**Note méthodologique : preuve de concept**

## **Dataset retenu**

**Le Code civil français** est un ensemble de lois qui encadre les relations entre les personnes en France. Il définit les règles sur :

* **La famille** (mariage, divorce, filiation),
* **Les contrats** (achats, locations, engagements),
* **La propriété** (achat, vente, héritage),
* **La responsabilité** (réparation des dommages).

C’est un texte fondamental, toujours en vigueur depuis 1804, qui influence le droit dans de nombreux pays. **Notre dataset retenu est ce Code civil**, car il reflète les principes juridiques essentiels qui régissent la société française. Il offre une base structurée pour analyser les droits et obligations des citoyens, ce qui en fait une source précieuse pour étudier le droit privé.

## **Les concepts de l’algorithme récent**

L'Agentic RAG (Retrieval-Augmented Generation) est une évolution avancée de la génération augmentée par récupération classique, intégrant l'autonomie d'agents d'IA intelligents dans le processus. Contrairement au RAG classique, où un modèle IA récupère des informations externes et génère une réponse en une seule étape, l'Agentic RAG utilise des agents autonomes capables d'effectuer plusieurs cycles d'amélioration, validation, et affinage de la réponse avant de la livrer. Cette approche favorise une interaction dynamique entre plusieurs modules spécialisés.

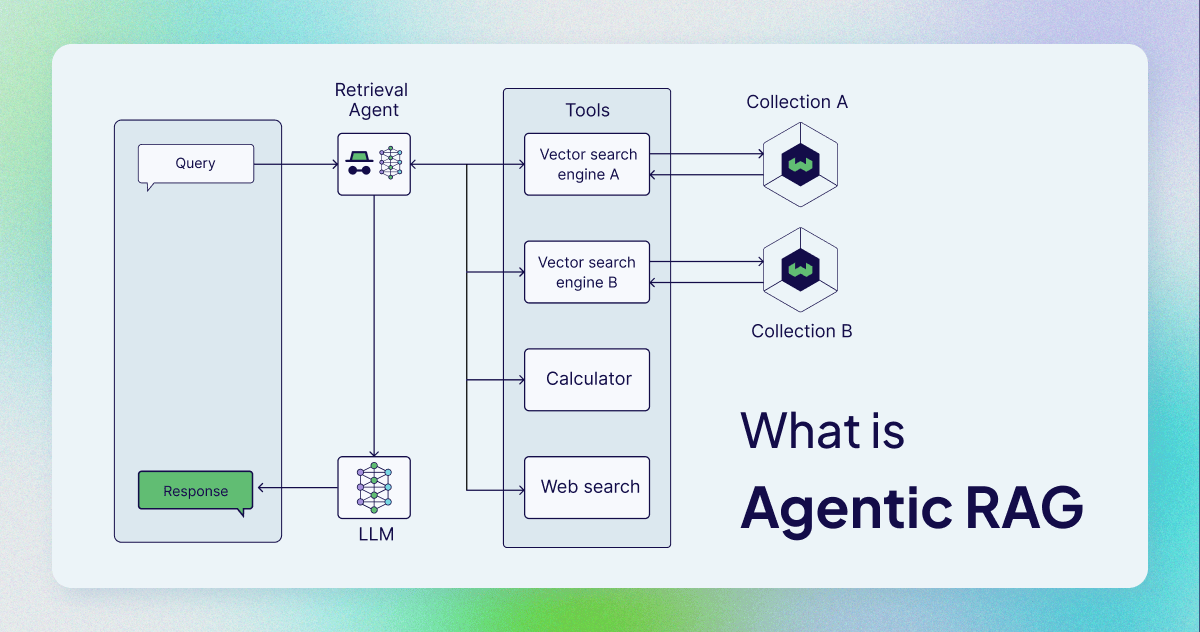


Figure 1- Agentic Rag Schema

**La modélisation**

La méthodologie présentée repose sur une approche hiérarchisée de recherche sémantique, conçue pour répondre de manière précise et efficace aux requêtes des utilisateurs. Ce processus s’inspire des techniques modernes de traitement automatique du langage (NLP) et d’intelligence artificielle, en exploitant des représentations vectorielles du texte, appelées *embeddings*. L’objectif principal est d’optimiser la pertinence des résultats tout en minimisant les ressources nécessaires, en adaptant la granularité de la recherche selon les besoins.

## Étape 1 : Réception et Reformulation de la Requête

Tout commence par la requête utilisateur, qui peut être une question, une demande d’information ou une recherche spécifique. Cette requête est ensuite reformulée pour en clarifier le sens et en extraire les éléments clés. La reformulation permet d’améliorer la compréhension du système et d’orienter la recherche vers les concepts les plus pertinents. Par exemple, une question vague comme *"Quels sont les droits en cas de divorce ?"* pourrait être reformulée en "Quels sont les articles du Code civil français relatifs à la répartition des biens lors d’un divorce ?". Cette étape est cruciale, car elle détermine la qualité des résultats obtenus par la suite.

## Étape 2 : Recherche Sémantique dans les Résumés

Le premier niveau de recherche s’effectue sur des embeddings de résumés. Ces embeddings sont des représentations vectorielles condensées des documents, qui capturent l’essentiel de leur contenu. Le système compare la requête reformulée à cette base de données de résumés pour identifier les documents potentiellement pertinents.

**Évaluation des résultats :** Si les résultats obtenus à ce stade sont jugés suffisants (c’est-à-dire qu’ils répondent de manière satisfaisante à la requête), le système génère directement une réponse et met fin au processus. Cette approche permet de gagner du temps et des ressources, car elle évite de traiter des données plus détaillées inutiles.En revanche, si les résultats sont insuffisants, le système passe au niveau de recherche suivant.

## Étape 3 : Recherche Sémantique dans les Représentations Visuelles

Lorsque les résumés ne suffisent pas, la recherche s’étend aux embeddings de représentations visuelles. Ces embeddings correspondent à des éléments visuels ou structurés associés aux documents, comme des tableaux, des graphiques ou des schémas. Cette étape permet de capturer des informations qui ne sont pas nécessairement présentes dans les résumés textuels, mais qui peuvent être essentielles pour répondre à la requête.

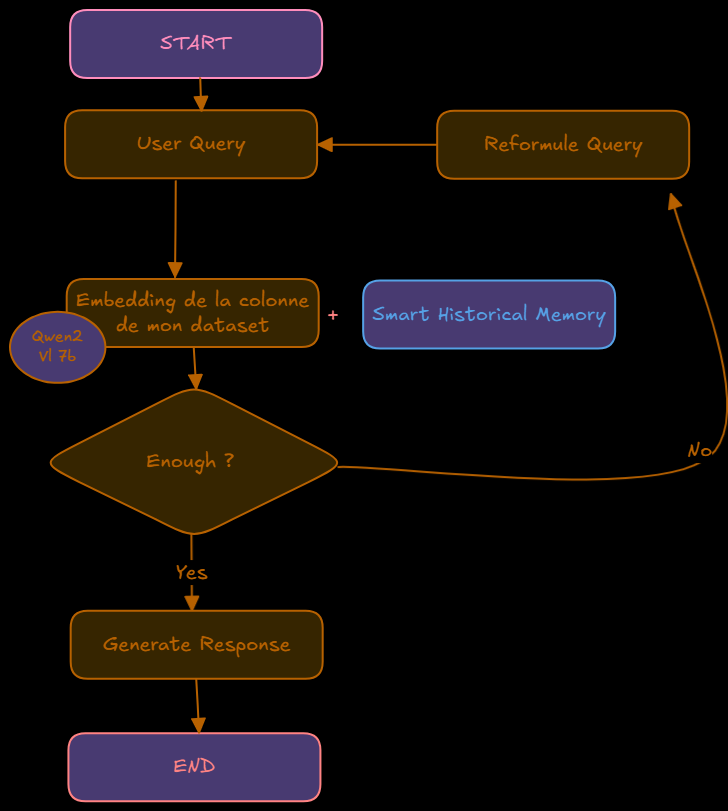
Une étape de fusion des résultats ("Collapse") peut être appliquée pour regrouper et synthétiser les informations pertinentes issues de cette recherche.Si les résultats sont cette fois suffisants, le système génère une réponse et termine le processus. Sinon, il passe au dernier niveau de recherche.

## Étape 4 : Recherche Sémantique sur les Pages Complètes

En dernier recours, la recherche est effectuée sur des embeddings de pages complètes. Ces embeddings représentent le contenu intégral et détaillé des documents, offrant une couverture exhaustive des informations disponibles. Cette étape est la plus coûteuse en termes de ressources, mais elle garantit une réponse précise et complète.Cela consiste à segmenter les documents en parties plus petites pour affiner la recherche et identifier les passages les plus pertinents.Dès que des résultats satisfaisants sont obtenus, le système génère une réponse et met fin au processus.

## Étape 5 : Génération de la Réponse et Conclusion

Une fois qu’un niveau de recherche a fourni des résultats jugés pertinents, le système synthétise ces informations pour produire une réponse claire et adaptée à la requête initiale. Cette réponse est ensuite retournée à l’utilisateur, marquant la fin du processus ("END").



**Avantages de cette Méthodologie**

Cette approche présente plusieurs atouts majeurs :

* **Efficacité** : En commençant par des recherches légères (résumés) avant de passer à des analyses plus approfondies, le système optimise l’utilisation des ressources.
* **Précision** : La recherche multi-niveaux permet d’affiner progressivement les résultats, garantissant une réponse pertinente même pour des requêtes complexes.
* **Flexibilité** : La reformulation de la requête et les étapes de fusion ("Collapse") ou de découpage ("Chunking") permettent de s’adapter à une grande variété de demandes.

**Une synthèse des résultats**

*Présentez une synthèse des résultats comparés entre la technique récente et les techniques utilisées précédemment et une conclusion, en 2 pages maximum.*

**Les limites et les améliorations possibles**

Pour optimiser au maximum les performances de mon système d’Agentic RAG, plusieurs axes d’amélioration peuvent être envisagés en parallèle : tout d’abord, migrer vers une base de données vectorielle plus performante (comme Qdrant, Weaviate ou Milvus) permettrait d’accélérer les recherches de similarité tout en améliorant la précision et la scalabilité. Ensuite, l’implémentation d’un système de retrieval plus intelligent — par exemple via la ré-évaluation des résultats (re-ranking), la recherche hybride (dense + sparse) ou l’ajout de métadonnées contextuelles — renforcerait la pertinence des documents récupérés. Par ailleurs, repenser l’architecture globale du RAG en adoptant des variantes avancées comme le RAG hybride, le RAG itératif ou le RAG avec feedback en boucle fermée permettrait d’adapter dynamiquement le système aux besoins complexes des requêtes. Le chunking lui-même peut être optimisé : passage à un découpage sémantique (basé sur les phrases ou les idées) plutôt que purement textuel, utilisation de fenêtres glissantes ou de techniques comme Semantic Chunking ou LLM-guided chunking, pour préserver le contexte et la cohérence. Enfin, une meilleure gestion de la mémoire — via des mécanismes de mémoire à long terme, de cache contextuel, ou d’historique dynamique des conversations — permettrait au système de maintenir une cohérence sur le long terme, d’éviter les redondances et d’exploiter les interactions passées pour affiner ses réponses. Ensemble, ces améliorations transforment un RAG basique en un système robuste, précis et conversationnellement fluide.ci dessous un exemple d’amelioration du système d’agentic RAG basé sur celui de la POC qui permet d’etre assez large et robuste pour tout type de documents.

