Centauri — Studio IA souverain

Réussir un projet d'agent IA et de RAG

Le cadre pour passer du chatbot générique à l'agent métier connecté à vos données

Type	**Cadre méthode**
Édition	Édition 2026
Pour	PME et agences visant un agent métier

Centauri

Agent IA et RAG sur-mesure

SOCIÉTÉ CENTAURI

Sommaire

Le problème des agents génériques

Pourquoi un chatbot no-code ne suffit pas

Beaucoup de projets d'IA conversationnelle s'arrêtent à un chatbot générique : il répond avec des connaissances générales, mais il ne connaît pas votre métier, il invente quand il ne sait pas, et il n'a ni mémoire ni isolation de vos données. Pour une PME ou une agence, ce n'est pas un outil de travail fiable.

Un **agent métier** est différent : il est connecté à vos données, il s'appuie sur une mémoire vérifiable, il exécute des actions encadrées, et il maintient une séparation stricte entre les données de projets différents. Ce cadre décrit comment passer de l'un à l'autre.

> *La différence n'est pas la « puissance du modèle ». C'est l'architecture autour du modèle : récupération de l'information, mémoire, isolation, supervision.*

Les quatre piliers d'un agent fiable

- **RAG** (Retrieval-Augmented Generation) : l'agent récupère l'information pertinente dans vos documents avant de répondre.
- **Mémoire** : il conserve un contexte persistant entre les sessions.
- **Isolation** : les données d'un projet ne fuient pas vers un autre.
- **Supervision** : les actions à conséquence passent par une validation humaine.

Comprendre le RAG

Ce que résout la récupération d'information

Un modèle de langage seul répond avec ce qu'il a appris ; il ne connaît pas vos documents internes et peut inventer. Le RAG corrige cela : avant de répondre, l'agent va chercher les passages pertinents dans votre base documentaire, puis répond en s'appuyant dessus. Le résultat est une réponse ancrée dans vos données, et traçable.

RAG hybride : dense et lexical

Un RAG robuste combine deux modes de recherche :

- **Recherche dense** (vectorielle) : retrouve les passages sémantiquement proches, même formulés différemment.
- **Recherche lexicale** (plein texte) : retrouve les correspondances de mots exacts, y compris les termes métier précis.

Les deux sont ensuite fusionnés pour ne garder que les meilleurs résultats. C'est l'approche que nous opérons en production : un RAG hybride combinant recherche vectorielle et recherche plein texte en français, avec une fusion des résultats, livré avec une suite de tests entièrement verte et une revue de code sans défaut bloquant. Cette combinaison rattrape les correspondances purement lexicales que la recherche dense seule manquait.

> *Une base de connaissance interne indexée peut représenter plusieurs milliers de documents et dizaines de milliers de fragments. L'enjeu n'est pas seulement de stocker, mais de retrouver le bon passage au bon moment.*

Connecter l'agent au métier

Exposer la logique métier comme des outils

Un agent utile ne fait pas que répondre : il agit. Pour cela, on expose la logique de votre application sous forme d'outils que l'agent peut appeler de façon contrôlée (selon le protocole MCP, Model Context Protocol). Chaque outil a un périmètre défini et des règles d'usage.

Nous avons par exemple exposé près de deux cents outils métier à un agent dans un projet réel, ce qui lui permet d'opérer sur l'application plutôt que de se contenter de discuter.

Exécution asynchrone et encadrée

Un agent de production ne s'exécute pas « en direct » sans filet. Le cadre éprouvé :

- **File d'attente** : les demandes sont mises en file et traitées de façon ordonnée.
- **Idempotence** : une demande rejouée ne produit pas d'effet en double.
- **Reprise sur erreur** : les échecs sont réessayés selon une politique définie.
- **Approbation humaine** : les actions sensibles attendent une validation.
- **Isolation stricte** : chaque projet a ses données cloisonnées, sans fuite vers les autres.

Les étapes d’un projet réussi

De l’idée à la production

Phase	Objectif	Livrable
1. Cadrage	Définir le périmètre et les cas d’usage	Spécification approuvée
2. Base de connaissance	Rassembler et indexer vos documents	Index RAG hybride
3. Outils métier	Exposer les actions utiles	Serveur d’outils (MCP)
4. Agent	Assembler récupération, mémoire, outils	Agent supervisé
5. Intégrations	Relier aux canaux (messagerie, workflows, webhooks)	Connexions opérationnelles
6. Exploitation	Surveiller, mesurer, ajuster	Run mensuel

La règle d’or : spec d’abord

Pas de développement sans spécification approuvée. Chaque projet vit dans une structure documentée, avec des critères d’acceptation explicites et des étapes vérifiables. C’est ce cadrage qui évite l’effet tunnel et garantit que l’agent livré correspond au besoin réel.

Anticiper les pièges

Ce qu'il faut vérifier dès le départ

- ☐ La qualité et la fraîcheur des documents indexés (un RAG ne vaut que sa base).
- ☐ La gestion des cas où l'agent ne sait pas : il doit le dire, pas inventer.
- ☐ L'isolation des données entre clients ou projets.
- ☐ Le périmètre des outils : que peut faire l'agent, et que ne peut-il pas faire.
- ☐ La supervision : qui valide quoi, et à quel moment.
- ☐ Le coût d'exploitation, suivi dans la durée.

Ce que nous ne survendons pas

Un projet d'agent IA et de RAG est un projet d'ingénierie, pas un branchement instantané. Le temps de mise en place dépend de la richesse de votre base documentaire et du nombre d'outils à exposer. Nous le disons d'entrée, et nous cadrons en conséquence.

Travailler avec Centauri

Centauri conçoit des agents métier sur-mesure : serveur d'outils exposant votre logique, mémoire vectorielle persistante avec RAG hybride, agent autonome par projet avec isolation stricte et exécution asynchrone, intégrations aux canaux que vous utilisez. Notre format type : un forfait de mise en place, puis un abonnement d'exploitation (run, hébergement, maintenance).

Nous opérons ces briques en production sur notre propre infrastructure — c'est cette expérience, pas une démonstration, que nous mettons à votre service.

Pour échanger sur votre cas : projetcentauri.com/contact.