# Cahier des Charges Global (CDC) et Détaillé pour la Plateforme PREDYKT

## I. Définition du Projet et Objectifs Stratégiques

### 1.1. Vision et Justification du Projet PREDYKT

Le projet **PREDYKT** vise à créer la première plateforme panafricaine d’analyse et de prédiction financière basée sur une Intelligence Artificielle nativement conçue pour le contexte africain.1 L'ambition est de transformer le pilotage d'entreprise en Afrique, en passant d'une prise de décision basée sur l'intuition à une approche proactive et anticipative, en temps réel.1 PREDYKT cible prioritairement les PME/PMI et ETI en Afrique francophone (CEMAC, UEMOA).1

**Justification Stratégique :** Face à la fragmentation des données et au déficit de pilotage, PREDYKT centralise dynamiquement la comptabilité, la fiscalité, la trésorerie et les flux financiers pour fournir des projections de trésorerie (J+30, J+60, J+90).1 L'avantage compétitif réside dans les **algorithmes localisés et propriétaires** entraînés sur des *datasets* africains pour garantir une précision supérieure sur les taux de change, l'inflation et la conformité locale.1

### 1.2. Architecture Technique Imposée (Stack Lock-in)

L'architecture sera un système de **microservices polyglotte** pour garantir la robustesse et la scalabilité.2

|  |  |  |
| --- | --- | --- |
| **Composant** | **Technologie Principale** | **Rôle Stratégique et Justification** |
| **Backend Critique (GL, Transactionnel)** | **Java (Spring Boot)** | Stabilité, performance transactionnelle, et **fiabilité ACID** requises pour les registres légaux et la conformité.2 Ce service est le **garant de l'immuabilité** des écritures comptables. |
| **Backend Data Science & IA** | **Python (FastAPI/Django)** | Standard pour l'analyse de données, le Machine Learning (ML) et les modèles de Séries Temporelles (ARIMA, Prophet).4 S'occupe des calculs intensifs et des prédictions. |
| **Frontend & UX** | **Next.js (React)** | Expérience utilisateur (UX) rapide, gestion des tableaux de bord dynamiques et utilisation des composants chargés côté serveur (RSC) pour améliorer la performance (Core Web Vitals).5 |
| **Base de Données Principale** | **PostgreSQL** | Base de données relationnelle préférée pour le stockage des données financières structurées (Grand Livre, écritures) en raison de sa robustesse et sa conformité ACID.2 |
| **Base de Données Cache/Temps Réel** | **Redis** | Caching des résultats d'IA et de calculs fréquents, gestion des sessions, suivi de transactions en temps réel.2 |
| **Base de Données Non Structurée** | **MongoDB** | Stockage flexible des logs d'audit, des documents numérisés (pièces jointes aux écritures) et des données utilisateurs.2 |

### 1.3. Structuration du Projet et Découpage des Phases

Le projet sera découpé en trois phases principales, en adoptant une approche agile et progressive pour minimiser les risques (selon le Plan d'Action PREDYKT 1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Axe Stratégique** | **Objectifs Clés** | **Modules Principaux Activés** |
| **PHASE I (MVP V1.0)** | **Validation et Faisabilité** | Valider l'intégration des données hétérogènes et prouver la fiabilité de la **Trésorerie Prédictive** (ROI immédiat pour le client).7 | Core Accounting (GL de base), Moteur d'Intégration (API/OCR de base), Module de Projection de Trésorerie J+30 (ARIMA simple). |
| **PHASE II (V2.0)** | **Croissance et Enrichissement** | Commercialiser le produit, prouver la scalabilité, intégrer les modules Premium basés sur l'IA (Alerte, Risque, Scénarios de base).1 | Automatisation Complète P2P/O2C, Détection d'Anomalies (Early Warning System), Projection J+90, Modélisation du Risque Client. |
| **PHASE III (V3.0)** | **Expansion et Leadership** | Devenir la plateforme de référence panafricaine et monétiser l'intelligence via des API externes.1 | Gestion des Immobilisations, Analyse de Scénarios Synthétiques (IA Générative), Lancement de l'API PREDYKT pour les partenaires. |

## II. Exigences Fonctionnelles Détaillées (RFE) par Module

### 2.1. Module I : Core Accounting et Conformité (Stabilité Java - GL)

Ce module garantit l'intégrité financière et la conformité réglementaire, en s'appuyant sur la robustesse de Java.3

#### 2.1.1. Comptabilité Générale (GL)

* **Plan Comptable :** Gestion multi-niveaux du plan comptable (général, auxiliaire, analytique) pour un suivi fin des finances.8
* **Tenue des Livres :** Saisie des écritures assistée et respect des règles comptables obligatoires, y compris la tenue en **langue française et en euros** (ou devise locale équivalente).10
* **Gestion TVA :** Calcul et gestion automatiques des différents régimes de TVA locaux (collectée, déductible).8
* **Clôture et Rapports Légaux :**
  + Établissement des **Comptes Annuels Obligatoires** (Bilan, Compte de Résultat et Annexe) avec option de présentation simplifiée pour les PME.11
  + Export des écritures comptables au format standardisé (ex. : FEC ou équivalent local) pour l'expert-comptable.8

#### 2.1.2. Conformité et Audit

* **Piste d'Audit Fiable :** Garantie de l'immuabilité et de la traçabilité complète de toutes les écritures (exigence fiscale).12
* **Multi-Normes (V2) :** En phase II, capacité de générer des rapports *secondaires* selon les normes comptables internationales (IFRS ou US GAAP).13
* **Gestion des Immobilisations (V3) :** En phase III, gestion du cycle de vie complet des actifs (fiches d'actif uniques, plans d'amortissement multiples - linéaire/dégressif, suivi des financements : emprunts/crédits-baux).9

### 2.2. Module II : Moteur d'Intégration et Traitement de la Donnée (IA-OCR)

Ce module résout la faiblesse critique de l'**Hétérogénéité des Données** 1 et doit être développé en priorité (MVP).

#### 2.2.1. Ingestion et Normalisation

* **Connexion Multi-Sources :** Intégration via **API sécurisées** (inspirées PSD2) 16 et/ou connecteurs aux systèmes comptables existants (SAGE, Quickbook, etc.).1
* **Import Bancaire :** Support de l'import des relevés bancaires via les protocoles standards (OFX, MT940).18
* **Normalisation :** Nettoyage, standardisation et harmonisation automatiques des données hétérogènes (fichiers, écritures) pour garantir la qualité de l'analyse IA.1
* **OCR/NLP Intelligent (MVP V1.0) :** Développement d'un moteur robuste utilisant le **Traitement du Langage Naturel (NLP)** et l'**OCR** pour l'extraction automatique des données clés (montant, date, compte) à partir de documents non structurés (factures scannées, reçus).4

#### 2.2.2. Automatisation (Rapprochement et Workflow)

* **Rapprochement Bancaire :** Algorithmes d'IA (Phase II) pour automatiser la réconciliation des lignes bancaires avec les opérations comptables et les documents (factures/commandes).1
* **Workflow O2C (Commande-Encaissement) :** Gestion du cycle de la commande, vérification de solvabilité, facturation, jusqu'à la finalisation du paiement.20 Le système doit calculer le **Délai Moyen de Recouvrement (DSO)**.20
* **Workflow P2P (Procure-to-Pay) :** Automatisation du processus d'achat, réception de la facture, validation des dépenses et préparation du règlement.12

### 2.3. Module III : Intelligence Prédictive et Décisionnelle (Moteur Python)

C'est le cœur de la proposition de valeur (**Anticipation** 1) et le moteur Python pour le Machine Learning.3

#### 2.3.1. Prévision Financière (Prédictif)

* **Modèles de Prévision de Trésorerie :** Utilisation d'algorithmes de **Séries Temporelles** avancés (ex. : **ARIMA ou Prophet**) 4 pour fournir :
  + **MVP V1.0 :** Projection de trésorerie **J+30** (objectif de précision élevé).1
  + **V2.0 :** Projection de trésorerie **J+90** et projection des ventes futures.1
* **Optimisation du BFR :** Modèles IA prédisant et analysant le Besoin en Fonds de Roulement (BFR) en se basant sur les flux P2P/O2C.21

#### 2.3.2. Gestion des Risques (Early Warning System)

* **Détection d'Anomalies (V2.0) :** Algorithmes d'apprentissage automatique (supervisés et non supervisés) pour surveiller en permanence les transactions (flux bancaires, écritures GL) et signaler immédiatement les valeurs aberrantes (fraude potentielle, erreurs comptables).22
  + **Implémentation Technique :** Utilisation des librairies Python comme **PyOD** (Outlier Detection) ou **dtaianomaly** (détection d'anomalies de séries temporelles).23
* **Évaluation du Risque Client (V2.0) :** Modèles de **Classification et Clustering** pour segmenter les clients par risque de non-paiement et adapter les stratégies de recouvrement.21

#### 2.3.3. Aide à la Décision (Simulation)

* **Module de Simulation (V2.0) :** Permettre aux dirigeants de simuler l'impact des décisions majeures (ex. : augmentation des stocks, investissement) sur la trésorerie et la rentabilité (*What-If Analysis*).1
* **IA Générative (V3.0) :** Exploration et intégration de modèles d'IA générative pour créer des **scénarios synthétiques** complexes et des **simulations de résistance** (*stress tests*) pour l'optimisation des flux de travail.21

## III. Exigences Non Fonctionnelles (RNF)

### 3.1. Architecture et Scalabilité

* **Scalabilité Horizontale :** L'architecture de microservices doit permettre l'ajout de ressources pour les services Python d'IA (calcul intensif) indépendamment des services Java (transactionnel), garantissant l'absence de défaillances critiques lors de la montée en charge.2
* **Contrats d'API (CDC) :** Mise en œuvre obligatoire des **tests Consumer-Driven Contracts (CDC)** dans le pipeline CI/CD pour sécuriser l'interaction entre les microservices Java (Provider) et Python (Consumer) et prévenir les régressions.26
* **Data Flow :** Mise en place d'une architecture de flux de données dédiée pour l'ETL des données (y compris l'intégration de données en temps réel) avant de les fournir aux modèles ML.28

### 3.2. Sécurité et Conformité Panafricaine

* **Standards de Sécurité :** Alignement du Système de Management de la Sécurité de l'Information (SMSI) sur la norme **ISO 27001** pour la gestion des risques et la confiance client.30 La conformité **PCI DSS** est requise si des données de paiement sont stockées.31
* **Protection des Données (RGPD-like) :** Respect des réglementations locales sur la protection des données (similaires au RGPD).32
  + **Anonymisation/Pseudonymisation :** Les données à caractère personnel (PII) utilisées pour l'entraînement des modèles IA (Python) doivent être anonymisées ou pseudonymisées pour respecter les droits des personnes (droit d'accès, d'effacement).33
* **Résidence des Données :** Utilisation d'infrastructures **cloud souveraines ou sécurisées** pour respecter les réglementations sur la résidence des données dans les pays ciblés.1
* **Résilience (Infrastructures Inégales) :** L'application doit être optimisée pour les faibles bandes passantes et supporter une **synchronisation différée** (Conception "Offline-First") pour permettre aux utilisateurs de travailler même avec une connexion instable.1

## IV. Interfaces Utilisateurs (UX) et Reporting Stratégique

L'interface utilisateur (Next.js) est le point de convergence de l'intelligence et doit transformer les données en **insights actionnables**.34

### 4.1. Tableau de Bord (Dashboard)

* **Design Axé sur la Décision :** Le tableau de bord stratégique (page d'accueil) doit se concentrer sur les **Indicateurs Clés de Performance (KPIs)** essentiels pour les objectifs stratégiques des dirigeants.34
* **KPIs Essentiels :**
  + **Liquidité/Trésorerie :** Solde réel, Flux de Trésorerie d'Exploitation (CF), Solde Prévisionnel (J+30/J+90), Cash Burn Rate.35
  + **Efficacité :** Délai Moyen de Recouvrement (DSO), Marge Brute, Taux d'Automatisation de la Saisie.35
  + **Risque :** Score de Risque Client (IA), Taux d'Anomalies Détectées.1
* **Visualisation :** Utilisation de bibliothèques de visualisation (ex. : **Chart.js**) pour les graphiques dynamiques et interactifs.36
* **Gestion de la Latence :** Utilisation des **Server Components et Client Components** de Next.js pour charger les données transactionnelles rapidement (Java) et les widgets d'IA complexes (Python) de manière asynchrone, optimisant ainsi la performance perçue.5

### 4.2. Explicabilité et Action

* **Explicabilité de l'IA (XAI) :** L'interface doit expliquer les raisons d'une prédiction (ex. : "Cette anomalie est due à [Facteur]") pour renforcer la confiance des utilisateurs dans les modèles Python.21
* **Alertes Contextuelles :** Affichage d'un **Système d'Alerte Proactif** (Early Warning System) 1 fournissant des notifications en temps réel pour des actions prioritaires (ex. : "Relancer le client X, risque de non-paiement prédit").1

### 4.3. Intégration Externe

* **API Unifiée (V3.0) :** Exposition d'une API robuste pour permettre l'intégration avec les systèmes métiers externes (ERP, CRM) et monétiser l'intelligence de PREDYKT auprès des partenaires FinTech.37
* **Intégration Paiements :** Connexion API avec des logiciels de paiement de factures tiers, assurant la synchronisation des données et la sécurité (authentification multifacteur).39

## V. Plan de Développement Détaillé par Phase (Go-to-Market)

Ce plan reprend et enrichit la feuille de route du projet PREDYKT 1 pour garantir une exécution alignée sur les objectifs de croissance.

### PHASE I : Produit Minimum Viable (MVP V1.0) - Validation (Mois 1 à 6)

L'objectif est de valider le cœur du problème (**Trésorerie**) et la faisabilité de l'intégration (**Hétérogénéité des données**).

|  |  |  |  |
| --- | --- | --- | --- |
| **Module/Composant** | **Fonctionnalités Clés du MVP V1.0** | **Technologie Impliquée** | **Livrable Stratégique (Output)** |
| **I. Core Accounting** | Tenue des comptes de base (GL, Plan comptable simple). 8 | Java, PostgreSQL | Base de données transactionnelle sécurisée. |
| **II. Ingestion & Data Flow** | Moteur d'intégration (Import OFX/MT940 ou API simple). OCR/NLP minimaliste (lecture d'un format de facture unique). 1 | Python (NLP/OCR), Java (Connecteurs) | Capacité à ingérer et normaliser les données bancaires et documentaires. |
| **III. Intelligence (IA)** | **Projection de Trésorerie J+30** (Modèle ARIMA simple). 1 | Python (ML/ARIMA) | Preuve de concept d'une prédiction financière localisée. |
| **IV. UX/Pilotage** | Tableau de bord simple (Solde réel + Graphique de projection J+30). 1 | Next.js (Frontend) | MVP stable pour 5-10 clients pilotes.1 |
| **V. Non-Fonctionnel** | Mise en place du **CDC testing** (Java/Python) et du Cadre Légal (Protection des données).27 | Architecture Microservices | Validation de la fiabilité et de l'intégration du *stack* polyglotte. |

### PHASE II : Croissance et Élargissement (V2.0) - Monétisation (Mois 8 à 24)

L'objectif est de consolider l'offre Premium grâce à l'IA et d'accélérer la base clients.

|  |  |  |  |
| --- | --- | --- | --- |
| **Module/Composant** | **Fonctionnalités Clés de la V2.0** | **Technologie Impliquée** | **Livrable Stratégique (Output)** |
| **I. Core Accounting** | Automatisation Complète P2P/O2C (Workflows complexes). 20 | Java | Augmentation de l'efficacité opérationnelle (réduction de la saisie). |
| **II. Intelligence (IA)** | **Détection d'Anomalies (Early Warning)** (via PyOD/dtaianomaly). 23 **Projection de Trésorerie J+90**. 1 | Python (ML) | Produit Premium justifiant un abonnement plus élevé.1 |
| **III. Risque et Décision** | Modélisation du **Risque Client** (Segmentation des débiteurs). Module de **Simulation de Scénarios** simple. 21 | Python (ML) | Outil de pilotage stratégique pour les dirigeants.1 |
| **IV. Technologie IA** | Collecte et labellisation intensive de **datasets** locaux pour affiner les modèles. 1 | Data Engineering, Python | Amélioration de la précision des prédictions (> 85%).1 |
| **V. Non-Fonctionnel** | Obtention de la certification **ISO 27001** (ou équivalent local). 1 | Sécurité, Cloud | Sécurité de niveau bancaire (Argument de vente clé).1 |

### PHASE III : Expansion et Leadership (V3.0) - Scale-up (Mois 25 à 36+)

L'objectif est d'atteindre le statut de plateforme panafricaine de référence et de diversifier les revenus.

|  |  |  |  |
| --- | --- | --- | --- |
| **Module/Composant** | **Fonctionnalités Clés de la V3.0** | **Technologie Impliquée** | **Livrable Stratégique (Output)** |
| **I. Core Accounting** | Gestion des **Immobilisations** (Amortissement, Financements). 9 | Java | Couverture complète des besoins comptables de l'ETI. |
| **II. Intelligence (IA)** | **IA Générative** pour Scénarios Synthétiques et simulations de résistance complexes. 21 | Python (IA Générative) | Innovation continue et avantage concurrentiel non reproductible. |
| **III. Innovation & API** | Lancement de l'**API PREDYKT** (Monétisation Indirecte). 1 | Microservices (Java/Python), API Gateway | Nouveau flux de revenus (API Revenue) et construction d'un écosystème partenaire. |
| **IV. Expansion** | Adaptabilité technique et légale pour l'ouverture dans 2 nouveaux pays stratégiques. 1 | Cloud, Légal | Preuve de la scalabilité régionale de la plateforme.1 |