

# Bruno Legeard

IA Générative : Kit de  
survie pour les Testeurs

**11 JUIN 2024**  
BEFFROI DE MONTROUGE



JOURNÉE  
FRANÇAISE  
DES TESTS  
LOGICIELS

# Au programme

## Utiliser l'IA générative pour accélérer les tests : aspects clés

-  Les bases de l'IA générative
-  Tester avec l'IA générative
-  Techniques de prompting
-  Maîtriser l'usage de l'IA générative et éviter les risques
-  Applications basées sur l'IA générative pour les tests logiciels

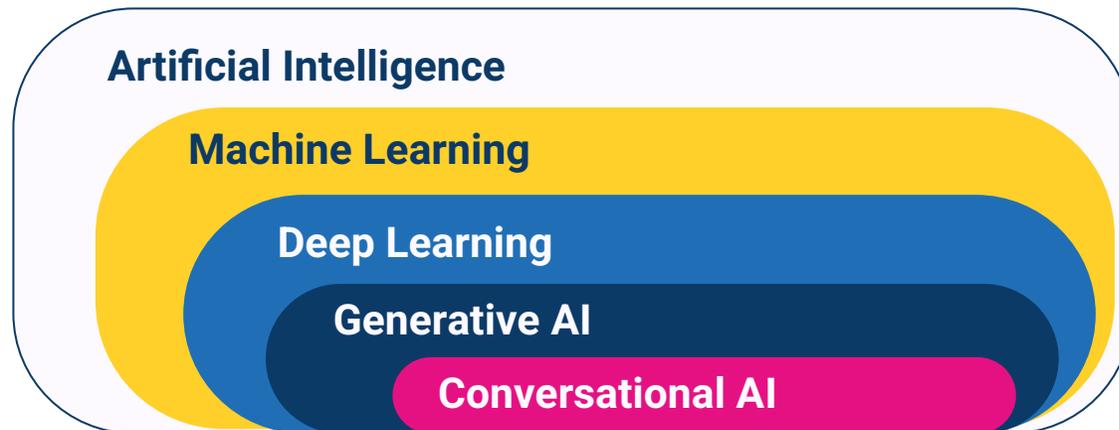
Cette présentation utilise des extraits de la formation “**Accélérer les processus de test grâce à l'IA générative**” par la pratique, conçue sur 2 jours pour la profession Testeur.



# IA générative - Positionnement

## Machine learning, deep learning et IA générative

Différents types de technologies IA



L'**IA générative** fait partie des techniques de Deep Learning, avec la particularité d'utiliser des **modèles pré-entraînés** sur de grands corpus de données.

**LLM** = Large Language Model



# A la source de l'IA générative

## Les modèles "Transformers"

[Submitted on 12 Jun 2017 (v1), last revised 2 Aug 2023 (this version, v7)]

### Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin

<https://arxiv.org/abs/1706.03762> - Google Research - Juin 2017

- Mécanisme d'attention : permet de sélectionner le meilleur lien sémantique dans une séquence de mots
- Plus efficace que les réseaux de neurones classiques pour les tâches de traitement du langage
- Passe à l'échelle pour l'apprentissage sur une grande quantité de données (parallélisable)

**GPT** = **G**enerative **P**re-Trained **T**ransformer



# Vocabulaire : Token & Embeddings

Découper du texte naturel en “morceaux” et les vectoriser

1) **Texte d'entrée** : “Le chat est assis sur le \_\_\_\_\_”

2) **Tokenization** : le chat est assis sur le

<https://tiktokenizer.vercel.app/>

Tokens	Characters
7	24
[2356, 6369, 1826, 1089, 285, 1765, 514]	

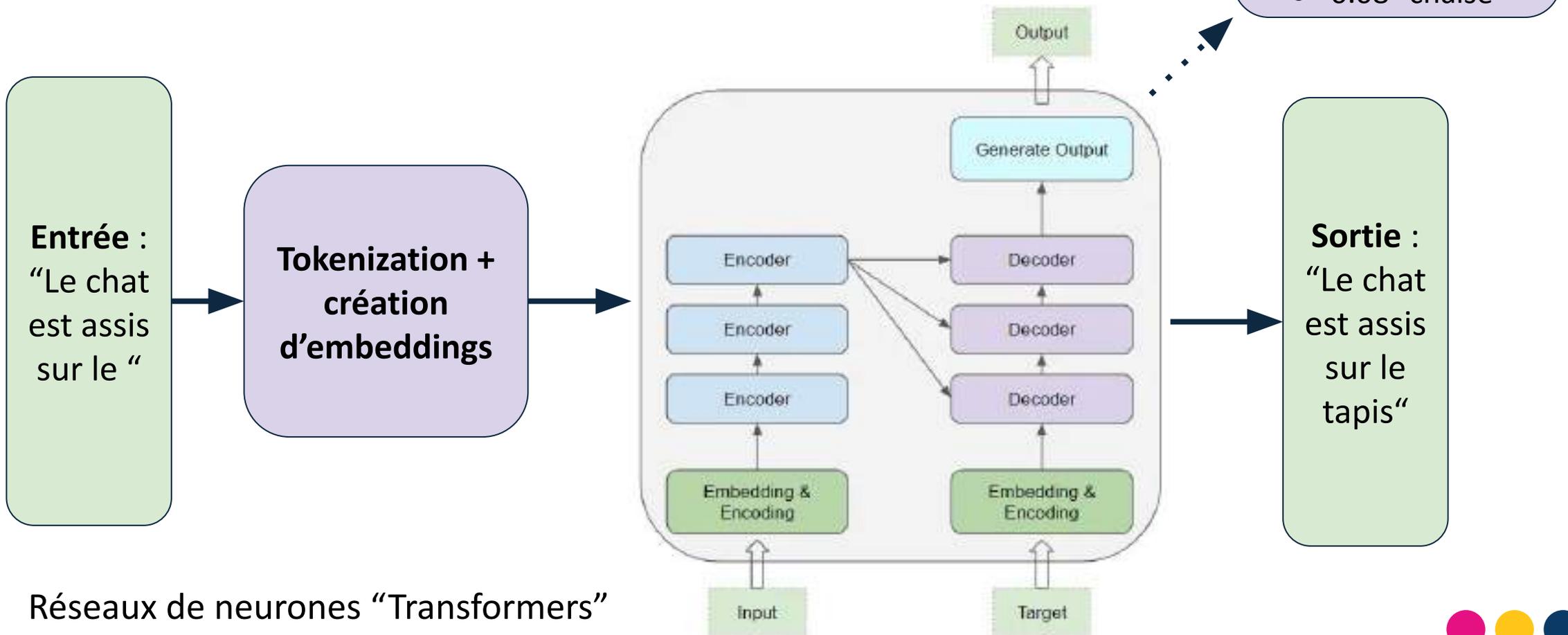
3) **Embeddings** : Chaque token est associé à un vecteur appelé embedding

- Les tokens sémantiquement similaires auront des embeddings avec des valeurs proches

```
[[-0.02388945  0.05525832 -0.01165408 ...  0.08577387  0.03409707 -0.0068091 ]
 [-0.0124876  0.04607412 -0.01050217 ... -0.02110218 -0.08270466  0.01047171]
 [ 0.00049438  0.11943285  0.08522909 ...  0.01607654 -0.02386131  0.00526433]
 ...
 [-0.03908795 -0.01860951 -0.08738371 ... -0.08390449  0.03768485  0.08211161]
 [-0.04590175 -0.06301160 -0.11006582 ...  0.08549041  0.1528919  0.02402813]
 [-0.01162949  0.00961934  0.01608981 ... -0.02821241 -0.00116556  0.0036672 ]]
```



# Les modèles “Transformers”



Réseaux de neurones “Transformers”



# La taille du modèle

## Que veut dire “Mistral-7B” ?

Le nombre de paramètres d’un modèle de Deep Learning est une **donnée structurelle**

- **Quantité de données d’apprentissage** sur lesquels le modèle peut être entraîné

**Plus de paramètres = plus gros corpus d’entraînement = plus grande capacité de réponse = plus de puissance de calcul nécessaire pour l’utiliser.**

Souvent indiqué dans le nom du LLM

- Mistral 7B → 7 Milliards de paramètres
- Llama 3 70B → 70 Milliards de paramètres
- GPT-4 → Composition de 8 modèles de 220 Milliards de paramètres (Mixture-of-Expert)

## Mais la taille ne fait pas tout !

La qualité des données d’entraînement, le fine-tuning, les astuces d’architecture, etc.... comptent pour beaucoup.



# La notion de “fenêtre contextuelle”

## Limite importante de capacité d'un LLM

- La “fenêtre contextuelle” - *context window* - d'un modèle de langage est la **quantité de texte** que ce modèle peut **traiter en une seule fois** pour réaliser des tâches.
- Cette fenêtre est limitée : GPT-3.5 Turbo - 16k tokens et Claude-3 - 200k tokens.
- Toute la conversation (requêtes et réponses) est considérée dans cette limite lors d'une suite de message → cela signifie qu'une partie de la conversation **sera coupée** au fur et à mesure de la suite des échanges
- Plus la fenêtre est grande et plus les calculs nécessitent de **mémoire et de puissance de calcul** → augmentation quadratique



# La “température” d’un modèle LLM

## Degré de liberté laissé dans le choix du token suivant lors des calculs du modèle

- **Hyper-paramètre dont la valeur varie entre 0 et 1 (entre 0 et 2 pour OpenAI)**
- **Proche de 0 : favorise le choix le plus probable, et augmente la répétabilité des réponses**
  - Diminue les effets de créativité et de variations dans les réponses
- **Proche de 1: augmente l’aléatoire dans le choix du token suivant**
  - Augmente la variabilité des réponses (et le risque d’incohérence)
- **A quelle température est mon LLM ?**
  - ChatGPT de OpenAI ou Gemini Chat de Google → vers 0.7
  - Copilot de Microsoft → autour de 0.5
  - Pour les activités de test → plutôt 0.5 de façon à rester sur les connaissances les plus probables



# Du LLM de base au chatbot sophistiqué

## LLM de base - (Base LLM)

Prédit le mot suivant à partir des données **textuelles** apprises et de la **proximité linguistique**.

### Exemples

**Il était un fois une princesse** nommée Isabella. Elle vivait dans un grand château au sommet d'une colline verdoyante. ...

**Quelle est la capitale de la mongolie ?**

Quelle est la superficie de la mongolie ?

Quelle est le nombre d'habitants de la mongolie ?



## LLM adapté aux instructions- (Instruction-Tuned LLM)

- Entraîné pour **suivre des instructions**
- RLFH : Reinforcement Learning with Human Feedback
- **Modération** des réponses : éviter les réponses agressives, racistes, dangereuses, ...

### Exemple

**Quelle est la capitale de la mongolie ?**

La capitale de la Mongolie est Oulan-Bator.



# LLM Multimodal

## Combiner textes, images, sons ...

- Entraîné sur de large corpus de **données multimodales** : textes, images, sons, etc.
- Utilisé pour des **tâches multimodales** : générer une image à partir d'un texte, générer une description textuelle d'une image ou d'une vidéo, etc.

Exemple d'usage pour les tests : **fournir un**

- **screenshot pour compléter l'information de contexte sur une User Story**



Génère des cas de test pour cette application.

🗨 ChatGPT

Pour générer des cas de test pour cette application de création de logos, voici quelques scénarios que vous pourriez envisager :

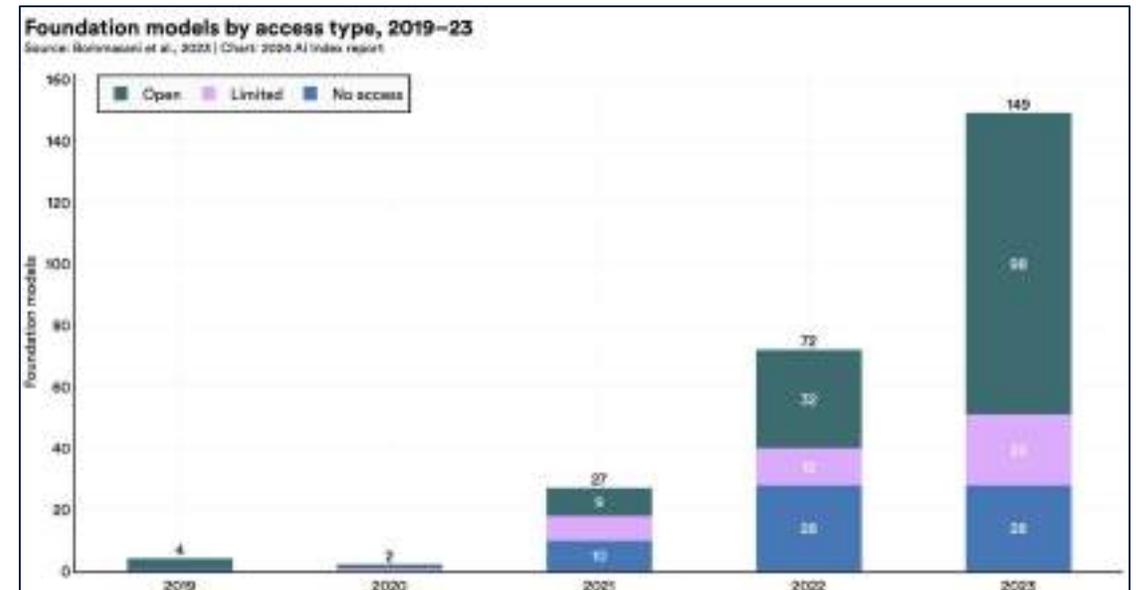
1. Test de Validation des Entrées
  - Entrée valide : Saisir un nom d'entreprise classique, une phrase d'accroche, et des mots-clés pertinents, puis cliquer sur "Suivant".
  - Entrée invalide : Laisser des champs obligatoires vides, entrer des caractères spéciaux ou inappropriés dans le nom d'entreprise ou les mots-clés, et tester la réaction de l'application.
2. Test de Fonctionnalité
  - Changer le type de police et observer si le rendu dans l'aperçu est modifié en conséquence.
  - Sélectionner différentes options de couleurs de fond et de finitions pour voir si elles sont appliquées correctement.



# Un grand choix de LLM disponibles

## Une accélération de l'offre LLM

- 👉 De plus en plus de LLM disponibles.
- 👉 Avec une grande variété de LLM : puissance / taille, type de licence, focalisation, etc...
- 👉 Plus d'opportunités pour choisir le modèle le plus pertinent dans son contexte (cf. section 5).



Source: 2024 AI Index Report Stanford University  
<https://aiindex.stanford.edu/report/>



# Au programme

## Utiliser l'IA générative pour accélérer les tests : aspects clés



Les bases de l'IA générative



Tester avec l'IA générative



Techniques de prompting



Maîtriser l'usage de l'IA générative et éviter les risques



Applications basées sur l'IA générative pour les tests logiciels

Cette présentation utilise des extraits de la formation “**Accélérer les processus de test grâce à l'IA générative**” par la pratique, conçue sur 2 jours pour la profession Testeur.



# Ce que l'IA générative apporte aux tests logiciels

## Un impact sur de nombreuses activités de test

 **Interpréter** des specs, de exigences, des images écrans, des cas de test, des rapports d'anomalies, etc...

 **Générer** des conditions de test, des cas de tests, du code de test, etc...

 **Synthétiser** des rapports d'avancement, d'anomalies, de couverture, etc...

 **Evaluer / corriger** la couverture de test, le code de test, etc...

1. Planification des tests

2. Suivi et contrôle des tests

3. Analyse de test

4. Conception des tests

5. Implémentation des tests

6. Exécution des tests

7. Clôture des tests



# IA générative & Tests

## Analyser et améliorer les exigences

Identifier les ambiguïtés, les incohérences et les informations manquantes

Générer des questions pertinentes pour guider les discussions avec les parties prenantes

## Aider à la conception des tests

Proposer des idées de cas de test pertinents

Générer des scénarios de test, des conditions limites et des combinaisons de données de test

## Aider à l'automatisation des tests

Générer du code de test automatisé à partir de descriptions en langage naturel

Proposer des améliorations au code existant

Identifier les modèles de conception de test appropriés

Générer des scripts de test pour différents frameworks d'automatisation

## Faciliter la maintenance des tests

Identifier les doublons

Suggérer des optimisations

Mettre à jour les tests obsolètes suite aux changements de spécifications

Générer de la documentation de test actualisée

## Générer des données de test

Créer des jeux de données de grande taille

Générer des cas de test aux limites

Produire des données synthétiques pour les tests de performance et de sécurité

## Analyser les résultats d'exécution

Identifier les modèles d'échec

Suggérer des causes potentielles

Générer des rapports de synthèse

Classer et hiérarchiser les défauts en fonction de leur sévérité et de leur impact

## Créer de la documentation

Générer des plans de test

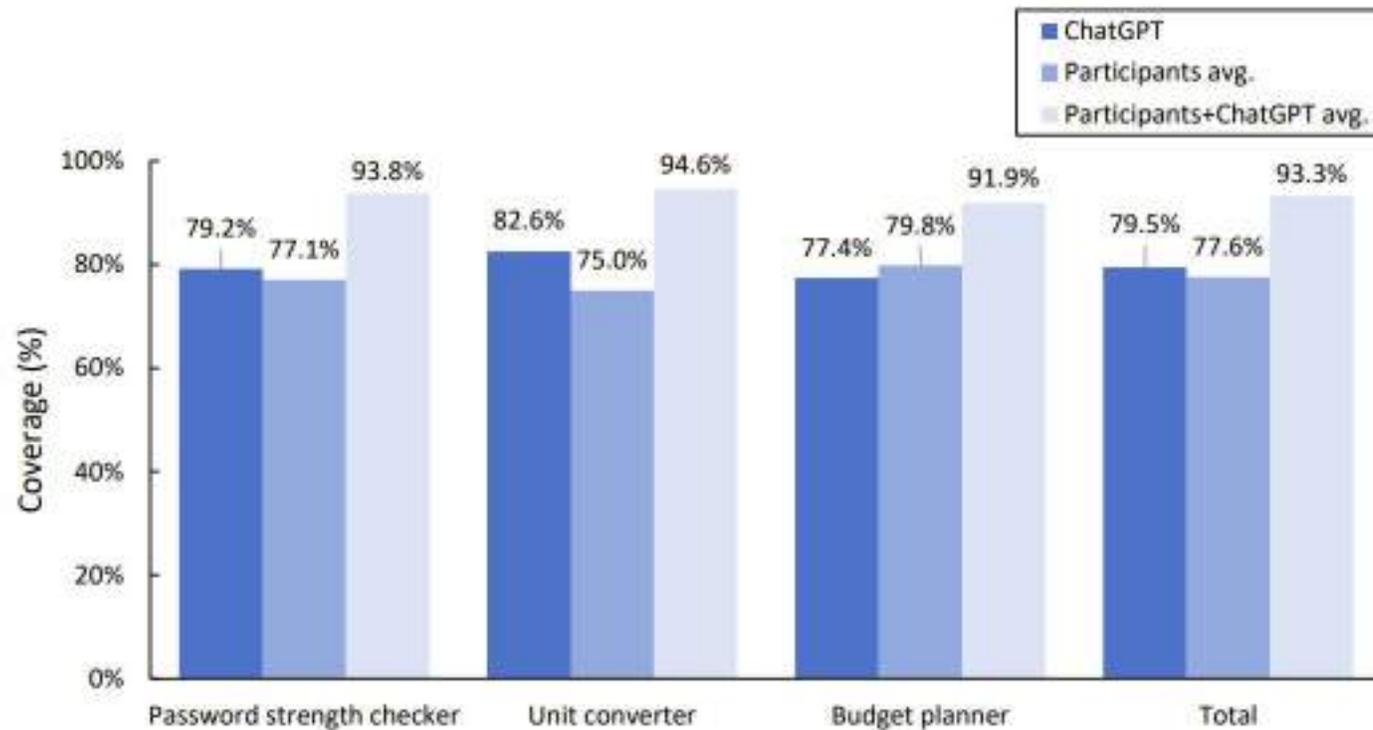
Produire des rapports de test

Créer des guides utilisateur

Tenir à jour la documentation en fonction des changements apportés au système et aux processus de test

# Synergie Humain et IA

Augmentation par l'IA générative pour des tâches de conception de tests



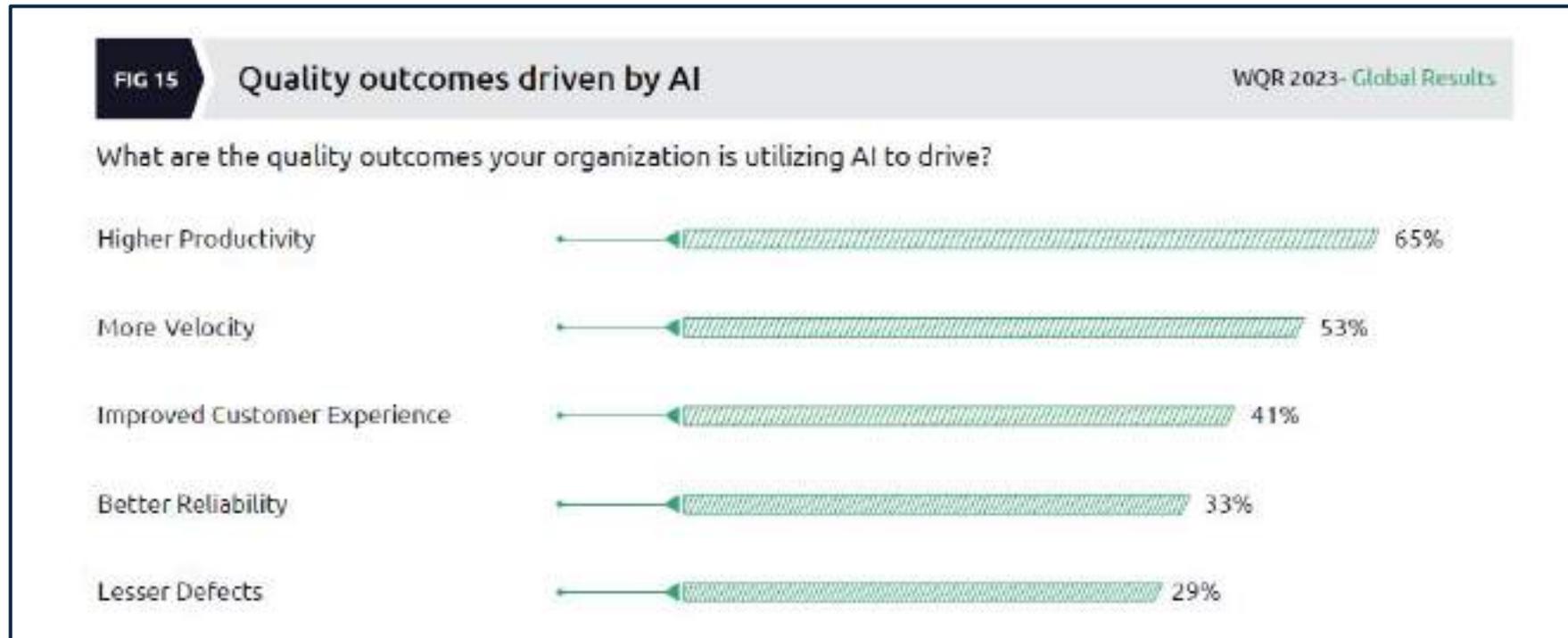
Source : article "ChatGPT and Human Synergy in Black-Box Testing: A Comparative Analysis" - Janvier 2024

<https://arxiv.org/abs/2401.13924>



# Des attentes fortes sur l'IA

Productivité et vélocité en tête des attentes des organisations



Source : Word Quality Report - 2023-24



# Utiliser l'IA générative pour les tests

## Comment utiliser un LLM ?

Les échanges sont toujours réalisés sous forme de conversation

Requête (Prompt) ➡ Réponse du LLM

- **Via Chatbot** : plus convivial pour converser / interroger directement le LLM
- **Via API** : plus souple et configurable. Permet de développer des applications basées LLM (cf. section 4 du cours).

Dans les deux modes, l'**ingénierie et les bonnes pratiques de prompting** sont requises (cf. section 2).



# Au programme

## Utiliser l'IA générative pour accélérer les tests : aspects clés



Les bases de l'IA générative



Tester avec l'IA générative



Techniques de prompting



Maîtriser l'usage de l'IA générative et éviter les risques



Applications basées sur l'IA générative pour les tests logiciels



# Prompt Engineering

## Signification du terme “Prompt Engineering”

**Prompt Engineering** fait référence à la conception de requêtes de qualité qui guident les modèles LLM pour produire des sorties précises.

💡 Cela implique de sélectionner le bon **type de requêtes**, d'**optimiser leur longueur** et **leur structure**, et de déterminer leur **ordre** et leur **pertinence** par rapport à la tâche.

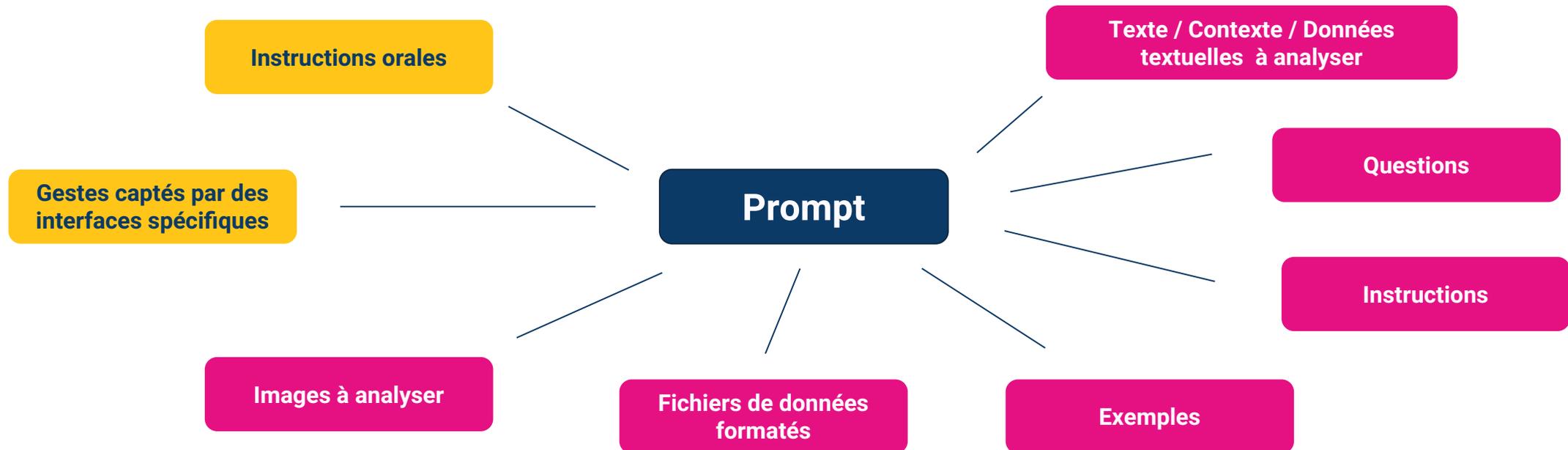
👉 Vos résultats vont dépendre du modèle LLM utilisé et de vos prompts

**Prompt Engineering = Travailler la structure, le style et le contenu des requêtes**



# Que trouve-t-on dans un “prompt” ?

Un prompt peut être multimodal



# Prompt Engineering pour une tâche

Le Prompt Engineering n'est pas universel, mais doit correspondre aux types d'activités à réaliser.

## Marketing - Prompts pour la production de messages & slogans

- Accent sur la **créativité**, la voix de la marque et la **persuasion**
- Requêtes conçues pour générer un contenu qui
  - Résonne avec un **public cible** spécifique
  - S'aligne sur **l'identité de la marque**
  - **Incite à l'action**
  - Intègre le **référencement et les mots-clés**

## Tests logiciels - Prompts pour la génération de cas de test

- Accent sur la **précision**, **l'exhaustivité** et la garantie que **tous** les scénarios possibles sont testés
- Requêtes très spécifiques et techniques
  - Incluant des **descriptions détaillées** des fonctionnalités à tester
- Langage **plus structuré** et moins axé sur la créativité que dans le domaine du marketing



# Structure de base du prompt

**Le prompt type contient 6 parties :**

## Rôle

Quel rôle (par exemple “concepteur de tests”) est assigné au LLM

## Contexte

- Décrire succinctement le contexte de la tâche

## Instructions

- Ce qu’il faut réaliser - éventuellement en plusieurs étapes

## Format

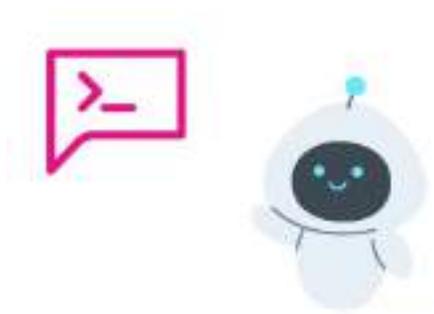
- Le format de sortie attendu

## Contraintes (éventuelles)

- Pour préciser des limitations ou exigences de réalisation

## Données (éventuelles)

- Exemples de réponse
- Données de la tâche à réaliser



# Exemple de prompt structuré

## Utiliser la structure Rôle-Contexte-Instructions-Contraintes-Format-Données

Tu es mon assistant pour l'écriture de scénarios de test en format Gherkin (Given/When/Then) à partir d'une exigence donnée. Le système sous test est l'application web Logogénie (voir <https://www.logogenie.fr/>).

En analysant cette exigence, réalise les tâches suivantes :

1. Détermine de manière exhaustive les classes d'équivalence à tester, en donnant quelques exemples de valeurs typiques et aux limites pour les classes valides et invalides. Explique ton raisonnement.
2. Crée les scénarios dans un fichier <nom\_feature>.feature, en regroupant les scénarios par fonctionnalité. Assure-toi que toutes les classes d'équivalence sont couvertes avec au moins 2 tests chacune.

#

Exigence :

La création du logo nécessite que les champs suivants soient saisis : nom de l'entreprise (obligatoire, 40 caractères max), phrase de slogan (50 caractères max, optionnel), et mots-clés (au moins un).

 Analyser ce prompt et trouver les éléments de la structure



# Analyse du prompt

Contexte

Contraintes

Rôle

Format

Instructions

Données

Tu es mon **assistant pour l'écriture de scénarios de test en format Gherkin (Given/When/Then)** à partir d'une exigence donnée. Le **système sous test est l'application web Logogénie (voir <https://www.logogenie.fr/>)**.

En analysant cette exigence, réalise les tâches suivantes :

1. Détermine de manière exhaustive les classes d'équivalence à tester, en donnant quelques exemples de valeurs typiques et aux limites pour les classes valides et invalides. Explique ton raisonnement.
2. Crée les scénarios dans un fichier **<nom\_feature>.feature**, en regroupant les scénarios par fonctionnalité. Assure-toi que toutes les classes d'équivalence sont couvertes avec **au moins 2 cas de tests chacune**.

#

Exigence :

La création du logo nécessite que les champs suivants soient saisis : nom de l'entreprise (obligatoire, 40 caractères max), phrase de slogan (50 caractères max, optionnel), et mots-clés (au moins un).





# Place à la pratique

## Le portail LLM Smartesting

The screenshot displays the Smartesting LLM portal interface. On the left, there is a sidebar with the Smartesting logo and a list of chat sessions. The main area shows a chat window with two models: GPT-4o and CLAUDE-3-OPUS-20240229. The GPT-4o chat shows a user prompt in French asking for test scenarios in Gherkin format for a login feature. The CLAUDE-3-OPUS-20240229 chat shows the model's response, which includes a list of equivalence classes and their values, and a list of test scenarios in Gherkin format.

**GPT-4o**  
Context window: 131069 tokens  
Used: 1794 tokens  
Edit system prompt

**CLAUDE-3-OPUS-20240229**  
Context window: 100000 tokens  
Used: 1398 tokens  
Edit system prompt

**You**  
Tu es mon assistant pour l'écriture de scénarios de test en format Gherkin (Given/When/Then) à partir d'une exigence donnée. Le système sous test est l'application web Logogème (voir <https://www.logogeme.fr/>). En analysant cette exigence, réalise les tâches suivantes :  
Détermine de manière exhaustive les classes d'équivalence à tester, en donnant quelques exemples de valeurs typiques et aux limites pour les classes valides et invalides. Explique ton raisonnement.  
Crée les scénarios dans un fichier «nom\_feature/features», en regroupant les scénarios par fonctionnalité. Assure-toi que toutes les classes d'équivalence sont couvertes avec au moins 2 tests chacune.  
#  
Exigence :  
La création du logo nécessite que les champs suivants soient saisis : nom de l'entreprise (obligatoire, 40 caractères max), phrase de slogan (50 caractères max, optionnel), et mots-clés (au moins un).

**gpt-4o**  
Voici comment procéder pour répondre à votre demande :  
**1. Classes d'équivalence et valeurs**  
Champ "nom de l'entreprise"

**claude-3-opus-20240229**  
Voici mon analyse de l'exigence et les scénarios de test correspondants en format Gherkin :  
**Classes d'équivalence**

# Techniques du Prompting

## Principaux types de requêtes

### Requêtage structuré

Ajouter du contexte, des instructions, des exemples (one-shot / few-shot)

### Requêtage par étapes

Découper votre demande en étapes intermédiaires - Chain-of-Thoughts

### Requêtage non-structuré en cascade

Dans le style d'une conversation humaine avec une suite d'échange de questions / réponses et rebond sur la question suivante

### Méta-requêtage et évaluation

Utiliser un LLM pour créer ou améliorer vos requêtes  
Demander d'évaluer votre prompt



# Au programme

## Utiliser l'IA générative pour accélérer les tests : aspects clés



Les bases de l'IA générative



Tester avec l'IA générative



Techniques de prompting



Maîtriser l'usage de l'IA générative et éviter les risques



Applications basées sur l'IA générative pour les tests logiciels



# Les risques de l'IA générative

## Maîtriser les risques de l'IA générative

- Hallucinations, Erreurs et Biais de l'IA
- Risques de cybersécurité : évocation des données
- Consommation d'énergie / Impact CO2
- Perte de compétences, dépendance à un service tiers
- Régulation de l'IA (European AI Act)



# Hallucinations, Erreurs et Biais

**Erreur = manque d'exactitude / incomplétude / résultat factuellement faux**

**Cause fondamentale : l'IA n'est pas intelligente - elle cherche la proximité linguistique**

- 🙄 Lorsque le modèle génère un **contenu vraisemblable mais erroné**, on utilise le terme **hallucination**
  - Des faits inventés et faux, et des informations fabriquées par l'algorithme mais erronées
  - Des réponses étranges, sans véritable lien avec le prompt
- ✗ Le modèle peut échouer à **remplir la tâche de façon satisfaisante** ou produire une **réponse erronée**
  - Les approximations et incomplétudes sont fréquentes
  - Une **erreur de raisonnement** du modèle peut conduire à un résultat faux
- ➡ La détection de ces erreurs peut être difficile car la réponse est souvent vraisemblable



# Éviter les erreurs ?

## Ou plutôt : comment gérer le risque

### Pour diminuer le risque :

 Fournir des **informations précises** sur le contexte et la tâche

- Dans la prompt (référence / hint / description)
- En **ajoutant des références** ou des document contenant les infos nécessaires
- Nouveau contexte  nouveau chat

 **Évaluer plusieurs LLM** pour une tâche donnée

 **Découper la tâche** avec vérification systématique de chaque résultat

- Tâche plus simple = moins de risque et résultat plus facilement interprétable
- Possibilité d'ajustement via feedback

 Éviter des causes d'erreur habituelles

- Mélange de langues, de contextes, un format de données difficile à interpréter...

 **IMPORTANT:** les LLM n'ont pas un comportement déterministe, et donc pas reproductible à 100%



# Risques de cybersécurité

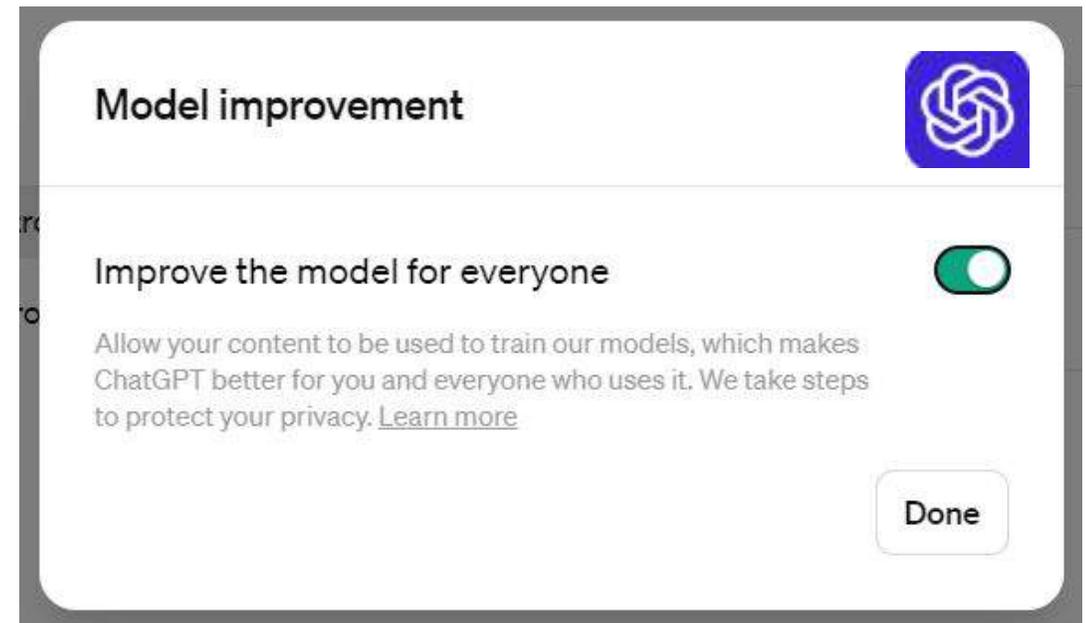
## Des risques spécifiques liés à l'IA générative : la confidentialité des données transmises

### 🔒 Nos requêtes peuvent être utilisées

- Pas défaut sur OpenAI : l'autorisation est donnée pour utiliser nos données pour l'amélioration des modèles
- Même situation chez les autres fournisseurs

### Impératif :

- Vérifier et modifier les options
- Les données sensibles ne doivent pas être envoyées sur un service cloud public



ChatGPT : réglage par défaut en partage des données



# Guide ANSSI

## Recommandations de sécurité IA générative

- 4 scénarios de mise en oeuvre LLM
- Analyse des risques pas scénarios
- 35 recommandations
  - générales
  - phase d'entraînement
  - phase de déploiement
  - phase de production



<https://cyber.gouv.fr/publications/recommandations-de-securite-pour-un-systeme-dia-generative>



# Au programme

## Utiliser l'IA générative pour accélérer les tests : aspects clés



Les bases de l'IA générative



Tester avec l'IA générative



Techniques de prompting



Maîtriser l'usage de l'IA générative et éviter les risques



Applications basées sur l'IA générative pour les tests logiciels



# Applications basées LLM

## Architecture générale

Une application basée LLM encode un flot d'activités incluant :



**Extraction et la préparation des données** de contexte nécessaire à l'activité



**Requêtage d'un modèle LLM** en instanciant un prompt avec les données précédentes et des paramètres de l'activité à réaliser



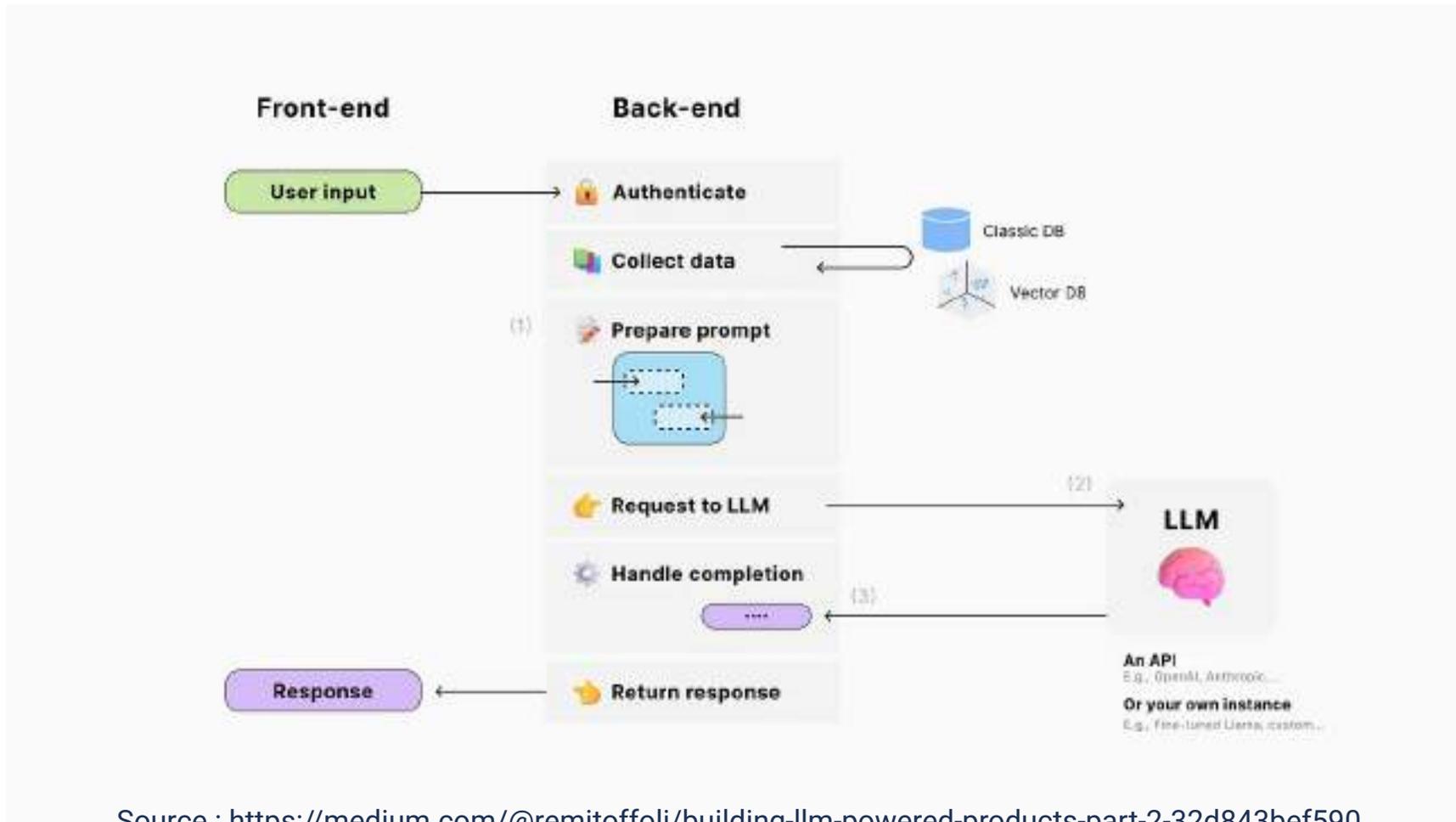
**Raisonnement** par "Chain of Thoughts" ou d'autres techniques de raisonnement en étapes des inférences du (ou des) LLM



**Formatage et export des sorties** de l'application vers une IHM et/ou d'autres outils



# LLM App : exemple d'architecture



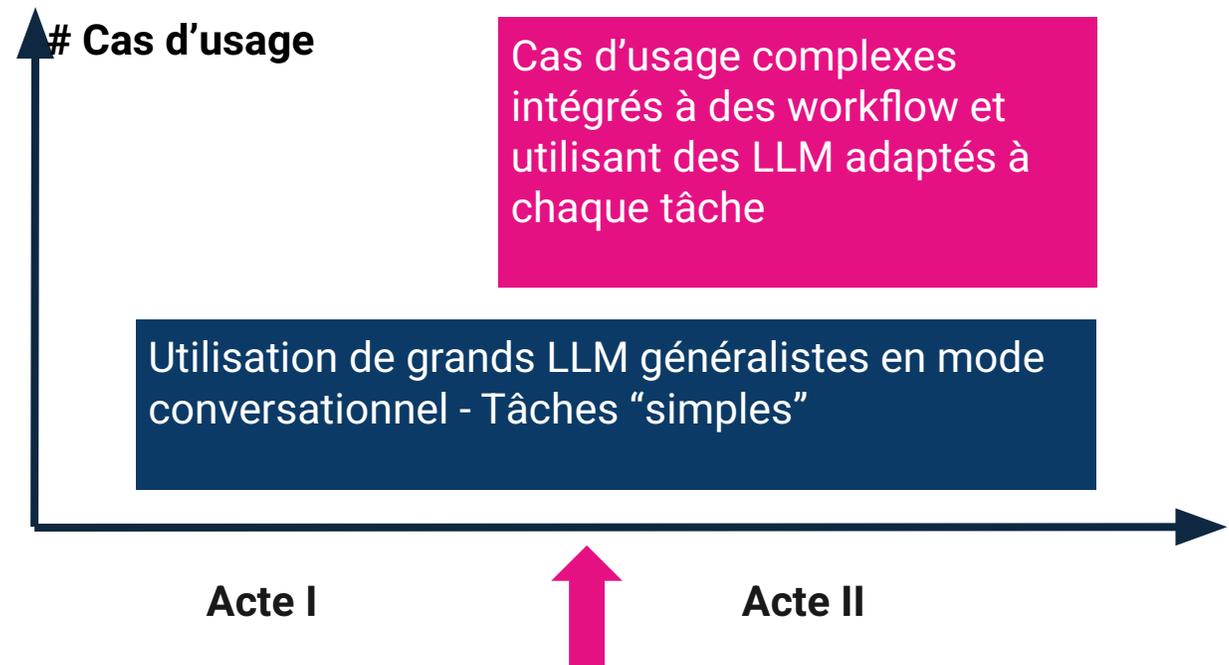
Source : <https://medium.com/@remitoffoli/building-llm-powered-products-part-2-32d843bef590>



# IA générative - Acte II

Nous sommes au début de l'acte II de l'IA générative

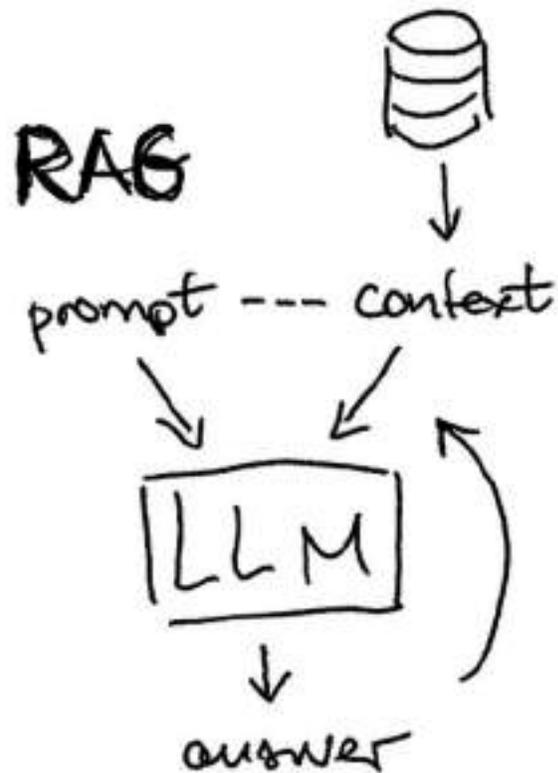
- Des modèles LLM **adaptés** aux tâches visées
- une intégration fine au service de l'utilisateur
- un **usage sécurisé** vis-à-vis des fuites de données et des failles de l'IA générative



# RAG et Agents IA

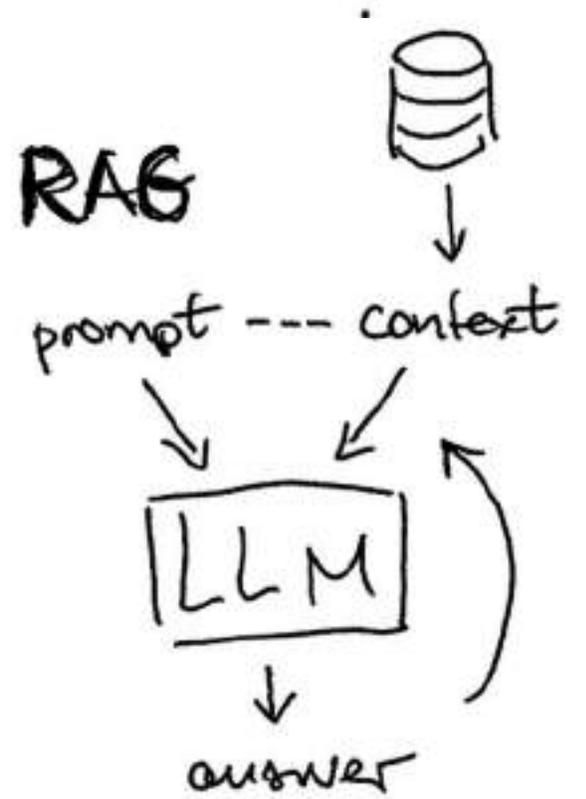
Deux types d'application de l'IA générative d'un grand intérêt pour les tests

**RAG** :  
Retrieval-  
Augmented  
Generation



# Applications RAG

Augmenter le contexte de la requête par l'extraction de sources documentaires



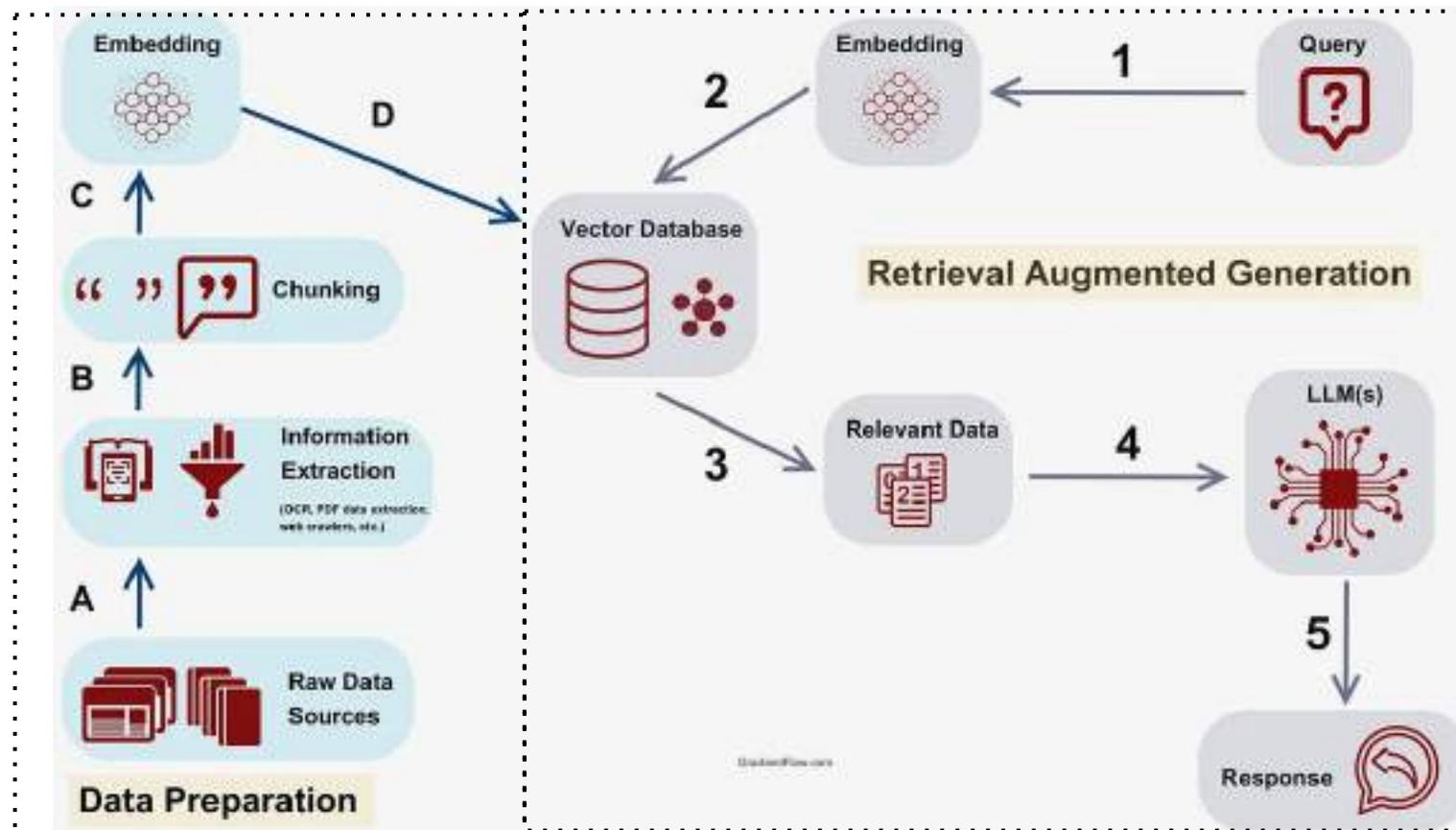
**RAG** :  
Retrieval-  
Augmented  
Generation

**Donner du contexte spécifique au projet / à l'organisation lors d'une tâche à réaliser via LLM**



# La technique RAG en un schéma

Phase 1 :  
Préparation  
des  
données



Phase 2 :  
Utilisation  
lors d'une  
requête à  
l'IA



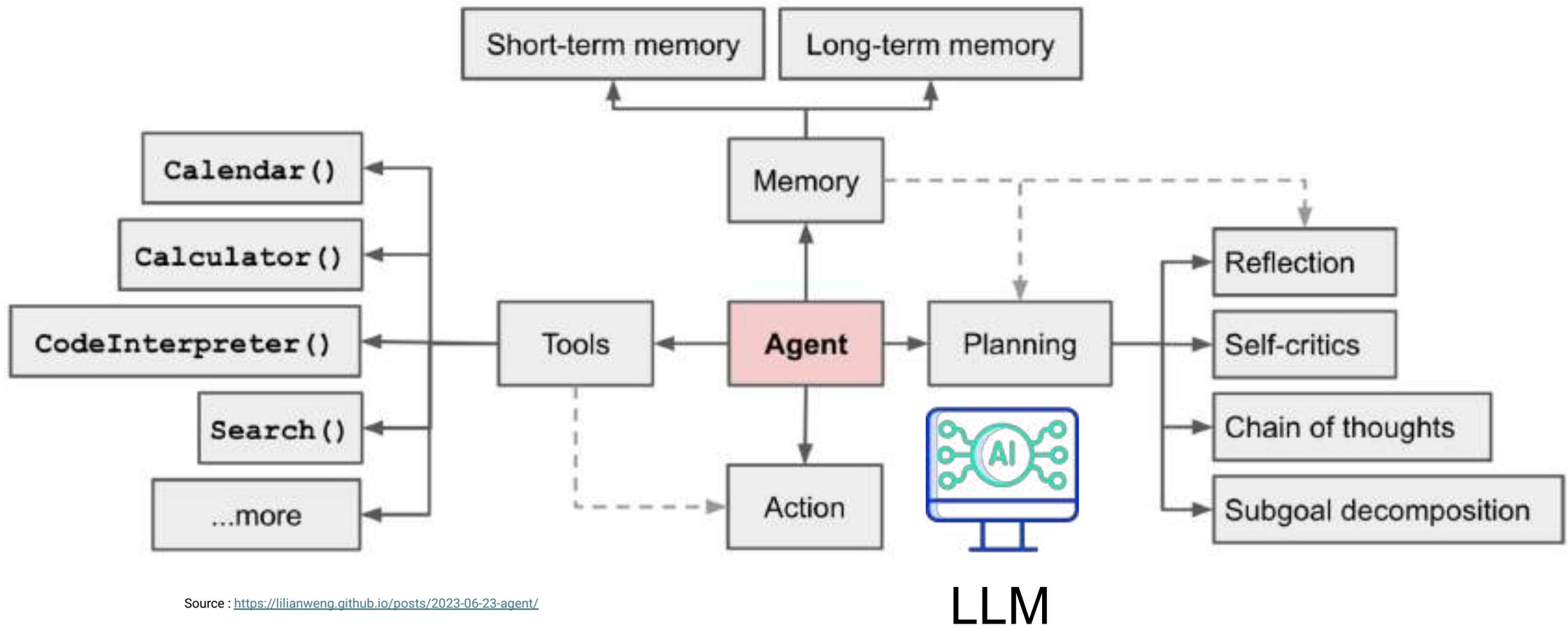
# Agents IA (basés sur l'IA générative)

Vers des agents testeurs intelligents et autonomes



Les agents basés sur l'IA générative visent à réaliser des tâches complexes de façon autonomes ou semi-autonomes

# Architecture Agent-LLM en un schéma

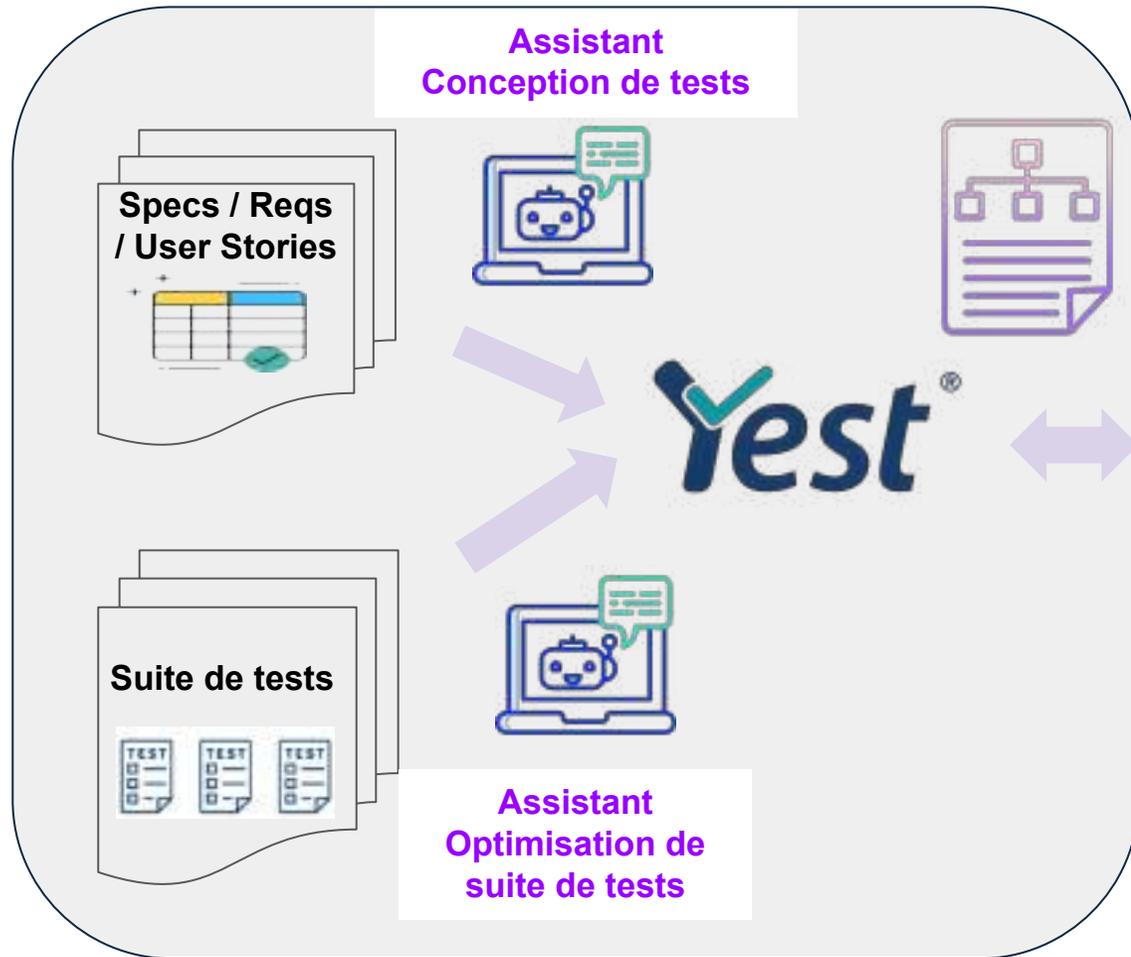


Source : <https://lilianweng.github.io/posts/2023-06-23-agent/>



# Testeur augmenté par l'IA

NÉE  
CAISE  
TESTS



Activités de conception et d'implémentation des tests



Testeur

Exemple :  
3 applications  
basées LLM  
intégrés au flot  
de travail du  
testeur

Activités d'exécution des tests et de reporting



Agent  
Testeur

Application



# Autres sujets

## Organiser l'usage de l'IA générative pour en obtenir les bénéfices

- 👉 LLMOps - Le DevOps des applications basées sur l'IA générative
- 👉 Le fine-tuning des modèles LLM : adapter un "petit" modèle pour une tâche spécifique
- 👉 Bibliothèques de prompt pour partager les bonnes pratiques de l'IA générative pour les tests
- 👉 Définir une stratégie et une feuille de route pour l'organisation de test (centre de services, centre de recette, ...)



# Stratégie / Feuille de route

Comment en pratique ?

**WEBINAIRE**

**IA GÉNÉRATIVE POUR LES TESTS :  
STRATÉGIE ET FEUILLE DE ROUTE**

**BRUNO LEGEARD**

**JEUDI 27 JUIN**  
12h15 (Heure Française)

Inscrivez-vous 



**Smartesting** x **UpSkill4IT**  
E-LEARNING & E-COACHING



# Quelques ressources

## Bases du Prompt Engineering

- [www.promptingguide.ai](http://www.promptingguide.ai)
  - Introduction au prompting
- Taxonomie des techniques de prompting
  - <https://arxiv.org/pdf/2402.07927.pdf>

## Aggrégateurs LLM

- POE - <https://poe.com/>
- Perplexity - <https://www.perplexity.ai>

## Utiliser un IDE de Prompt Engineering

- <https://platform.openai.com/playground>
  - L'interface de mise au point des Prompts GPT-x lorsque utilisé en mode API
- <https://console.anthropic.com/workbench>
  - L'interface de mise au points de Prompts de Anthropic
- <https://promptmetheus.com/>
  - Permet d'itérer et de qualifier vos prompts



# Bruno Legeard

IA Générative : Kit de  
survie pour les Testeurs

**11 JUIN 2024**  
BEFFROI DE MONTROUGE



JOURNÉE  
FRANÇAISE  
DES TESTS  
LOGICIELS