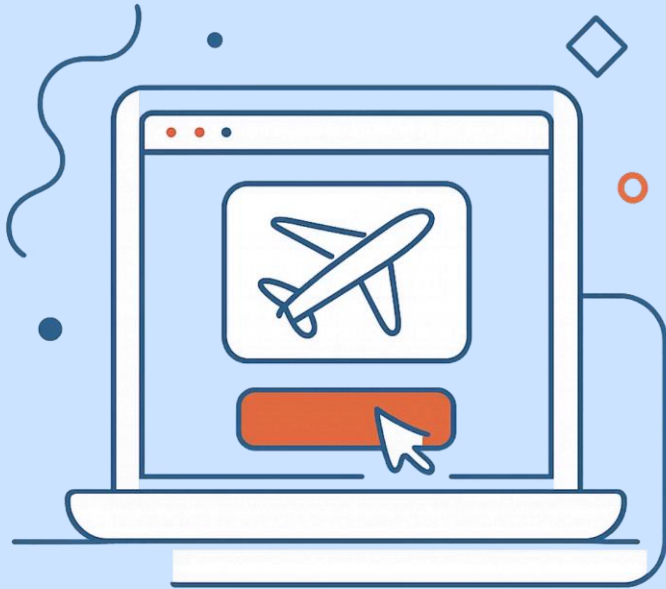
Opérer sans intervention humaine constante

Proactivité

Anticiper et initier des actions

Réactivité

S'adapter aux résultats intermédiaires



Un exemple concret

À la différence d'un LLM classique qui attend une instruction pour générer une réponse unique, un agent IA possède la capacité de décomposer un **objectif complexe** en sous-tâches, de **planifier** l'ordre d'exécution de ces tâches, d'utiliser des **outils externes** et d'ajuster sa stratégie en fonction des résultats obtenus

LLM classique

Si vous demandez à un LLM de "réserver un vol pour Paris la semaine prochaine", il vous fournira des **conseils** ou des **étapes à suivre** — mais n'agira pas.

Agent IA

Un agent IA peut **rechercher les vols disponibles**, comparer les prix, vérifier votre calendrier, et **procéder à la réservation** si vous lui en donnez l'autorisation et les accès nécessaires.

LM vs Agent IA : les différences clés

Un LLM est un outil passif de prédiction de texte. Un agent IA l'utilise comme moteur de raisonnement et l'enrichit de capacités supplémentaires.

Caractéristique	LLM classique	Agent IA
Mode d'interaction	Requête-réponse unique	Boucle d'actions itérative
Utilisation d'outils	Non	Oui (APIs, bases de données, code)
Mémoire	Limitée au contexte	Persistante et structurée
Planification	Absente	Décomposition et séquençage
Autonomie	Passive	Active

Les composants essentiels d'un agent



Modèle de raisonnement

Le LLM au cœur de l'agent : interprète les objectifs, planifie les actions et génère les instructions pour les outils.



Système de planification

Décompose un objectif global en étapes réalisables : planification linéaire, dynamique ou hiérarchique.



Outils (Tools)

Interfaces vers le monde extérieur : APIs, bases de données, exécution de code, systèmes de fichiers.



Mémoire

Court terme (contexte en cours), long terme (base de données), et mémoire de travail (résultats intermédiaires).



Cas d'usage : Données & Processus Métier

Gestion de données & ETL (Extract, Transform, and Load)

Collecte, transformation et chargement de données depuis diverses sources. L'agent s'adapte aux changements de format, gère les erreurs et identifie les anomalies

Exemple : téléchargement CIM 11 pour alimenter un agent chargé d'enrichir le DRC vers une correspondance CIM 11

Automatisation des processus

Tâches administratives répétitives : génération de rapports, mise à jour de tableaux de bord, coordination entre services — exécutées de manière fiable et régulière.

Défis et limites de l'Agentic IA

Fiabilité & Hallucinations

Les LLM peuvent générer des informations incorrectes. Des **mécanismes de vérification** sont nécessaires lorsqu'un agent prend des décisions **autonomes**.

Sécurité & Contrôle

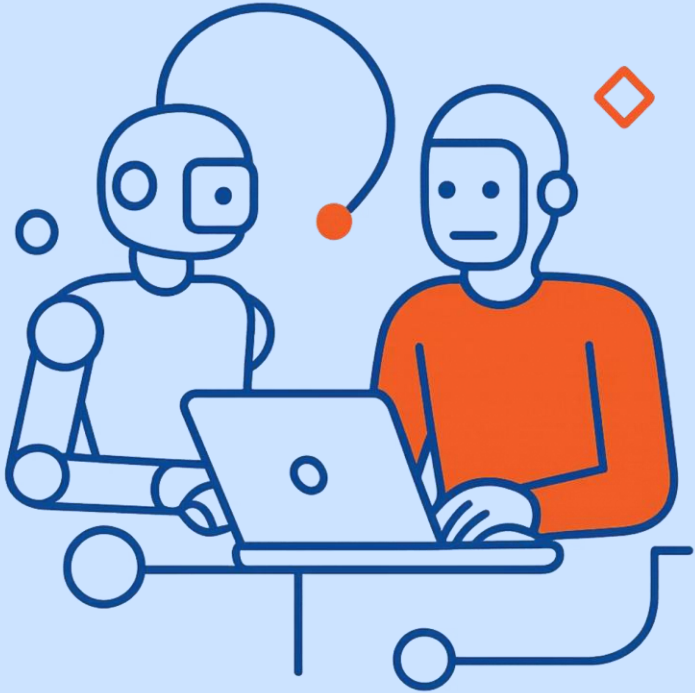
Donner à un agent la capacité d'envoyer des emails ou de modifier des bases de données soulève des questions critiques. Des garde-fous et des audits sont indispensables.

Coût computationnel

La boucle de **raisonnement-action** peut nécessiter de nombreux appels au LLM, générant des **coûts significatifs en tokens** consommés.

Complexité du debugging

Le comportement émergent de ces systèmes rend l'identification des problèmes **moins prévisible** que pour des logiciels traditionnels.



L'avenir de l'Agentic AI



Multi-agents spécialisés

Un agent expert en **recherche** collabore avec un agent **rédacteur** et un agent de **vérification** des faits pour produire du contenu de qualité.



Agentic AI + RAG

La combinaison avec les systèmes **RAG** ancre les agents dans une base de connaissances spécifique, **réduisant** les hallucinations.

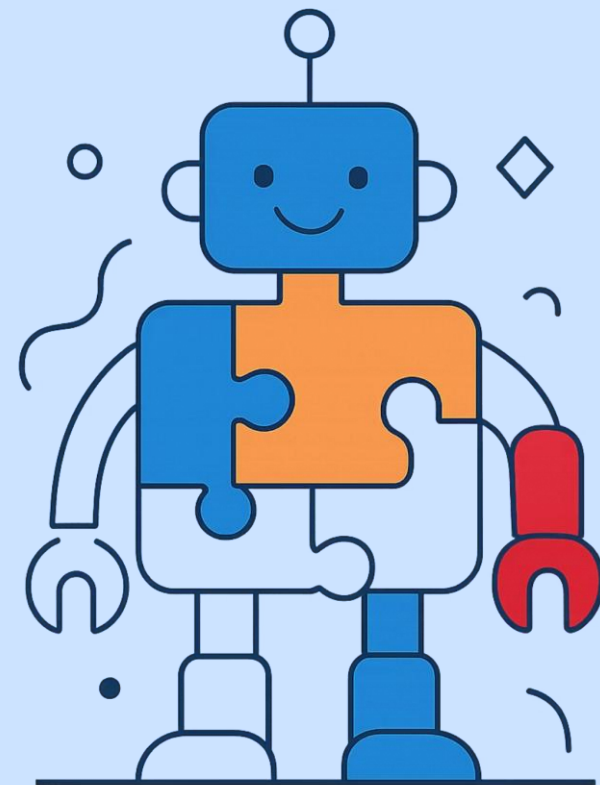


Personnalisation poussée

Des agents adaptés à chaque utilisateur, **apprenant** ses préférences et son contexte au fil du temps.

Au-delà du chatbot : les Agent Skills

Ce qui distingue un agent véritablement utile d'un simple chatbot amélioré, c'est sa capacité à maîtriser des **compétences spécifiques et contextualisées** : les agent skills (compétences).



Qu'est-ce qu'un Agent Skill ?

Un skill est une **unité de compétence encapsulée** qu'un agent peut mobiliser pour accomplir une catégorie spécifique de tâches. Il va bien au-delà d'un simple outil.



Instructions spécialisées

Directives sur quand et comment utiliser la compétence, incluant bonnes pratiques et pièges à éviter.



Outils associés

Fonctions que le skill peut invoquer : APIs, systèmes de fichiers, bases de données.



Contexte métier

Connaissances spécifiques au domaine : conventions de code, règles métier, APIs internes.



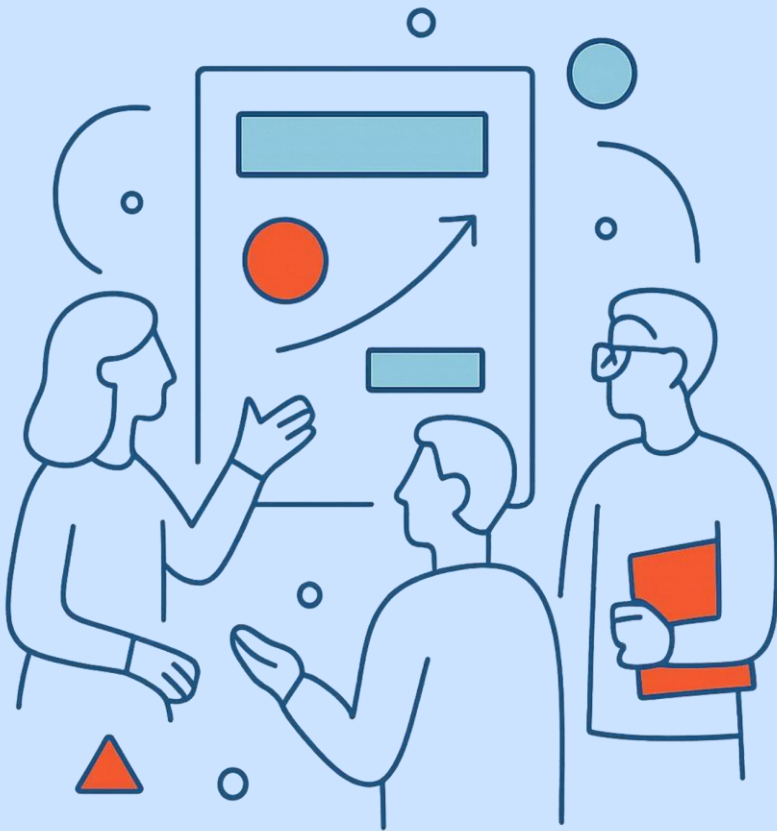
Patterns de raisonnement

Stratégies de résolution de problèmes propres à ce type de tâche.

Outil classique vs Agent Skill

Aspect	Outil classique	Agent Skill
Granularité	Fonction atomique	Compétence complète
Intelligence	Exécution mécanique	Raisonnement contextuel
Configuration	Paramètres techniques	Instructions + contexte métier
Réutilisabilité	Générique	Spécialisé et adaptable
Autonomie	Nécessite orchestration externe	Auto-suffisant pour son domaine

Cette modularité permet une **personnalisation progressive** : on commence avec les capacités natives de l'agent, puis on ajoute des skills au fur et à mesure que des besoins spécifiques émergent.



Les avantages stratégiques des Skills

→ **Capitalisation des connaissances**

L'expertise tacite des seniors devient un asset explicite et partageable. Quand un expert quitte l'équipe, son savoir reste disponible dans les skills.

→ **Cohérence des pratiques**

Tous les membres d'une équipe utilisant les mêmes skills convergent vers les mêmes standards.

→ **Évolutivité maîtrisée**

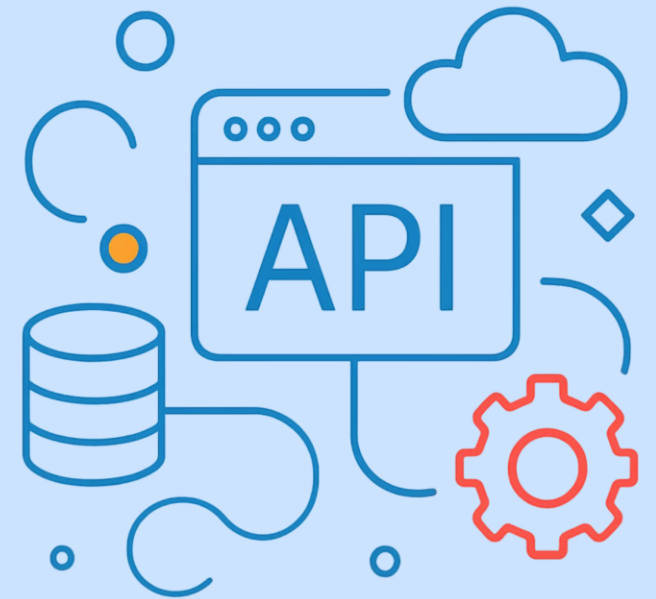
Chaque skill est isolé, modifiable et testable indépendamment. L'expérimentation sur un périmètre restreint avant généralisation est facilitée.

→ **Personnalisation contextuelle**

Un même agent de base peut être adapté à des contextes radicalement différents via des skills spécifiques.

Le Tool Calling : pont vers le monde extérieur

Un LLM, aussi puissant soit-il, reste confiné à la génération de texte. Le **tool calling** (ou function calling) comble ce fossé en permettant au modèle de reconnaître quand une requête nécessite **une fonction externe**, de déterminer laquelle appeler, et d'intégrer le résultat dans sa réponse.



Le cycle du Tool Calling



Le LLM joue un rôle d'**interprète et de routeur** — il ne peut pas lui-même exécuter du code, mais produit une sortie structurée que l'application exécute.

Tool Calling vs Agent IA : une distinction fondamentale

Tool Calling

Interaction **contrôlée et prévisible**. Le LLM analyse la requête, décide de l'outil, produit un appel structuré, et formule sa réponse. C'est le **mécanisme d'action**.

Agent IA

Introduit une **boucle de raisonnement autonome** permettant de traiter des problèmes complexes nécessitant planification et adaptation. C'est **l'intelligence qui décide comment et quand agir**.

- 📄 Le tool calling est le prérequis incontournable pour construire des architectures agentiques plus sophistiquées.

Les Tokens : l'unité de base de l'IA

Un **token** est l'unité fondamentale que les modèles d'IA utilisent pour lire, écrire et **facturer** :
1000 tokens $\approx$ 4000 caractères ou 750 mot en anglais (et **moins** en Français...).

Tokens d'entrée (**Prompt**)

Votre texte, question ou contexte envoyé à l'IA.

Tokens de sortie (**Completion**)

La **réponse** générée par l'IA — souvent facturée plus cher que l'entrée.

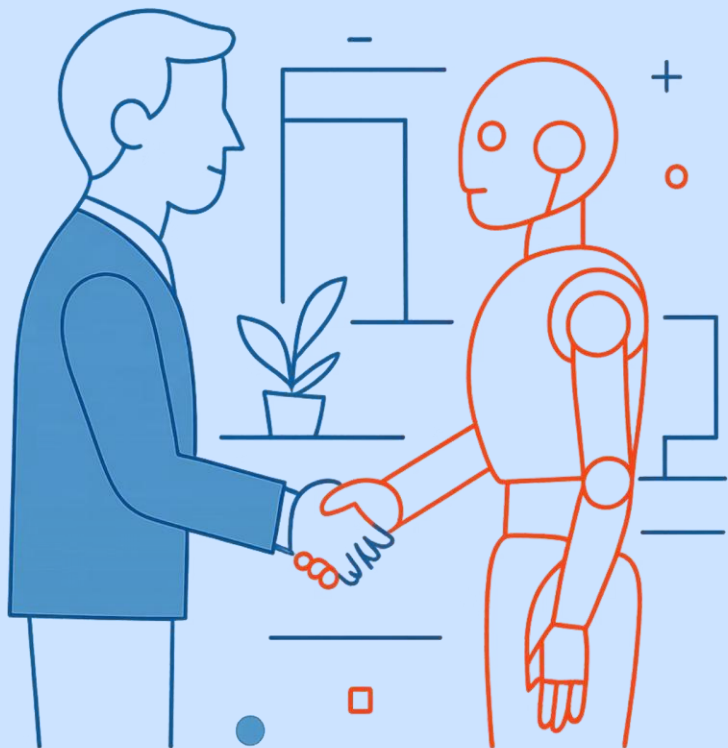
Fenêtre de contexte

Nombre maximal de tokens gérés simultanément.
Dépasser cette limite **fait planter ou dégrade** la discussion.

 Tableau comparatif des prix (en € pour 1 million de tokens)

Fournisseur & Modèle	Rôle / Catégorie	Prix Input (Entrée) / 1M	Prix Output (Sortie) / 1M
 DeepSeek-V3	Ultra-économique / Flash	~0,18 € à 0,35 €	~0,60 € à 0,90 €
 OpenAI GPT-6 Luna	Très léger (ex-mini)	~0,18 €	~1,10 €
 Anthropic Claude Haiku 4.5	Léger et rapide	~0,92 €	~4,60 €
 Google Gemini 3.6 Flash	Intermédiaire rapide	~1,38 €	~6,90 €
 OpenAI GPT-6 Sol	Flagship raisonnement	~3,68 € (promo)	~18,40 €
 Anthropic Claude Sonnet 5	Flagship polyvalent	~1,84 €	~9,20 €
 Google Gemini 3 Pro	Raisonnement haut de gamme	~1,84 €	~11,04 €
 OpenAI GPT-6 Astra	Ultra-frontière / Expert	~9,20 €	~46,00 €
 Anthropic Claude Fable 5.1	Très complexe / Expert	~9,20 €	~46,00 €

Les Tokens :
coût de l'unité de base



Conclusion : vers une IA véritablement autonome

L'Agentic AI, les agent skills et le tool calling forment un écosystème cohérent qui transforme les LLM en véritables collaborateurs autonomes. Cette puissance accrue s'accompagne de responsabilités nouvelles en termes de sécurité, de fiabilité et de contrôle.

Implémenter progressivement

Commencer par des cas d'usage simples avant de déployer des agents complexes.

Superviser et auditer

Mettre en place des garde-fous et des systèmes d'approbation humaine pour les actions sensibles.

 Questions ? 

Merci de votre attention !

