

Charte de Projet : Plateforme d'Agents IA "Talaria"

Version : 2.0

Date : 24 septembre 2025

Statut : Document de Cadrage Stratégique

Synthèse (Le Projet en Bref)

- **Pourquoi (La Mission)** : Démocratiser l'automatisation des processus métier complexes pour les PME/ETI en transformant l'expertise humaine en actifs numériques fiables, sécurisés et scalables.
- **Quoi (Le Produit)** : Une plateforme SaaS ("Caméléon") permettant d'exécuter des "Recettes" d'orchestration d'agents IA. Ces recettes, co-construites avec des experts métier, automatisent des workflows à haute valeur ajoutée.
- **Comment (La Stratégie)** : Une approche progressive en 3 phases (Service > Produit > Plateforme) où chaque étape finance la suivante. Les recettes sont créées en partenariat avec des experts via un modèle de partage de revenus (royalties).
- **Pour Qui (Les Cibles)** : Les PME et ETI dans des secteurs à forte intensité de processus et réglementaire (ex: fiscalité, logistique, juridique), qui cherchent des gains de productivité et de fiabilité.
- **Qui (L'Équipe)** : Un fondateur ("Architecte de Processus IA") qui collabore avec un écosystème d'experts métier pour chaque vertical.
- **Pour Quoi (Les Bénéfices)** : Fournir des résultats métier quantifiables : réduction des coûts opérationnels, diminution drastique des erreurs, conformité assurée, et libération du temps des experts pour des tâches à plus haute valeur stratégique.

1. Vision & Positionnement Différenciant

1.1. La Vision Long Terme

Devenir la plateforme de référence pour les PME/ETI qui souhaitent opérer et gouverner des systèmes d'agents autonomes. "Caméléon" n'est pas un simple outil d'automatisation, mais un véritable **système d'exploitation pour l'intelligence artificielle opérationnelle**, garantissant contrôle, sécurité et performance.

1.2. Le Positionnement Stratégique : Notre "Moat"

Notre avantage concurrentiel défendable ne réside **pas** dans la technologie sous-jacente (les LLM et les frameworks d'orchestration sont en voie de commoditisation), mais dans notre **Bibliothèque de "Recettes" d'Orchestration**.

Le véritable actif de "Caméléon" est sa collection croissante de modèles de workflows d'agents pré-construits, testés en production et optimisés pour des cas d'usage verticaux. Chaque "recette" est une **encapsulation de propriété intellectuelle métier**, combinant la bonne séquence d'agents, les bons outils et les bons prompts pour résoudre un problème commercial spécifique. **Nous ne vendons pas de l'IA, nous vendons de l'expertise métier automatisée.**

2. Stratégies de Création et de Déploiement

2.1. Stratégie de Création des Recettes : Le Modèle Partenarial

Les recettes sont le cœur de notre valeur. Leur création est un processus collaboratif :

1. **Identification d'un Vertical** : Nous ciblons un secteur avec un processus complexe et coûteux.
2. **Partenariat avec un Expert** : Nous nous associons avec un expert reconnu du domaine (ex: avocat fiscaliste, logisticien).
3. **Co-construction** : L'expert apporte son savoir-faire et sa connaissance des cas limites ; nous apportons l'architecture et l'ingénierie IA.
4. **Partage de la Valeur** : L'expert n'est pas rémunéré pour son temps, mais reçoit des **royalties (15-30%)** sur les revenus récurrents générés par la vente de "sa" recette. Il est ainsi directement intéressé au succès et à la qualité du produit.

2.2. Stratégie de Déploiement en 3 Phases

Cette approche permet de construire le projet sur des bases solides, en validant chaque étape avec des revenus réels.

- **Phase 1 : Service "Tête de Pont" (Mois 0-12)**
 - **Offre** : Un projet de service sur-mesure pour co-créer la première recette avec un client-partenaire.
 - **Modèle Éco** : Frais de projet (one-shot) + Frais de service (récurrents).
 - **Architecture** : Déploiement mono-client (un serveur par client) pour minimiser la complexité.
- **Phase 2 : Produit "Recette sous Licence" (Mois 12-24)**
 - **Offre** : Vente de la recette déjà éprouvée comme une solution "clé en main".

- **Modèle Éco** : Frais d'installation + Abonnement mensuel (MRR).
- **Architecture** : Déploiements mono-client automatisés.
- **Phase 3 : Plateforme SaaS "Caméléon" (Mois 24+)**
 - **Offre** : Accès par abonnement à la plateforme et à sa bibliothèque de recettes.
 - **Modèle Éco** : Modèle SaaS multi-paliers.
 - **Architecture** : Infrastructure SaaS multi-tenant complète.

3. Architecture Technique de Référence

L'architecture est conçue pour la modularité, la robustesse et la scalabilité. Elle sépare clairement le moteur, les capacités et la logique métier.

3.1. La Stack Technologique Fondamentale

- **Framework Backend** : **FastAPI** (Python) pour sa performance, son écosystème et sa nature asynchrone.
- **Base de Données** : **PostgreSQL** pour sa robustesse, avec l'extension **pgvector** pour la recherche sémantique.
- **Moteur de Tâches Asynchrones** : **Celery** avec un broker (ex: RabbitMQ ou Redis) pour gérer les workflows longs et les traitements en arrière-plan (ingestion de documents, exécution d'agents...).
- **Moteur d'Orchestration d'Agents** : **LangGraph** (Python) pour modéliser les "recettes" sous forme de graphes d'états, garantissant une logique complexe, une gestion fine des erreurs et une excellente observabilité.
- **Déploiement (Phase 1)** : **Docker** sur un serveur privé virtuel (VPS) chez un fournisseur européen (OVHcloud, Scaleway).

3.2. L'Architecture Conceptuelle "Moteur / Outils / Recettes"

1. **Le Cœur / Moteur (Votre code core)** :
 - C'est la fondation robuste et générique que vous avez déjà construite.
 - **Responsabilités** : Gestion des API (FastAPI), authentification (Auth), gestion des clients (Tenants), accès à la base de données (SQLAlchemy), sécurité, logging.
 - **Nouveau composant central** : Le **Moteur d'Orchestration** qui intègre LangGraph. Son rôle est de charger une recette et d'exécuter le graphe en appelant les bons outils.

2. La Boîte à Outils (Agents Spécialisés) :

- Ce sont des **modules Python indépendants et réutilisables**. Chaque module expose une ou plusieurs fonctions ("outils") qu'un agent peut appeler.
- **Exemples d'Outils :**
 - outil_recherche_documentaire (votre RAG actuel, refactorisé).
 - outil_classification_document (utilise un LLM pour étiqueter un fichier).
 - outil_extraction_donnees_json (utilise un LLM pour extraire des données structurées).
 - outil_calcul_complexe (utilise un LLM de raisonnement pour la logique métier).
 - outil_api_externe (un connecteur pour interagir avec une API tierce, ex: une API météo).
- Ces outils sont la seule partie qui interagit avec les LLM ou les API externes.

3. La Bibliothèque de Recettes (Votre Propriété Intellectuelle) :

- Ce ne sont **PAS** des fichiers Python. Ce sont des **fichiers de configuration déclaratifs** (ex: JSON ou YAML).
- Chaque fichier décrit un graphe LangGraph. Il définit la séquence des nœuds, les transitions conditionnelles entre eux, et quel "Outil" chaque nœud doit exécuter.
- **Exemple (extrait de recette_fiscale.yaml):**

```
graph:  
  
  start: classifier_docs  
  
  nodes:  
    classifier_docs:  
      tool: outil_classification_document  
      transitions:  
        - target: extracteur_donnees  
    extracteur_donnees:  
      tool: outil_extraction_donnees_json  
      transitions:  
        - target: recherche_regles  
    # ... etc.
```

- Cette séparation garantit que votre expertise métier (les recettes) est dissociée de votre code technique, ce qui la rend plus facile à gérer, versionner et protéger.

