Introduction RAG

Milliman × Université Gustave Eiffel

4–5 juin 2026 · 36h · 2 jours

Milliman

Université Gustave Eiffel

Milliman

Du choix du modèle à l'intégration opérationnelle

Complexité et gouvernance croissantes

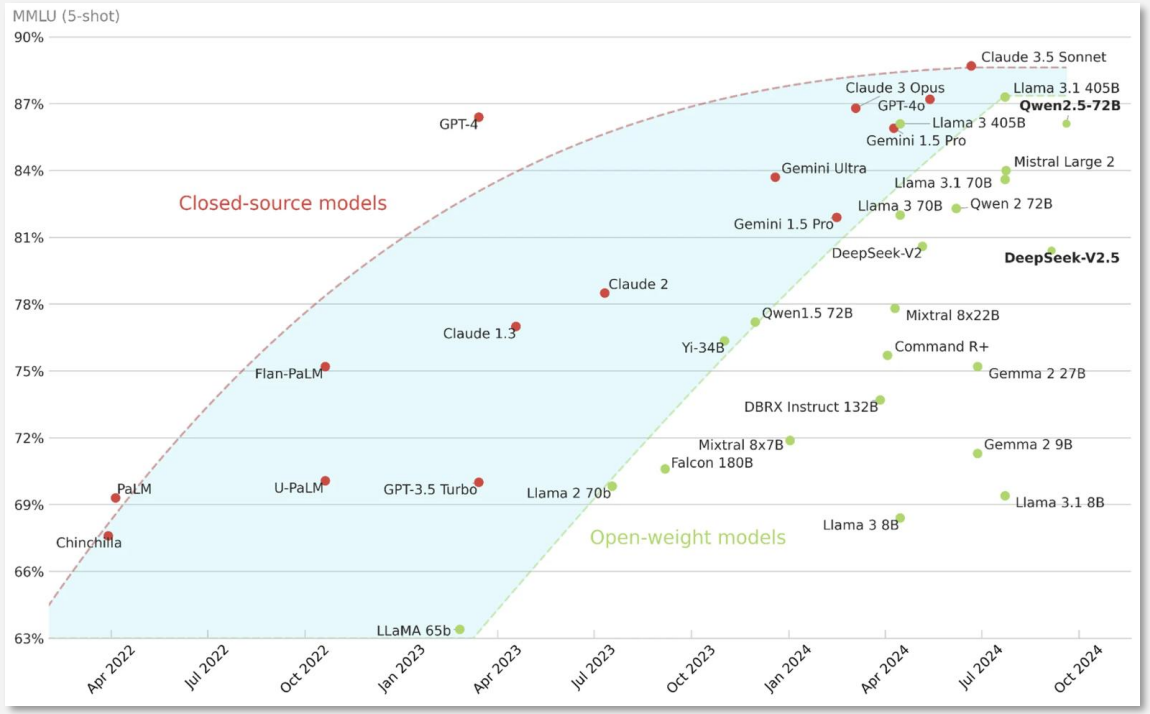


Figure 4. Comparative performance of closed-source and open-weight LLMs on the 5-shot MMLU benchmark, from (Riedemann et al., 2024)

Riedemann, L., Labonne, M., & Gilbert, S. (2024). The path forward for large language models in medicine is open. *npj Digital Medicine*, *7*(1), 339.

Choix du modèle LLM

Capacités brutes, taille, coût

Du choix du modèle à l'intégration opérationnelle

Complexité et gouvernance croissantes

COSTAR Framework

Context (Contexte)

Dans quel cadre s'applique le prompt ?

Objective (Objectif)

Que doit produire l'IA ?

Style (Style)

Comment doit être formulée la réponse ?

Tone (Ton)

Quel ton adopter ?

Audience (public)

À qui s'adresse la réponse ?

Response (format attendu)

Sous quelle forme livrer le résultat ?

Alignement au niveau prompt

Prompt engineering, exemples, instructions

Choix du modèle LLM

Capacités brutes, taille, coût

Du choix du modèle à l'intégration opérationnelle

Complexité et gouvernance croissantes

RAG (Retrieval-Augmented Generation)

Couple le LLM à une base documentaire interne : à chaque requête, le système **recherche les passages pertinents** et les fournit au modèle pour ancrer sa réponse.

- **Sans réentraîner** le modèle
- Réponses **traçables** et **actualisables**
- Réduit les **hallucinations**

Spécialisation au niveau modèle

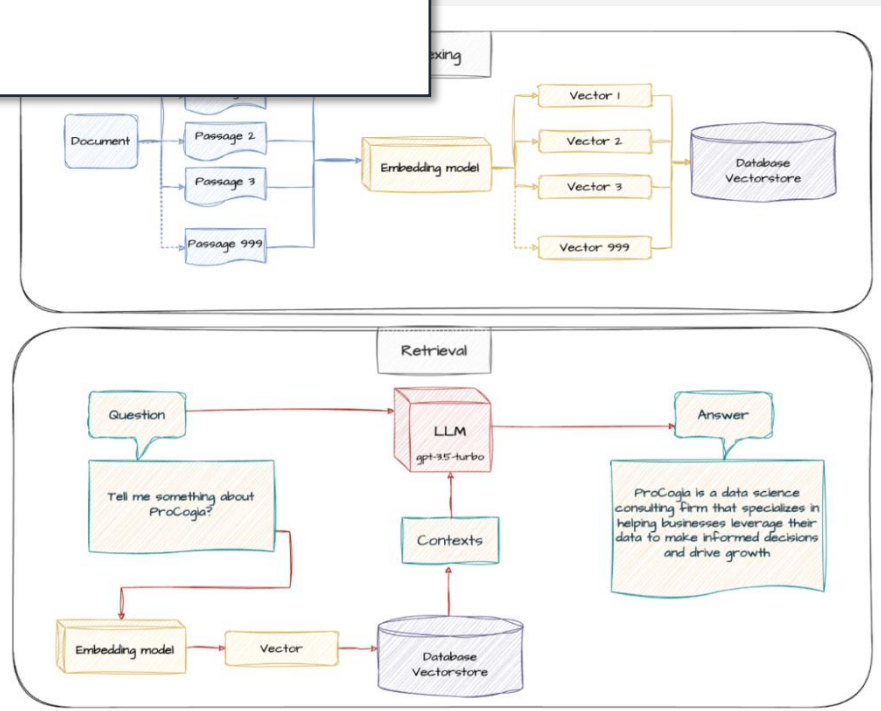
RAG, fine-tuning, adaptation au domaine, LoRA

Alignement au niveau prompt

Prompt engineering, exemples, instructions

Choix du modèle LLM

Capacités brutes, taille, coût



Du choix du modèle à l'intégration opérationnelle

Complexité et gouvernance croissantes

Multi-Agent

Characteristic	SAS	MAS (Independent)	MAS (Decentralized)	MAS (Centralized)	MAS (Hybrid)
Interaction Type	Agent	Agent Agent Agent → Aggregation	Agent Agent Agent → Majority Voting	Orchestrator → Sub-Agent Sub-Agent Sub-Agent	Orchestrator → Sub-Agent Sub-Agent Sub-Agent

Raisonnement au niveau système

Agents, planification, workflows multi-étapes

Spécialisation au niveau modèle

RAG, fine-tuning, adaptation au domaine, LoRA

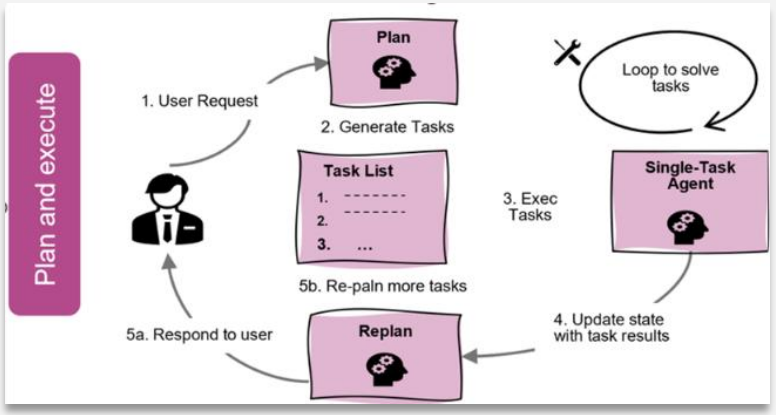
Alignement au niveau prompt

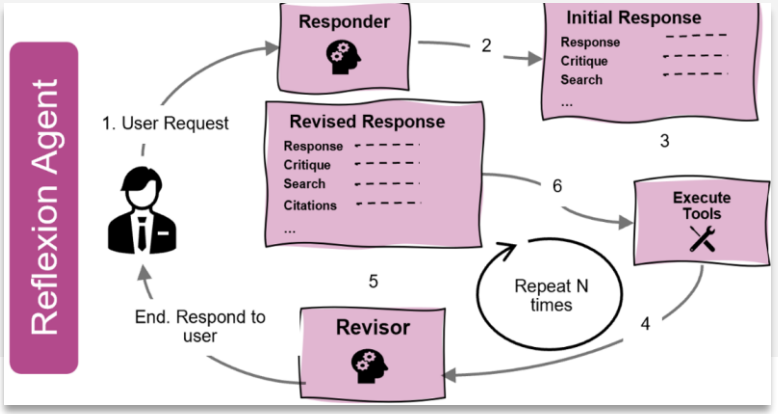
Prompt engineering, exemples, instructions

Choix du modèle LLM

Capacités brutes, taille, coût

Single Agent





Milliman

Le principe du RAG (1/2)

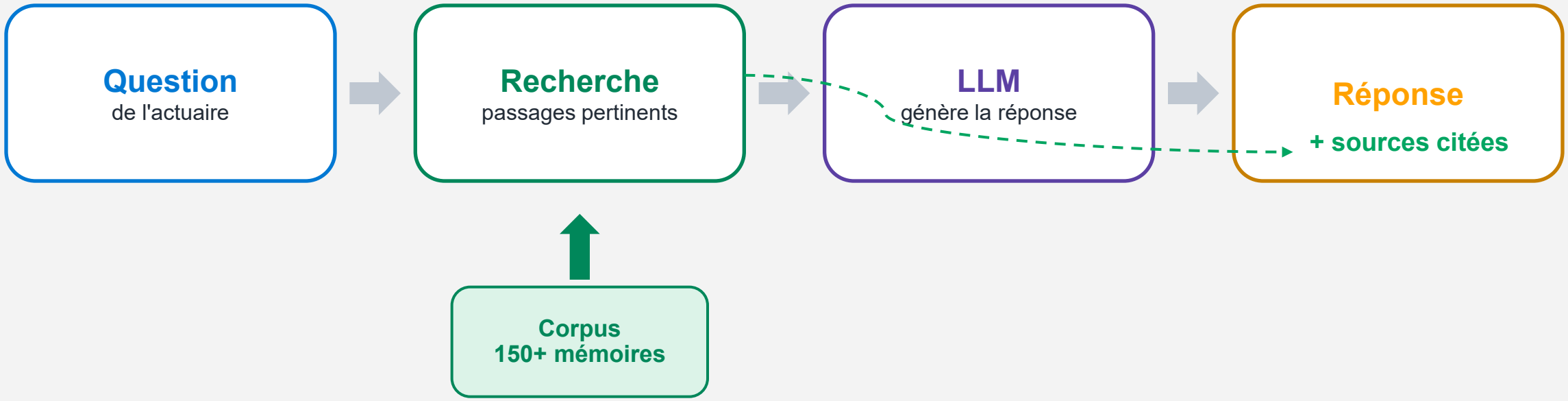
Chercher d'abord dans le corpus, répondre ensuite

Schéma classique d’une IA générative

Le principe du RAG (2/2)

Chercher d'abord dans le corpus, répondre ensuite

Schéma classique d'une IA générative … avec RAG

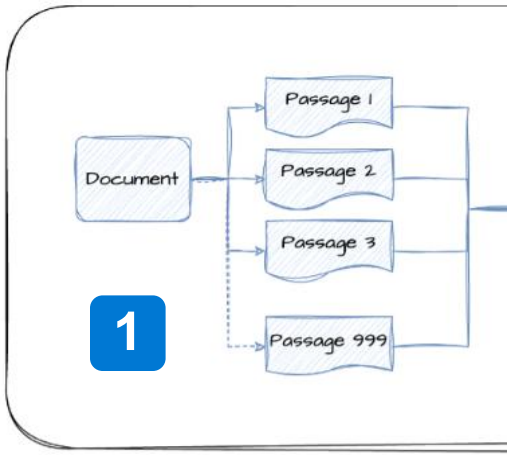


Comprendre le RAG, brique par brique

Transformer 150+ PDF en passages exploitables par le modèle

1

Ingestion & chunking



CE QUE C'EST

Extraire le texte des PDF et le découper en passages (chunks) de taille maîtrisée.

CE QUI COMPTE

Un bon découpage garde les idées ensemble : ni **trop gros** (bruit), ni **trop petit** (perte de contexte).

LEVIER COÛT / CARBONE

Des chunks bien dimensionnés = moins de passages à envoyer au LLM = moins de tokens, moins de coût et de CO_2.

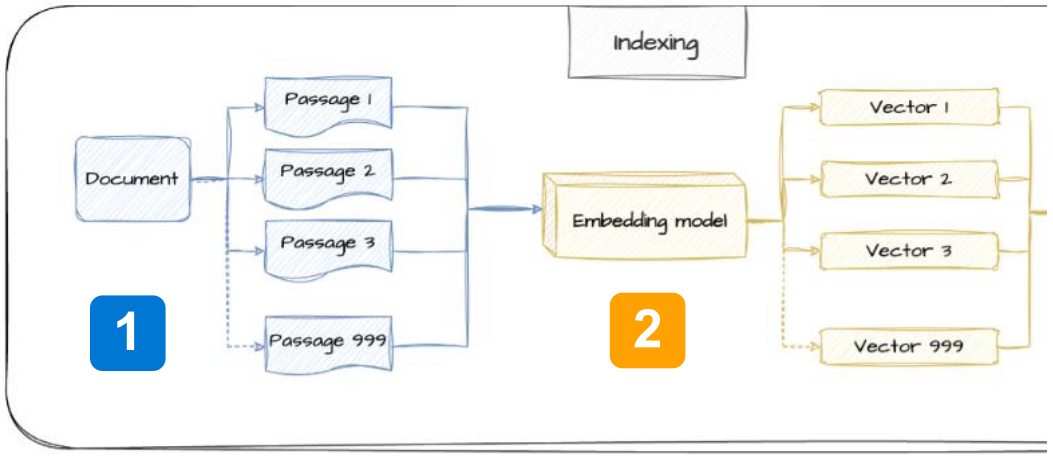
Comprendre le RAG, brique par brique

Le sens du texte, encodé en vecteurs

2

Embeddings

Transformer le texte en vecteurs numériques



CE QUE C'EST

Chaque passage devient un vecteur qui capture son sens, via un modèle d'embedding.

CE QUI COMPTE

Un bon modèle **rapproche les questions de leurs vrais passages** : meilleur rappel.

LEVIER COÛT / CARBONE

Embeddings calculés une seule fois (offline) : coût quasi nul ensuite.

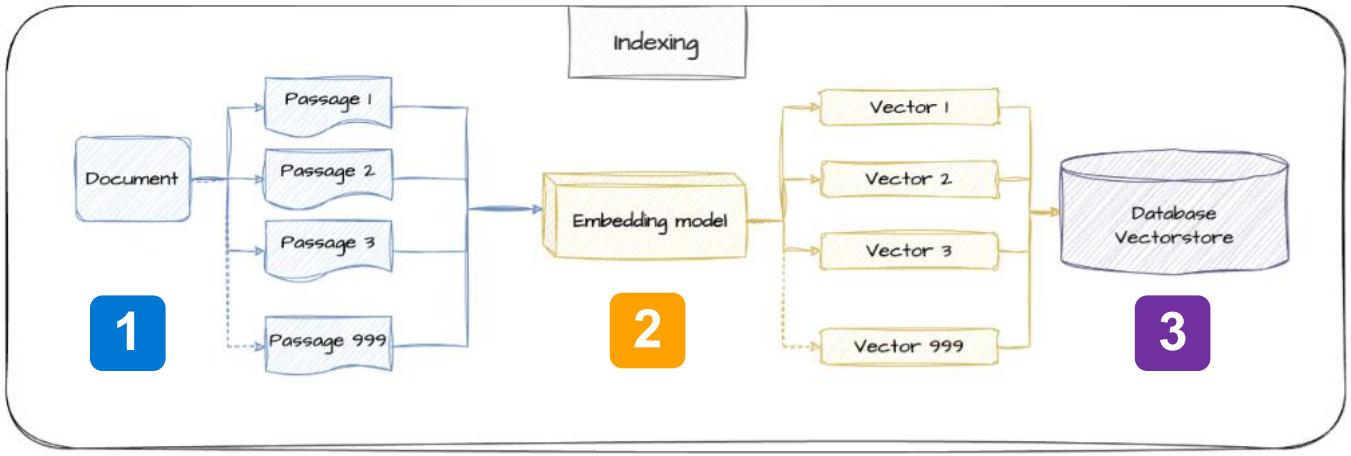
Comprendre le RAG, brique par brique

La mémoire qui retrouve les bons passages

3

Vector store

Stocker et indexer tous les vecteurs



CE QUE C'EST

Une base spécialisée (FAISS, Chroma) qui garde les vecteurs et les retrouve vite.

CE QUI COMPTE

La recherche par similarité doit **rester rapide** même sur 150+ mémoires entiers.

LEVIER COÛT / CARBONE

Un index efficace évite de tout parcourir : moins de calcul, moins d'énergie.

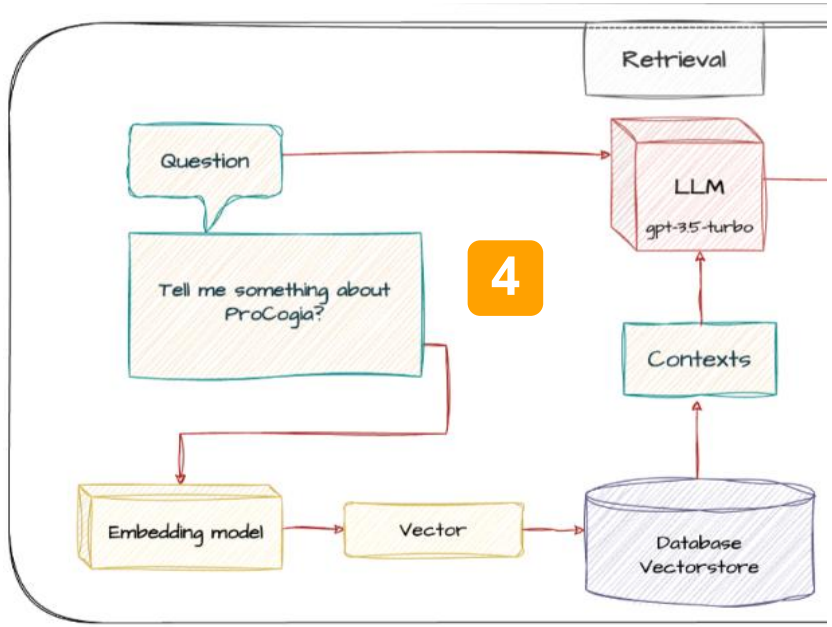
Comprendre le RAG, brique par brique

Le bon contexte, ni trop ni trop peu

4

Retrieval

Sélectionner les passages les plus pertinents



CE QUE C'EST

À la question, on encode et on récupère les k passages les plus proches.

CE QUI COMPTE

- **Trop peu** de passages : réponse incomplète.
- **Trop** : bruit et coût en hausse.

LEVIER COÛT / CARBONE

Bien régler k et filtrer : moins de tokens envoyés au LLM, donc moins cher et plus sobre.

Comprendre le RAG, brique par brique

La réponse, sourcée et maîtrisée

5

Génération

Le LLM rédige la réponse à partir des passages

CE QUE C'EST

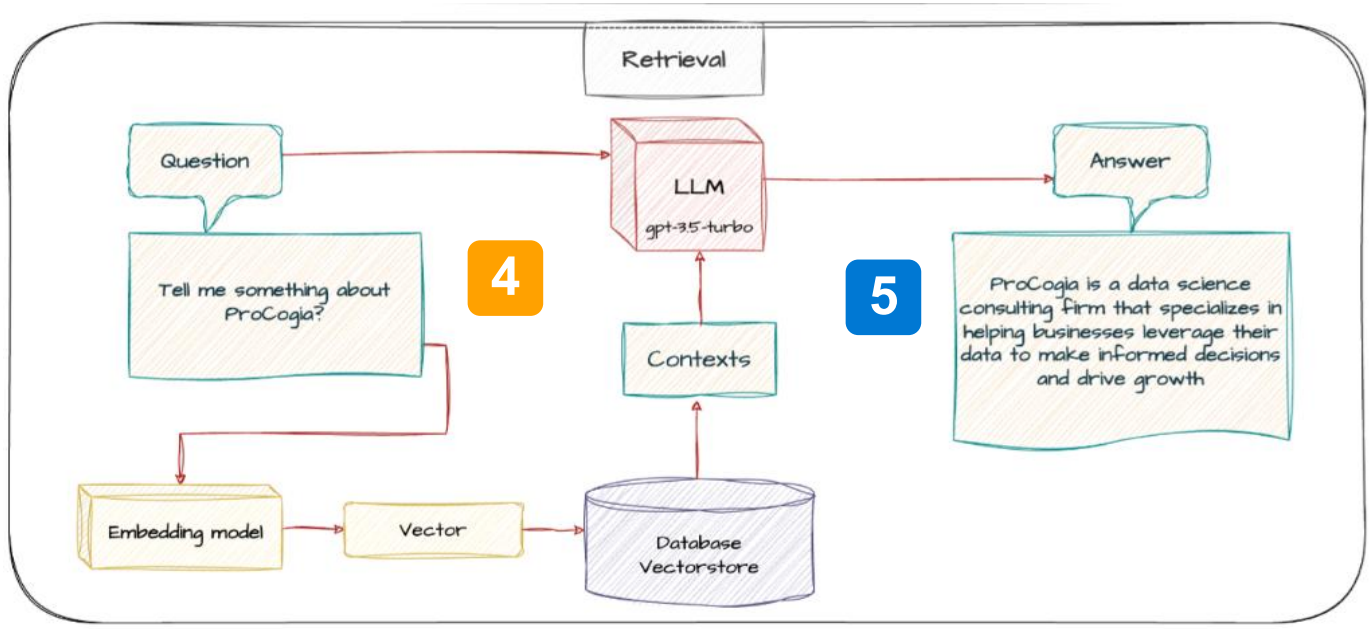
Le LLM reçoit la question + les passages et rédige une réponse sourcée.

CE QUI COMPTE

Un bon prompt **force à citer les sources** et à répondre « je ne sais pas » si besoin.

LEVIER COÛT / CARBONE

Router vers le petit modèle quand la question est simple : gros gain coût/carbone.



Architectures de RAG

Exemples d'architectures

Architecture naïve

- **Chunking fixe bien calibré** : le découpage est fixe
- **Recherche vectorielle seule** : le sens de la question est comparé aux chunks
- **Résultats bruts du retrieval :** on prend les k chunks les plus proches
- **Question utilisateur directe** : la question est envoyée au retriever sans modification
- **Pas de filtrage** : on cherche dans toute la base, sans distinction
- **Prompt Système bien rédigé** : on guide le LLM (par exemple: citer les sources)

Architecture avancée

- **Chunking sémantique par sections** : le découpage est fait selon la structure du document (titres, sections)
- **Recherche vectorielle + BM25** : on ajoute une recherche par mots-clés exacts pour les termes techniques
- **Reranking par CrossEncoder** : un second modèle reclasse les résultats par pertinence réelle
- **Query rewriting par le LLM** : le LLM reformule la question pour la rendre plus précise avant la recherche
- **Filtrage par métadonnées** : la recherche est restreinte à un chapitre, une source, un type de contenu
- **Evaluation (RAGAS)** : la qualité des réponses est mesurée (métriques de fidélité, pertinence, précision)

(Option) Pistes avancées

Des approches plus riches que le RAG classique : optionnelles

GraphRAG

RAG sur un graphe de connaissances

- Relie les concepts entre mémoires
- Bon pour les questions transverses
- Plus lourd à construire

RAG agentique

Le système raisonne en plusieurs étapes

- Décompose la question
- Cherche, vérifie, recommence
- Décide quand s'arrêter

LangGraph

Outil pour orchestrer ces workflows

- Enchaîne les étapes en graphe
- Gère mémoire et branchements
- Couche au-dessus de LangChain

Commencez par faire tourner la baseline : améliorez-la seulement après

C’est parti ! Optimiser les 3 critères à la fois

L'accuracy fait gagner, coût et carbone ne servent qu'à départager

Accuracy → PRIORITÉ N°1	Coût → DÉPARTAGE	Carbone → DÉPARTAGE
• Bon chunking & retrieval précis • Prompt clair, sources citées • Gros modèle sur cas difficiles	• Petit modèle sur les tâches simples • Embeddings mis en cache • Pas de retries inutiles • Prompts sans verbosité superflue	• Cache (corpus fixe) • Éviter les appels redondants • Réponses sans remplissage inutile • À accuracy égale seulement

Commencez par faire tourner la baseline : améliorez-la seulement après