

# **Cahier des Charges préliminaire pour Simbaw ERP**

**31/03/2025 par Expert ETOUNDI Simon Pierre Kisito**



## Table des matières

|         |                                                                       |    |
|---------|-----------------------------------------------------------------------|----|
| 1.      | CAHIER DES CHARGES PRELIMINAIRE                                       | 4  |
| 1.1     | Introduction                                                          | 4  |
| 1.1.1   | Contexte et Objectifs                                                 | 4  |
| 1.1.2   | Portée du Projet                                                      | 4  |
| 1.2     | Besoins Fonctionnels                                                  | 4  |
| 1.2.1   | Module de Nomenclature                                                | 4  |
| 1.2.2   | Module de Budget                                                      | 5  |
| 1.2.3   | Module de Dépenses                                                    | 5  |
| 1.2.4   | Module de Recettes                                                    | 5  |
| 1.2.5   | Module de Trésorerie                                                  | 6  |
| 1.2.6   | Module de Comptabilité                                                | 6  |
| 1.2.7   | Module de Pilotage                                                    | 6  |
| 1.2.8   | Module de Gestion de la Solde et des Ressources Humaines              | 7  |
| 1.2.9   | Module de Gestion du Patrimoine                                       | 7  |
| 1.2.10  | Module de Gestion des Marchés Publics                                 | 7  |
| 1.2.11  | Module de Fiscalité Locale et Recouvrement                            | 7  |
| 1.2.12  | Module de Décentralisation et Transfert de Compétences                | 8  |
| 1.2.13  | Module de Gouvernance, Conformité et Reporting Stratégique            | 8  |
| 1.2.14  | Module de Communication et Gestion Documentaire                       | 8  |
| 1.2.15  | Module d'Intelligence Artificielle Transverse et LLM                  | 8  |
| 1.2.16  | Module Infrastructure Sécurisée et Interopérable                      | 9  |
| 1.3     | Spécifications Fonctionnelles et Non-Fonctionnelles                   | 9  |
| 1.3.1   | Spécifications Fonctionnelles                                         | 9  |
| 1.3.2   | Spécifications Non-Fonctionnelles (Besoins Techniques)                | 10 |
| 1.4     | Plan de Développement sur 6 Mois                                      | 11 |
| 1.4.1   | Objectifs Généraux                                                    | 11 |
| 1.4.2   | Calendrier et Phases de Développement                                 | 12 |
| 1.4.2.1 | Mois 1 : Phase de Planification et Conception                         | 12 |
| 1.4.2.2 | Mois 2 : Développement des Modules Fondamentaux                       | 12 |
| 1.4.2.3 | Mois 3 : Développement des Modules Financiers et Comptables           | 13 |
| 1.4.2.4 | Mois 4 : Développement des Modules RH, Patrimoine et Marchés Publics  | 13 |
| 1.4.2.5 | Mois 5 : Développement des Modules de Gouvernance, Communication & IA | 13 |
| 1.4.2.6 | Mois 6 : Intégration, Tests, Déploiement et Recette                   | 14 |
| 1.4.3   | Diagramme de Gantt (Planification sur 6 mois)                         | 15 |
| 1.4.4   | Conclusion (plan de développement)                                    | 16 |
| 1.5     | Conclusion                                                            | 16 |
| 2.      | ANNEXES TECHNIQUES                                                    | 17 |
| 2.1     | Répartition des modules de Simbaw ERP                                 | 17 |
| 2.1.1   | Modules en Python (FastAPI)                                           | 17 |
| 2.1.2   | Modules en Java Spring Boot                                           | 18 |
| 2.1.3   | Synthèse de la Répartition                                            | 19 |

|       |                                                     |    |
|-------|-----------------------------------------------------|----|
| 2.2   | Wireframe pour le dashboard principal de Simbaw ERP | 20 |
| 2.3   | Cartographies et architectures                      | 22 |
| 2.3.1 | Cartographie Fonctionnelle Complète                 | 22 |
| 2.3.2 | Cartographie Applicative Complète                   | 23 |
| 2.3.3 | Architecture Fonctionnelle Complète                 | 25 |
| 2.3.4 | Architecture Applicative Complète                   | 26 |

# 1. Cahier des Charges préliminaire

## 1.1 Introduction

### 1.1.1 Contexte et Objectifs

Simbaw ERP est une solution ERP de gestion publique conçue pour répondre aux exigences des administrations (Ministère des Finances et Collectivités Territoriales) en matière de gestion budgétaire, de dépenses, de recettes, de trésorerie, de comptabilité, de paie, de gestion du patrimoine, de marchés publics, de fiscalité, de décentralisation, de gouvernance et de communication. Le système intègre également une couche d'intelligence artificielle (IA) et un Large Langage Model (LLM) afin d'offrir un support décisionnel avancé.

### 1.1.2 Portée du Projet

Ce cahier des charges couvre l'ensemble des fonctionnalités attendues, la modélisation des données, ainsi que les exigences techniques et non-fonctionnelles afin de garantir une solution performante, sécurisée, évolutive et interopérable.

## 1.2 Besoins Fonctionnels

Les besoins fonctionnels se déclinent par module. Chacun définit les processus métier à couvrir.

### 1.2.1 Module de Nomenclature

- **Configuration et Mise à jour de l'Organigramme :**
  - Création et hiérarchisation des entités (Ministère, CTD, Services, Départements)
  - Attribution des rôles et responsabilités
- **Définition des Comptes Budgétaires :**
  - Classification et hiérarchisation (classe, chapitre, article, paragraphe)
  - Paramétrage adaptatif aux exigences réglementaires
- **Élaboration des Programmes et Projets :**
  - Définition des programmes budgétaires et des projets d'investissement
  - Découpage en tranches et en activités détaillées

## 1.2.2 Module de Budget

- **Ouverture et Clôture des Exercices :**
  - Création d'un nouvel exercice avec reprise des lignes budgétaires antérieures
  - Clôture d'exercice et génération automatique des annexes réglementaires
- **Élaboration Interactive du Budget :**
  - Saisie des prévisions de dépenses et recettes via une interface conviviale
  - Simulation prédictive et ajustement dynamique des crédits
- **Gestion des Lignes Budgétaires :**
  - Répartition automatisée des crédits aux entités selon la nomenclature
  - Suivi et ajustement en temps réel via des tableaux de bord et reporting automatisé

## 1.2.3 Module de Dépenses

- **Engagement des Dépenses :**
  - Multiples procédures (normale, simplifiée, dérogatoire)
  - Association numérique des pièces justificatives
- **Liquidation et Ordonnancement :**
  - Workflows de validation multi-niveaux pour séparer les rôles (ordonnateur, comptable)
  - Journalisation complète et traçabilité
- **Suivi et Contrôle :**
  - Tableaux de bord pour surveiller la consommation des crédits
  - Génération automatique de rapports et alertes en cas de dépassements

## 1.2.4 Module de Recettes

- **Émission des Titres de Recettes :**
  - Création de titres "à priori" et "à posteriori"
  - Intégration d'imputations comptables et pièces justificatives
- **Recouvrement et Régularisation :**
  - Processus automatisé de suivi du recouvrement et de régularisation

- Validation collaborative et notifications automatiques

### 1.2.5 Module de Trésorerie

- **Suivi en Temps Réel :**
  - Tableaux de bord interactifs affichant la situation de trésorerie
  - Simulation des paiements et des transferts selon des critères définis
- **Gestion des Caisses et Billetage :**
  - Administration multi-caisses avec affectation des caissiers
  - Proposition automatique de billetage ajustable
- **Opérations de Trésorerie :**
  - Enregistrement et validation des encaissements, décaissements et transferts
  - Génération automatique des reçus et rapports de caisse

### 1.2.6 Module de Comptabilité

- **Gestion des Écritures Comptables :**
  - Saisie, reprise et report des écritures sur plusieurs exercices
  - Automatisation de la génération des écritures (issues des modules Dépenses, Recettes, Trésorerie)
- **Assistance à la Saisie et Contrôles Internes :**
  - Auto-complétion, calcul de TVA et vérification des doublons
  - Exportation des états financiers détaillés (Balance, Grand Livre, Bilans)

### 1.2.7 Module de Pilotage

- **Tableaux de Bord et Reporting :**
  - Création et personnalisation de tableaux de bord interactifs
  - Génération de rapports en temps réel et export dans divers formats
- **Gestion des Indicateurs de Performance :**
  - Définition, agrégation et comparaison des KPIs par module
  - Système de notifications et alertes pour déclencher des actions correctives
- **Support Décisionnel par LLM :**
  - Rapports en langage naturel et recommandations stratégiques
  - Assistance via un chatbot intégré pour l'analyse des indicateurs

## 1.2.8 Module de Gestion de la Solde et des Ressources Humaines

- **Gestion de la Paie :**
  - Calcul automatique des salaires, primes, retenues et cotisations
  - Mises à jour dynamiques basées sur la réglementation en vigueur
- **Administration RH :**
  - Suivi complet du cycle de vie des employés (recrutement, formation, évolutions, absences, congés, retraite)
  - Portail self-service et chatbot RH pour une assistance immédiate

## 1.2.9 Module de Gestion du Patrimoine

- **Inventaire et Suivi des Actifs :**
  - Recensement et digitalisation des actifs (immobiliers, mobiliers)
  - Intégration de capteurs IoT pour le suivi en temps réel
- **Maintenance et Valorisation :**
  - Planification des maintenances préventives et correctives
  - Calcul des amortissements et suivi de la dépréciation

## 1.2.10 Module de Gestion des Marchés Publics

- **Passation des Marchés :**
  - Publication et gestion des appels d'offres via une plateforme collaborative
  - Évaluation assistée par LLM des offres soumises par les fournisseurs
- **Suivi Contractuel et Audit :**
  - Gestion intégrée des contrats, suivi de l'exécution et génération automatique de rapports d'audit

## 1.2.11 Module de Fiscalité Locale et Recouvrement

- **Gestion des Impôts et Taxes :**
  - Définition des taux, exonérations et calcul automatisé des montants
  - Émission d'avis d'imposition avec mise à jour en temps réel
- **Recouvrement :**
  - Processus automatisé de relance et suivi des paiements
  - Gestion des contentieux et régularisation des écarts

## 1.2.12 Module de Décentralisation et Transfert de Compétences

- **Suivi des Transferts de Compétences :**
  - Enregistrement des transferts (source, destination, date, description)
  - Détail des ressources transférées et tableaux de bord consolidés
- **Coordination Collaborative :**
  - Interfaces partagées et workflows communs entre l'État et les collectivités locales

## 1.2.13 Module de Gouvernance, Conformité et Reporting Stratégique

- **Gestion des Accès et des Rôles :**
  - Définition des rôles (User\_Role) et gestion des utilisateurs (User)
  - Journalisation des accès et suivi via Access\_Log
- **Reporting et Tableaux de Bord :**
  - Création de rapports stratégiques et de dashboards personnalisables
  - Génération de rapports en langage naturel via LLM et notifications d'alerte

## 1.2.14 Module de Communication et Gestion Documentaire

- **Gestion Électronique des Documents (GED) :**
  - Stockage, indexation et recherche des documents (avec support OCR)
  - Suivi des versions et historique via Document\_Version
- **Portail Collaboratif :**
  - Messagerie instantanée, partage de fichiers et organisation de visioconférences
  - Chatbot documentaire pour faciliter l'accès aux informations

## 1.2.15 Module d'Intelligence Artificielle Transverse et LLM

- **Analyse Prédicative et Détection d'Anomalies :**
  - Algorithmes de machine learning pour anticiper les écarts et anomalies
  - Recommandations automatiques pour optimiser les processus
- **Génération de Rapports en Langage Naturel :**

- Synthèse des données et génération de rapports intelligents
- Assistance via chatbot pour répondre aux requêtes en langage naturel
- **Automatisation par RPA :**
  - Exécution de tâches répétitives et optimisation des workflows grâce à des robots logiciels
  - Système d'apprentissage continu pour adapter les processus

### 1.2.16 Module Infrastructure Sécurisée et Interopérable

- **Architecture Cloud-Native et API-First :**
  - Déploiement via Docker/Kubernetes assurant scalabilité et haute disponibilité
  - Exposition d'API RESTful pour une intégration fluide avec d'autres systèmes
- **Sécurité Avancée :**
  - Mise en œuvre d'authentification multi-facteur, chiffrement AES-256, et gestion des accès via OAuth2/JWT
  - Monitoring en temps réel et journalisation des événements de sécurité via Security\_Log
- **Configuration et Interopérabilité :**
  - Paramétrage centralisé du système et gestion des configurations via la table Configuration

## 1.3 Spécifications Fonctionnelles et Non-Fonctionnelles

### 1.3.1 Spécifications Fonctionnelles

Chaque module devra répondre aux besoins fonctionnels suivants :

- **Exactitude et Traçabilité :**

Toutes les opérations (budgétaires, comptables, paie, etc.) doivent être enregistrées avec précision et permettre une traçabilité complète.
- **Interconnexion et Intégration :**

Les données de chaque module doivent pouvoir être partagées via des clés étrangères et jointures, assurant ainsi une cohérence transversale dans l'ERP.
- **Automatisation et Assistance :**

Les processus récurrents (engagement, recouvrement, paie, etc.) doivent

être automatisés, avec des workflows configurables et un support via IA et LLM.

- **Personnalisation et Adaptabilité :**  
L'ERP doit permettre un paramétrage poussé des règles métier (nomenclature, règles de paie, politiques budgétaires, etc.) pour s'adapter aux exigences réglementaires et organisationnelles spécifiques.
- **Reporting et Analyse :**  
Fourniture de tableaux de bord interactifs et de rapports détaillés, avec export dans divers formats, afin de faciliter la prise de décision à tous les niveaux.

### 1.3.2 Spécifications Non-Fonctionnelles (Besoins Techniques)

- **Performance :**
  - Temps de réponse rapide (objectif : moins de 2 secondes pour les requêtes courantes).
  - Optimisation des index et des jointures pour assurer des performances élevées même avec de gros volumes de données.
- **Scalabilité :**
  - Architecture cloud-native permettant une scalabilité horizontale via des conteneurs et une orchestration (Kubernetes).
  - Capacité à gérer l'augmentation du nombre d'utilisateurs et du volume des transactions.
- **Sécurité :**
  - Authentification multi-facteur et chiffrement des données (en transit et au repos).
  - Mise en place d'un monitoring et d'une journalisation exhaustive pour détecter et répondre aux incidents.
- **Interopérabilité :**
  - Exposition d'API RESTful pour faciliter l'intégration avec des systèmes externes (ERP legacy, systèmes bancaires, outils BI).
  - Utilisation de standards de communication (JSON, XML).
- **Disponibilité et Tolérance aux Pannes :**
  - Système conçu pour une haute disponibilité (SLA > 99,5 %) avec des mécanismes de redondance et de basculement.
  - Procédures de sauvegarde et de restauration régulières.
- **Maintenance et Évolutivité :**

- Documentation technique complète et mise à jour du dictionnaire de données.
- Architecture modulaire facilitant l'ajout de nouvelles fonctionnalités sans interruption des services existants.
- **Internationalisation et Accessibilité :**
  - Interface multilingue et adaptabilité aux formats régionaux (dates, devises).
  - Conformité aux normes d'accessibilité pour assurer une utilisation aisée par tous les utilisateurs.
- **Conformité aux Normes :**
  - Respect des normes ISO/IEC 25010:2011 pour la qualité des logiciels.
  - Conformité aux exigences réglementaires locales (ex. OHADA pour la comptabilité).
- **Support et Formation :**
  - Fourniture de guides utilisateur, de formations et de supports de documentation pour faciliter l'adoption du système.

## 1.4 Plan de Développement sur 6 Mois

Ce plan détaille l'ensemble des modules et processus fonctionnels ainsi que leur intégration dans une architecture technique moderne (microservices, multi-tenant, Docker/Kubernetes, CI/CD, etc.). Chaque phase est décrite avec ses objectifs, ses livrables et ses principales activités. Un diagramme de Gantt illustre également ce planning.

### 1.4.1 Objectifs Généraux

- **Développer une solution ERP multi-tenant complète** intégrant la gestion des finances publiques et locales, couvrant l'ensemble des modules suivants :
  - Nomenclature
  - Budget (Budget des Moyens et Budget Programme)
  - Dépenses
  - Recettes
  - Trésorerie
  - Comptabilité
  - Pilotage & Reporting
  - Solde & RH
  - Patrimoine
  - Marchés Publics
  - Fiscalité & Recouvrement
  - Décentralisation & Transfert
  - Gouvernance, Conformité & Reporting Stratégique

- Communication & GED
- Intelligence Artificielle & LLM
- **Implémenter une architecture microservices** basée sur Java (Spring Boot) et Python (FastAPI) avec conteneurisation (Docker) et orchestration (Kubernetes).
- **Mettre en place un pipeline CI/CD avec GitLab** pour automatiser la compilation, les tests et le déploiement.
- **Respecter les normes ITIL, Lean, ISO, etc.** pour assurer la qualité, la sécurité, l'amélioration continue et l'indépendance du SGBDR.

## 1.4.2 Calendrier et Phases de Développement

### 1.4.2.1 Mois 1 : Phase de Planification et Conception

- **Analyse des besoins et étude de faisabilité**
  - Collecte des exigences fonctionnelles et techniques auprès des parties prenantes.
  - Analyse des référentiels (ITIL, Lean, ISO, etc.) et des meilleures pratiques.
- **Rédaction du cahier des charges**
  - Documentation détaillée des modules, processus métiers, architecture technique et fonctionnelle.
- **Conception de l'architecture**
  - Définition de la cartographie fonctionnelle et applicative complète.
  - Conception de l'architecture microservices (API Gateway, répartition entre FastAPI et Spring Boot, intégration de Kafka, Airflow, Hop, Superset).
- **Mise en place de l'environnement de développement**
  - Installation des outils CI/CD (GitLab, Docker, Kubernetes).

### 1.4.2.2 Mois 2 : Développement des Modules Fondamentaux

- **Module de Nomenclature**
  - Développement des entités, API CRUD et logique multi-tenant.
- **Module Budget des Moyens**
  - Mise en place du budget des moyens avec suivi détaillé des crédits, ajustements et traçabilité.
- **Développement du Data Middleware et de la couche d'abstraction SGBDR**

- Implémentation des repositories et ORM pour garantir l'indépendance vis-à-vis du SGBDR.

- **Mise en place de l'API Gateway**

- Configuration et déploiement initial de l'API Gateway.

#### 1.4.2.3 Mois 3 : Développement des Modules Financiers et Comptables

- **Module Budget Programme**

- Implémentation de la structure à 4 niveaux : Programme > Actions > Activités > Tâches.
- Définition des KPI et des processus de validation et de simulation.

- **Modules Dépenses et Recettes**

- Développement des interfaces de saisie, workflows de validation, association aux lignes budgétaires.

- **Module Comptabilité**

- Mise en place des journaux, saisie des écritures comptables et intégration des flux budgétaires.

- **Tests Unitaires et d'Intégration**

- Exécution des tests pour chaque module développé.

#### 1.4.2.4 Mois 4 : Développement des Modules RH, Patrimoine et Marchés Publics

- **Module Solde & RH**

- Développement des fonctionnalités de gestion des employés, de la paie, du temps et des congés.

- **Module Patrimoine**

- Enregistrement, suivi, maintenance et gestion des actifs.

- **Module Marchés Publics et Fiscalité**

- Développement des processus de passation des marchés (appel d'offres, offres, contrats) et du recouvrement fiscal.

- **Intégration continue et tests fonctionnels**

- Vérification de l'interopérabilité entre modules financiers, comptables, RH et patrimoniaux.

#### 1.4.2.5 Mois 5 : Développement des Modules de Gouvernance, Communication & IA

- **Module Gouvernance, Conformité & Reporting Stratégique**

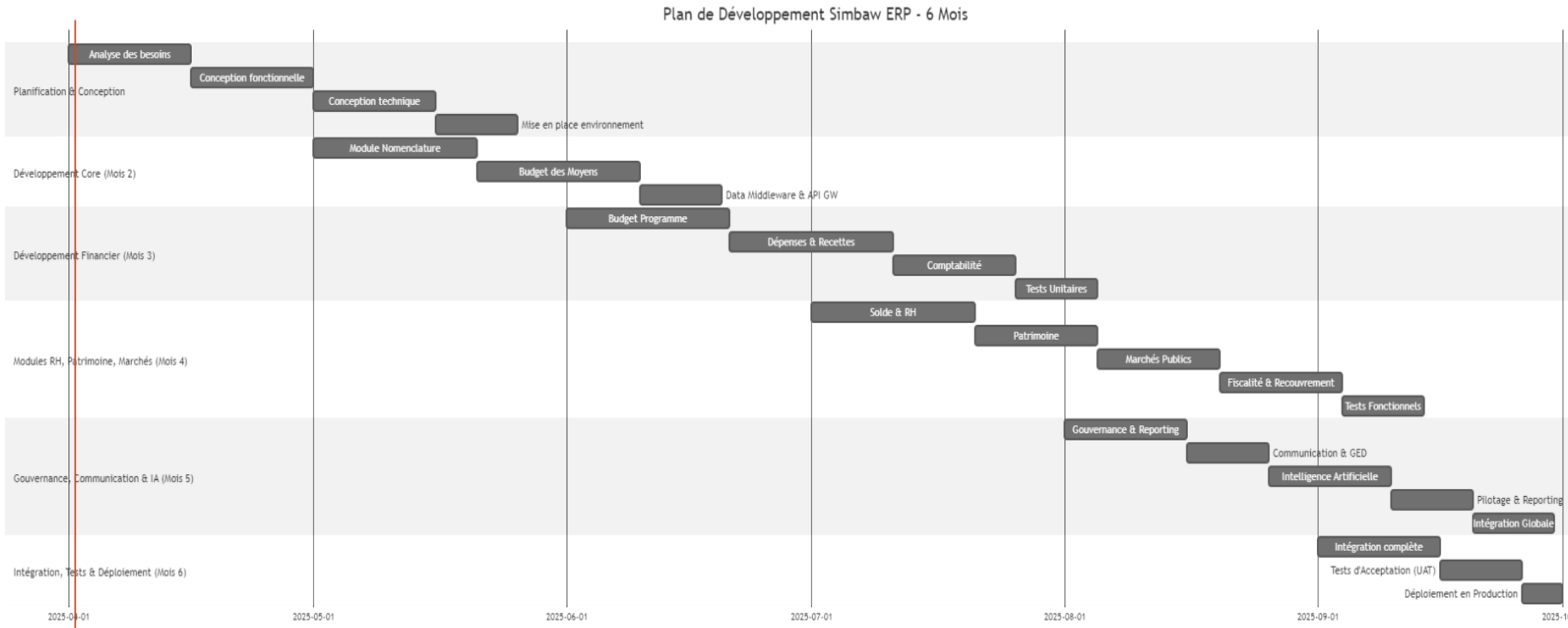
- Développement des fonctionnalités de gestion des rôles, journalisation et génération de rapports stratégiques.

- **Module Communication & GED**
  - Implémentation de la gestion documentaire, du versioning, et de la collaboration interne.
- **Module Intelligence Artificielle & LLM**
  - Déploiement des modèles d'IA, intégration du chatbot et des outils d'analyse prédictive.
- **Développement du module Pilotage & Reporting**
  - Intégration des KPIs, dashboards interactifs et système d'alertes.
- **Tests d'intégration globale**
  - Vérification de la cohérence des flux entre tous les modules.

#### 1.4.2.6 Mois 6 : Intégration, Tests, Déploiement et Recette

- **Intégration complète**
  - Rassembler et intégrer tous les modules dans une plateforme cohérente.
- **Mise en place du Pipeline CI/CD avec GitLab**
  - Configuration complète du pipeline pour build, tests, création d'images Docker et déploiement sur Kubernetes.
- **Déploiement sur Environnement de Staging**
  - Tests d'acceptation utilisateur (UAT) et ajustements.
- **Documentation, Formation et Passage en Production**
  - Rédaction de la documentation finale, formation des utilisateurs et déploiement final sur l'environnement de production.

### 1.4.3 Diagramme de Gantt (Planification sur 6 mois)



## 1.4.4 Conclusion (plan de développement)

Ce plan de développement détaillé sur 6 mois pour Simbaw ERP intègre :

1. **La cartographie fonctionnelle complète** – décrivant tous les modules métiers et leurs interactions.
2. **La cartographie applicative complète** – précisant la répartition des microservices entre les technologies Python (FastAPI) et Java (Spring Boot) ainsi que l'intégration des solutions Apache pour la communication et l'orchestration.
3. **L'architecture fonctionnelle complète** – exposant l'organisation hiérarchique des processus métiers, du pilotage stratégique aux opérations quotidiennes.
4. **L'architecture applicative complète** – détaillant la distribution des composants logiciels, leur communication via API Gateway et Kafka, et leur déploiement sur une infrastructure cloud-native avec Docker/Kubernetes.

Ce planning fournit une feuille de route claire pour le développement de Simbaw ERP, assurant une mise en œuvre progressive, des tests rigoureux et un déploiement en production dans un environnement multi-tenant moderne et performant.

## 1.5 Conclusion

Ce cahier des charges présente de manière exhaustive les besoins fonctionnels et techniques de Simbaw ERP. La modélisation logique, avec la définition des clés primaires, des clés étrangères, des index et la normalisation (3NF ou plus), servira de base solide pour le schéma de base de données.

En combinant des modules fonctionnels détaillés couvrant l'ensemble des processus métiers (nomenclature, budget, dépenses, recettes, trésorerie, comptabilité, pilotage, paie/RH, patrimoine, marchés publics, fiscalité, décentralisation, gouvernance, communication, intelligence artificielle et infrastructure), Simbaw ERP se positionne comme une solution ERP intégrée, sécurisée, évolutive et interopérable, répondant aux exigences complexes des finances publiques et locales modernes.

Ce cahier des charges servira de référence tout au long du projet pour guider la conception, le développement et le déploiement de Simbaw ERP,

tout en assurant que la solution finale offre des performances optimales et une flexibilité maximale pour s'adapter aux évolutions futures.

## 2. Annexes techniques

### 2.1 Répartition des modules de Simbaw ERP

Ci-dessous la répartition des modules de Simbaw ERP en fonction des critères suivants :

- **Modules nécessitant rapidité de développement, flexibilité et intégration poussée avec des capacités d'IA et LLM**  
(Développement en Python avec FastAPI)
- **Modules nécessitant robustesse, gestion de la concurrence et maturité en environnement d'entreprise**  
(Développement en Java avec Spring Boot)

#### 2.1.1 Modules en Python (FastAPI)

Ces modules bénéficient d'un cycle de développement rapide et d'une intégration aisée avec des outils d'intelligence artificielle et des frameworks ML/LLM.

##### 1. Module de Pilotage

- **Rôle** : Fournir des tableaux de bord interactifs et des rapports en temps réel, avec génération de rapports en langage naturel via un LLM.
- **Avantage FastAPI** : Rapidité de prototypage et intégration fluide avec des bibliothèques d'analyse de données et de visualisation (par exemple, Plotly, Dash).

##### 2. Module d'Intelligence Artificielle Transverse et LLM

- **Rôle** : Réaliser l'analyse prédictive, la détection d'anomalies, et générer des rapports en langage naturel.
- **Avantage FastAPI** : Facilité d'intégration avec des frameworks Python (TensorFlow, PyTorch, Hugging Face) et mise en place rapide de microservices d'IA.

##### 3. Module de Communication et Gestion Documentaire

- **Rôle** : Centraliser la gestion documentaire, permettre la collaboration et intégrer un chatbot pour la recherche d'information.
- **Avantage FastAPI** : Rapidité de développement des API REST et intégration aisée de solutions NLP pour le chatbot.

## 2.1.2 Modules en Java Spring Boot

Ces modules sont critiques pour le cœur fonctionnel de l'ERP, nécessitant robustesse, performance transactionnelle, gestion de la concurrence et évolutivité en environnement d'entreprise.

### 1. Module de Nomenclature

- **Rôle** : Centraliser et normaliser les données de référence (organigramme, comptes, programmes, projets).
- **Justification Java** : Besoin d'une gestion transactionnelle robuste pour garantir l'intégrité des données de base.

### 2. Module Budget

- **Rôle** : Préparation et suivi dynamique des budgets de fonctionnement et d'investissement.
- **Justification Java** : Processus critiques en termes de transactions et de validation multi-niveaux.

### 3. Module Dépenses

- **Rôle** : Automatiser et sécuriser l'engagement, la liquidation et l'ordonnancement des dépenses.
- **Justification Java** : Forte exigence en termes de fiabilité et de traçabilité des opérations financières.

### 4. Module Recettes

- **Rôle** : Gérer l'émission, la régularisation et le recouvrement des recettes.
- **Justification Java** : Nécessite une gestion robuste des flux financiers et de la sécurité transactionnelle.

### 5. Module Trésorerie

- **Rôle** : Suivi en temps réel des liquidités et gestion des flux de trésorerie.
- **Justification Java** : Haute concurrence et exigences en termes de performance pour la gestion des opérations bancaires.

### 6. Module Comptabilité

- **Rôle** : Centraliser les écritures comptables et produire les états financiers.
- **Justification Java** : Environnement de haute fiabilité et forte charge transactionnelle, avec besoin de normes strictes.

### 7. Module de Gestion de la Solde et des RH

- **Rôle** : Gestion de la paie et des ressources humaines, avec suivi du cycle de vie des employés.

- **Justification Java** : Critique en matière de confidentialité, de sécurité et de robustesse des transactions.

#### 8. Module de Gestion du Patrimoine

- **Rôle** : Suivi, maintenance et valorisation des actifs.
- **Justification Java** : Gestion d'opérations complexes et besoin d'une forte cohérence des données sur le long terme.

#### 9. Module des Marchés Publics

- **Rôle** : Automatiser la passation des marchés publics et assurer un suivi contractuel rigoureux.
- **Justification Java** : Exigences de conformité, transparence et sécurité accrues dans un processus réglementé.

#### 10. Module de Fiscalité Locale et Recouvrement

- **Rôle** : Gestion des impôts, avis d'imposition et suivi du recouvrement fiscal.
- **Justification Java** : Nécessite robustesse, haute disponibilité et gestion des contentieux.

#### 11. Module de Décentralisation et Transfert de Compétences

- **Rôle** : Coordonner les transferts de compétences et de ressources entre l'État et les collectivités.
- **Justification Java** : Gestion d'opérations inter-étatiques complexes nécessitant une forte intégrité transactionnelle.

#### 12. Module de Gouvernance, Conformité et Reporting Stratégique

- **Rôle** : Fournir des outils de contrôle, reporting et sécurité pour les instances dirigeantes.
- **Justification Java** : Environnement exigeant une haute sécurité, une gestion fine des accès et une performance de traitement.

#### 13. Module Infrastructure Sécurisée et Interopérable

- **Rôle** : Assurer la robustesse, la sécurité et l'intégration fluide de l'ERP dans un environnement cloud-native.
- **Justification Java** : Nécessite une architecture éprouvée pour la gestion de la concurrence, de la sécurité et des transactions dans un contexte d'entreprise.

### 2.1.3 Synthèse de la Répartition

| Modules à développer en Python (FastAPI)   | Modules à développer en Java Spring Boot |
|--------------------------------------------|------------------------------------------|
| - Pilotage (Dashboard, Reporting avec LLM) | - Nomenclature (données de référence)    |

|                                                        |                                                                              |
|--------------------------------------------------------|------------------------------------------------------------------------------|
| - Intelligence Artificielle Transverse et LLM          | - Budget (élaboration, suivi budgétaire)                                     |
| - Communication et Gestion Documentaire (GED, Chatbot) | - Dépenses (engagement, liquidation, ordonnancement)                         |
|                                                        | - Recettes (émission, recouvrement)                                          |
|                                                        | - Trésorerie (gestion des flux, caisses)                                     |
|                                                        | - Comptabilité (écritures, reporting financier)                              |
|                                                        | - Solde et RH (paie, gestion des employés)                                   |
|                                                        | - Gestion du Patrimoine (actifs, maintenance, cessions)                      |
|                                                        | - Marchés Publics (passation, suivi contractuel)                             |
|                                                        | - Fiscalité Locale et Recouvrement (impôts, contentieux)                     |
|                                                        | - Décentralisation et Transfert de Compétences (coordination inter-étatique) |
|                                                        | - Gouvernance, Conformité et Reporting Stratégique (sécurité, audits)        |
|                                                        | - Infrastructure Sécurisée et Interopérable (API, sécurité, monitoring)      |

- **Explications**
- **Modules Python (FastAPI) :**

Ces modules sont destinés à offrir des services rapides et flexibles, notamment pour le support décisionnel, l'analyse en langage naturel et la gestion documentaire. L'intégration d'un LLM se fait naturellement en Python, grâce à l'écosystème riche en bibliothèques d'IA.

- **Modules Java (Spring Boot) :**

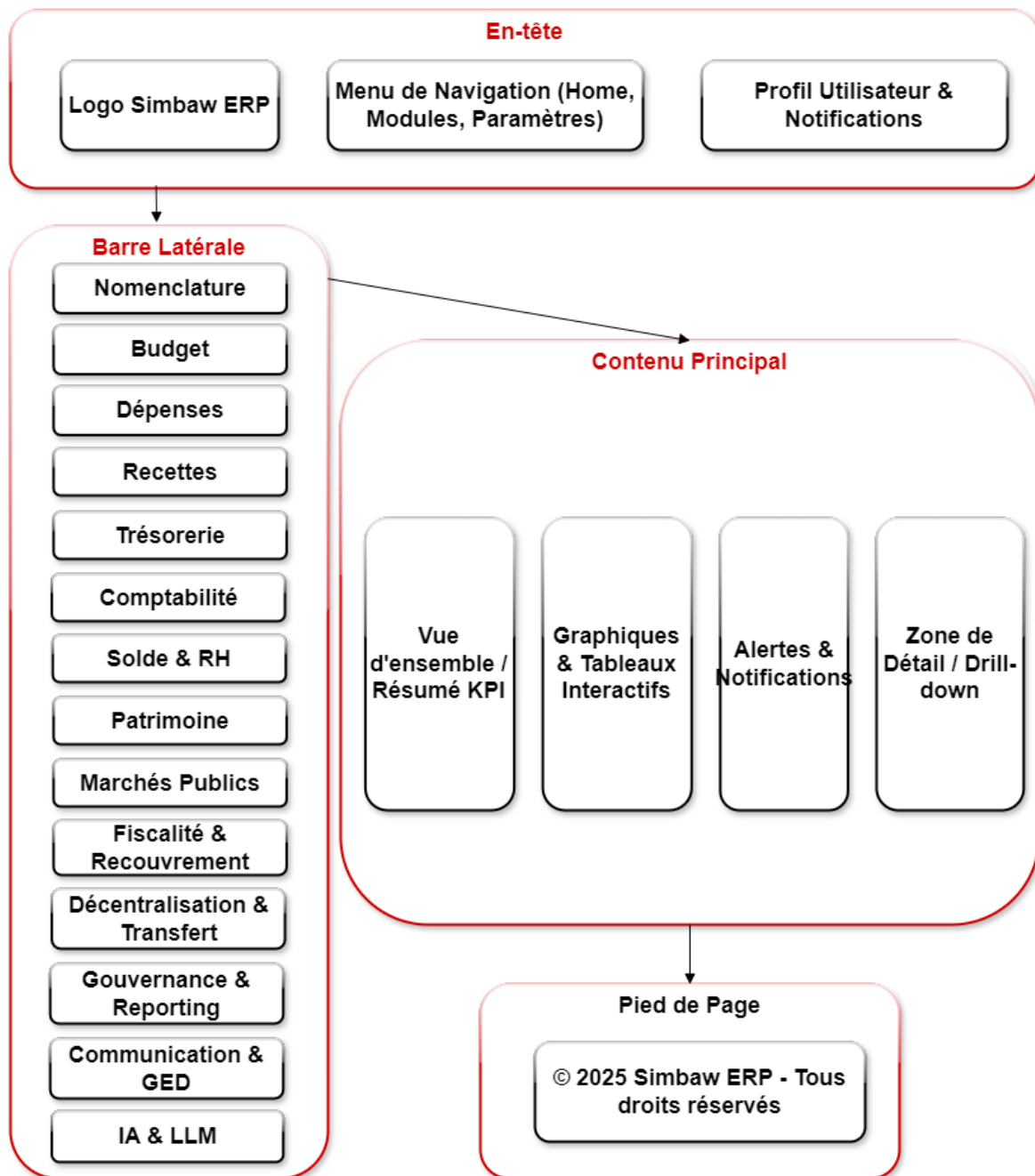
Les modules critiques et transactionnels (comptabilité, budget, dépenses, trésorerie, RH, etc.) nécessitent une gestion robuste, une forte gestion de la concurrence et une maturité éprouvée en environnement d'entreprise, ce que Spring Boot offre par sa scalabilité, sa sécurité et son support pour les transactions.

Cette répartition permet d'allier la rapidité de développement et l'intégration de technologies avancées (Python/FastAPI pour l'IA et le reporting interactif) avec la robustesse et la performance attendues dans les systèmes ERP d'entreprise (Java Spring Boot pour les modules transactionnels et critiques).

## 2.2 Wireframe pour le dashboard principal de Simbaw ERP

Ci-dessous une première mouture de wireframe pour le dashboard principal de Simbaw ERP. Ce wireframe illustre une interface utilisateur intuitive et

complète, comprenant un en-tête, une barre latérale de navigation, une zone de contenu principal affichant des indicateurs clés et des graphiques, ainsi qu'un pied de page.



**Explications :**

- **En-tête (Header) :**

Contient le logo de l'ERP, un menu de navigation global (pour accéder aux principales sections et paramètres), ainsi que l'accès au profil utilisateur et aux notifications.

- **Barre Latérale (Sidebar) :**

Propose une navigation verticale listant tous les modules de Simbaw ERP (Nomenclature, Budget, Dépenses, Recettes, Trésorerie, Comptabilité, Solde & RH, Patrimoine, Marchés Publics, Fiscalité, Décentralisation, Gouvernance, Communication, IA & LLM) pour un accès rapide à chaque section.

- **Contenu Principal (MainContent) :**

Affiche les indicateurs clés (résumé KPI) sous forme de cartes, des graphiques interactifs et des tableaux de bord pour le suivi en temps réel, ainsi que les alertes et notifications. Une zone de détail permet d'approfondir l'analyse via des drill-down.

- **Pied de Page (Footer) :**

Affiche des informations légales et la mention du copyright.

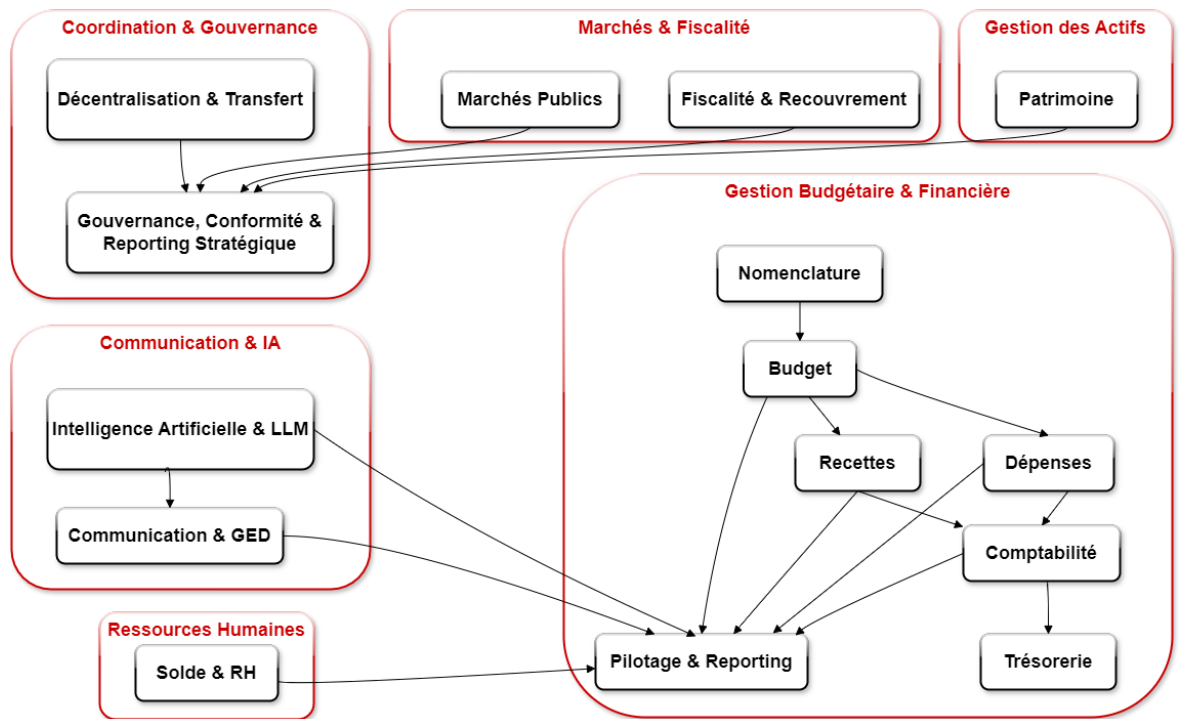
Ce wireframe fournit une base visuelle pour la conception de l'interface utilisateur du dashboard principal, assurant ainsi une expérience intuitive et complète pour les utilisateurs de Simbaw ERP.

## 2.3 Cartographies et architectures

Ci-dessous une version finale et complète qui intègre la cartographie fonctionnelle et applicative, ainsi que l'architecture fonctionnelle et applicative de Simbaw ERP. Chaque section est accompagnée d'un schéma pour visualiser clairement les modules, leurs interactions et l'intégration technologique associée.

### 2.3.1 Cartographie Fonctionnelle Complète

Ce schéma présente l'ensemble des modules métiers de Simbaw ERP et leurs interconnexions fonctionnelles. Il couvre la gestion des données de référence, la planification budgétaire, l'exécution financière, la gestion des ressources humaines, du patrimoine, des marchés publics, de la fiscalité, de la décentralisation, ainsi que la gouvernance, la communication et l'analyse via l'IA.

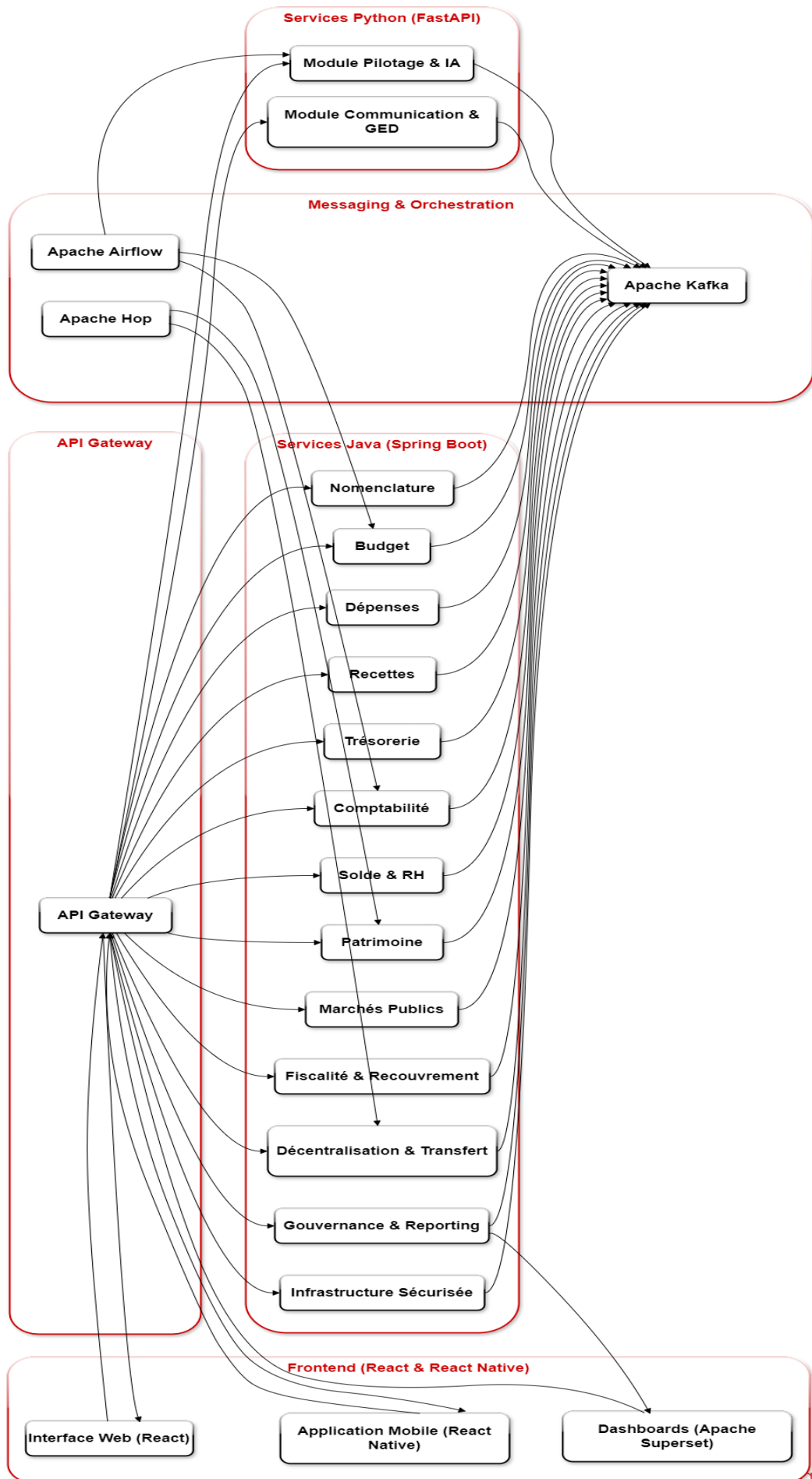


### Explications :

- **Gestion Budgétaire & Financière** : La nomenclature alimente le budget, qui génère ensuite des dépenses et des recettes. La comptabilité et la trésorerie garantissent l'intégrité des flux financiers, le tout piloté par des outils de reporting.
- **Ressources Humaines** : Gère la solde et la paie des employés.
- **Gestion des Actifs** : Suivi, maintenance et cession des actifs.
- **Marchés & Fiscalité** : Couvrent la passation des marchés publics et le recouvrement fiscal.
- **Coordination & Gouvernance** : Assurent la décentralisation, le transfert de compétences et le reporting stratégique.
- **Communication & IA** : Centralisent la gestion documentaire et apportent une couche d'analyse prédictive avec l'IA.

## 2.3.2 Cartographie Applicative Complète

Ce schéma montre la répartition des composants logiciels de Simbaw ERP et leur interaction. Il distingue les services développés en Python (FastAPI) pour les modules agiles et analytiques et ceux en Java (Spring Boot) pour les modules transactionnels critiques. L'ensemble est intégré à une infrastructure de messaging et d'orchestration avec Apache Kafka, Airflow et Hop, et les dashboards BI sont réalisés avec Apache Superset.

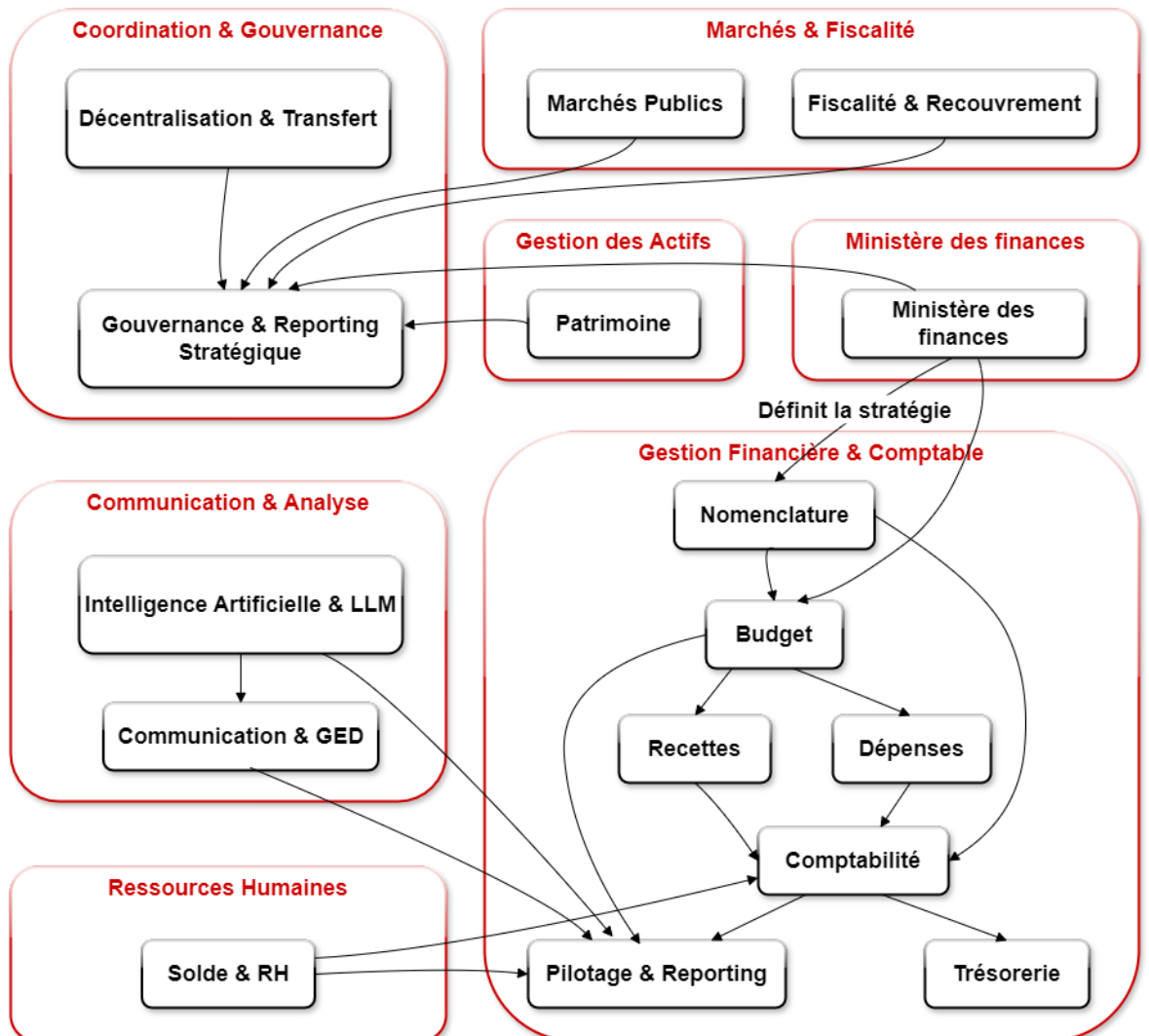


**Explications**

- **Frontend** : Développé en React et React Native, il fournit l'interface utilisateur ainsi que les dashboards BI via Superset.
- **API Gateway** : Centralise l'accès aux microservices, garantissant la sécurité et la gestion multi-tenant.
- **Services Python (FastAPI)** : Gèrent les modules agiles comme le pilotage et l'IA, ainsi que la communication et la GED.
- **Services Java (Spring Boot)** : Traitement des modules critiques (nomenclature, budget, dépenses, etc.).
- **Messaging & Orchestration** : Apache Kafka assure la communication asynchrone, et Airflow/Hop orchestrent les flux ETL.

### 2.3.3 Architecture Fonctionnelle Complète

Cette architecture fonctionnelle décrit l'organisation des processus métiers de Simbaw ERP, de la stratégie jusqu'au support décisionnel. Elle illustre comment le ministère des finances oriente les différents modules pour fournir une gestion intégrée des finances publiques et locales.



### Explications :

- Le **ministère des Finances** établit la stratégie qui se traduit par la mise en œuvre de processus financiers, comptables et décisionnels.
- Les modules de **Gestion Financière & Comptable** offrent une vision intégrée des flux (budget, dépenses, recettes, trésorerie, comptabilité) et alimentent le pilotage et le reporting.
- Les autres modules (RH, patrimoine, marchés, fiscalité, décentralisation, gouvernance, communication & IA) apportent des fonctionnalités spécifiques, assurant une gestion complète et cohérente.

## 2.3.4 Architecture Applicative Complète

Le schéma, idem à celui de la cartographie applicative décrit la répartition des composants logiciels, leur communication via une API Gateway, ainsi que leur déploiement sur une infrastructure cloud-native conteneurisée (Docker/Kubernetes). Il intègre également les technologies Apache (Kafka, Airflow, Hop, Superset) pour la messagerie, l'orchestration des flux ETL et le reporting BI.

### Explications :

- **Frontend** : Développé en React pour le web et React Native pour le mobile, avec Apache Superset pour la visualisation BI.
- **API Gateway** : Point d'entrée central qui oriente le trafic vers les différents microservices.
- **Services Python (FastAPI)** : Gèrent les modules agiles (pilotage et IA, communication/documentation).
- **Services Java (Spring Boot)** : Traitent les modules critiques (nomenclature, budget, dépenses, recettes, etc.) nécessitant robustesse et sécurité.
- **Messaging & Orchestration** : Apache Kafka gère la communication interservices, et Airflow ainsi que Hop orchestrent les flux ETL et les processus d'intégration.
- **Infrastructure Sécurisée** : Les microservices sont déployés sur une infrastructure cloud-native (Docker/Kubernetes), assurant haute disponibilité, scalabilité et gestion multi-tenant.

### Conclusion

Nous avons présenté :

#### 1. La cartographie fonctionnelle

- Décrivant tous les modules métiers de Simbaw ERP et leurs interactions fonctionnelles, depuis la gestion budgétaire et comptable jusqu'à la gouvernance, en passant par la gestion RH, des actifs, des marchés et de la communication/IA.

#### 2. La cartographie applicative

- Illustrant la répartition des microservices entre les technologies Python (FastAPI) et Java (Spring Boot), ainsi que l'intégration des outils Apache (Kafka, Airflow, Hop, Superset) et des frontends en React/React Native.

### 3. **L'architecture fonctionnelle**

- Exposant l'organisation hiérarchique des processus métiers et la manière dont la stratégie se traduit en opérations via les différents modules.

### 4. **L'architecture applicative**

- Détaillant la répartition des composants logiciels, leur communication via l'API Gateway et Kafka, l'orchestration ETL avec Airflow et Hop, et le déploiement sur une infrastructure cloud-native.

Ces documents fournissent une vue d'ensemble et servent de référence pour la conception, le développement et le déploiement de Simbaw ERP dans un environnement multi-tenant moderne, évolutif et performant.