



Le meilleur de l'expertise en IA d'IBM avec
l'approche agentique de CrewAI



Aya Cherkaoui,
Spécialiste Technique Data & AI
aya.cherkaoui@ibm.com



Mattéo Procoppe,
Spécialiste Technique Data & AI
mprocoppe@ibm.com

Agenda

1. Introduction aux agents IA : c'est quoi un agent IA? Pourquoi on a besoin d'un agent IA?
2. C'est quoi CrewAI?
3. Comment CrewAI s'intègre avec watsonx.ai?
4. Démo

Assistant traditionnels

VS.

Assistants IA

VS.

Agents IA

Assistant = chatbot

Assistant traditionnel



Parcours scriptés

Les interactions sont basées sur des règles et les réponses sont pré-codées dans l'outil de manière déterministe.



Utilisation des synonymes

L'outil a besoin de nombreux synonymes pour déterminer les intentions et les entités.



Expérience robotique

Interaction limitée à des réponses basiques et pas de compréhension du contexte ou des nuances.

Assistant IA



Conversationnel

Gère les conversations complexes grâce à la détection d'intention, à la clarification et au changement de sujet.



Connaissance du Contexte

Offre des expériences contextualisées à chaque étape du processus en s'améliorant avec le temps.



Gen AI

Met en évidence des compétences d'IA générative (génération, extraction, résumer, classifier, etc.).

Agent IA



Décision autonome de la Gen AI

Les modèles d'IA générative planifient et répondent au besoin de manière autonome.



Utilisation de « Tools » ou outils

Ressources externes que l'agent peut utiliser pour effectuer des tâches spécifiques (automatisation, API, etc.)



Multi-agents

Plusieurs modèles d'IA générative collaborent pour atteindre un objectif commun.

Hier

Demain

Définition

Le terme *agent* désigne une entité (une personne ou un phénomène) qui a une action déterminante. ([wikipedia](https://fr.wikipedia.org/wiki/Agent))

Système classique :

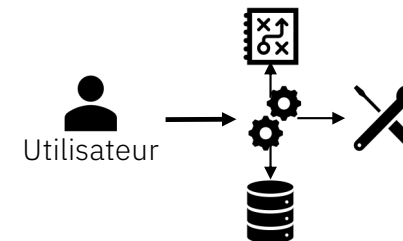
Système qui effectue des tâches spécifiques en fonction de *règles prédéfinies* (règles métier, règles NLP, statistiques)

Système agentique :

Un système conçu pour *raisonner* à travers des problèmes complexes, *créer* des plans d'action et *exécuter* ces plans à l'aide d'une suite *d'outils*.

Composants des agents AI :

- **Modèles d'IA Générative** : variété de LLM spécialisés permettant d'analyser les problèmes, de planifier leur résolution et d'appeler des outils.
- **Outils** : de l'Anglais *tools*, ces ressources externes et API permettent à l'agent d'accéder à des informations externes, d'agir et d'accomplir des tâches spécifiques.
- **Module de mémoire** : stocke et récupère des informations pour maintenir le contexte de la conversation, le besoin de l'utilisateur, et la continuité au fil du temps.
- **Noyau de l'agent** : l'unité centrale de traitement qui intègre toutes les fonctionnalités du système (e.g., *frameworks* opensource [LangChain](#), [Crew.ai](#), [Bee](#), [AutoGen](#)).



Exemple

"Cherche la météo à Paris, et enregistre le résultat dans le fichier meteo.txt"

- Le LLM peut
 - « comprendre » la question
 - prévoir comment y répondre
- Mais n'a pas
 - pas l'information à jour
 - pas de accès internet
 - pas d'accès au système de fichiers
-

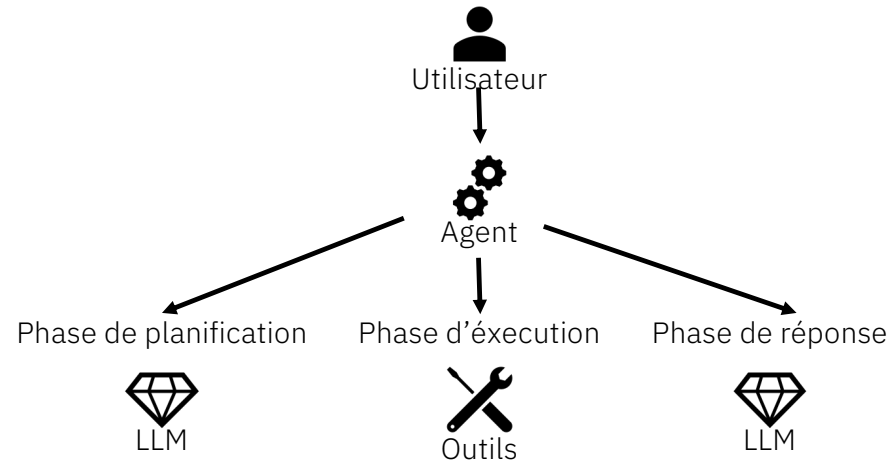
- Un Agent va
 - Planifier comment répondre
 - Utiliser des outils pour accéder à des données externes et agir

Exemple d'outils :

```
def get_meteo(city: str):  
    """Utilise cet outils pour récupérer la météo dans une ville.  
    Exemple:  
    city = Paris  
    """  
    redef write_to_file(file_path: str, text: str):  
        """Utilise cet outils pour écrire du texte dans un fichier.  
        Exemple:  
        file_path = ./output.txt  
        text = Hello World  
        result = Success  
        """  
        try:  
            with open(file_path, "w") as f:  
                f.write(text)  
                f.close()  
        except IOError:  
            logging.error(f"Impossible d'écrire dans {file_path}")
```

Exemple

"Cherche la météo à Paris, et enregistre le résultat dans le fichier meteo.txt"

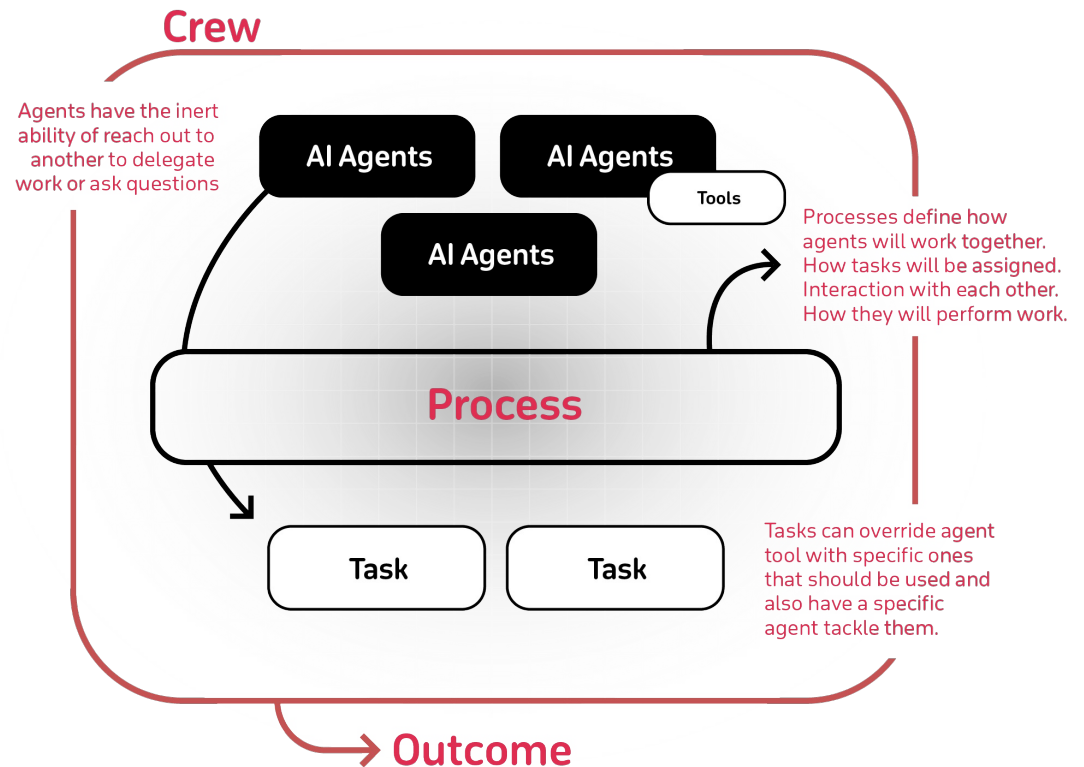


Grandes étapes:

1. Identifier les outils avec un LLM entraîné sur l'utilisation d'outils, ex LLAMA3.1
 - Prompt: « Parmi les outils {tools descriptions} quels sont ceux nécessaires pour répondre à la question {question} »
 - Réponse du LLM: [`{"tool": "get_meteo", "args": {"city": "paris"}, "return": "str"}, ...]`]
2. Extraire les paramètres de la question (LLM), appeler chaque outils (Python Classique), enchaîner les appels
3. Préparer la réponse finale à l'utilisateur (LLM)

CrewAI c'est quoi?

- crewAI est un framework d'orchestration multi-agents open source basé sur Python.
- crewAI permet l'orchestration des agents IA autonomes qui travaillent ensemble comme une « équipe » ou un “crew” pour accomplir des tâches.
- L'objectif de crewAI est de fournir un cadre solide pour automatiser les flux de travail multiagents.¹



Démo

watsonx.ai™ + *crewai*

Cas d'usage : Stratégie marketing d'un produit