

# Référentiel de la Stack Technique et de la Gouvernance du Projet HRComplianceTech

1. Introduction et Objectifs du Document .....	<b>Erreur ! Signet non défini.</b>
1.1 Rôle du Référentiel Technique.....	<b>Erreur ! Signet non défini.</b>
1.2 Périmètre et Alignement avec le Cahier des Charges (CdC) ....	<b>Erreur ! Signet non défini.</b>
1.3 Règles de Mise à Jour du Document.....	<b>Erreur ! Signet non défini.</b>
2. Architecture de la Solution et Flux de Données.....	<b>Erreur ! Signet non défini.</b>
2.1 Schéma Architectural Global (Modules 1, 2, 3) .....	<b>Erreur ! Signet non défini.</b>
2.2 Zones de Sécurité et Confidentialité .....	<b>Erreur ! Signet non défini.</b>
2.3 Flux d'Information entre les Modules (API).....	<b>Erreur ! Signet non défini.</b>
Flux 1 : Dépôt d'un Signalement Anonyme (F.1.4) .....	<b>Erreur ! Signet non défini.</b>
Flux 2 : Consultation/Modification (RH/Juriste) .....	<b>Erreur ! Signet non défini.</b>
Flux 3 : Gestion Offline (F.2.7) .....	<b>Erreur ! Signet non défini.</b>
3. Stack de Développement .....	<b>Erreur ! Signet non défini.</b>
3.1 Module 1 : Application de Dépôt (Webapp).....	<b>Erreur ! Signet non défini.</b>
3.2 Module 2 : Application de Gestion (Client Lourd - Windev).....	<b>Erreur ! Signet non défini.</b>
3.3 Module 3 : API Core et Services de Sécurité.....	<b>Erreur ! Signet non défini.</b>
4. Stack de Données et Services Critiques .....	<b>Erreur ! Signet non défini.</b>
4.1 Base de Données Relationnelle (SGBD) .....	<b>Erreur ! Signet non défini.</b>
4.2 Stockage de Fichiers Chiffrés (Pièces Jointes) .....	<b>Erreur ! Signet non défini.</b>
5. Outils et Méthodologie.....	<b>Erreur ! Signet non défini.</b>
5.1 Gestion de Version (Git) .....	<b>Erreur ! Signet non défini.</b>
5.3 Environnement de Développement.....	<b>Erreur ! Signet non défini.</b>
6. Récapitulatif des Versions et des Licences.....	<b>Erreur ! Signet non défini.</b>
6.1 Tableau Récapitulatif des Technologies et Versions Stables....	<b>Erreur ! Signet non défini.</b>
6.2 Contrainte de Licences (C.8.3.5) .....	<b>Erreur ! Signet non défini.</b>
6.3 Synthèse des Outils Open Source.....	<b>Erreur ! Signet non défini.</b>

# 1. Introduction et Objectifs du Document

## 1.1 Rôle du Référentiel Technique

Ce document a pour objectif principal de servir de guide normatif pour l'ensemble des développeurs de la Maitrise d'Œuvre (MOE) (Baptiste et Maxime) tout au long du cycle de vie du projet.

Son rôle est triple :

- **Assurer l'Homogénéité** : Définir une **Stack Technique** (ensemble d'outils et de technologies) unique et validée. Ceci garantit que toutes les parties du projet sont développées avec les mêmes standards et spécifications.
- **Contrôler la Conformité des Versions** : Éviter les **conflits de versions** et les problèmes d'incompatibilité entre les dépendances. Il sert de **base claire** pour l'installation des paquets et l'environnement de développement.
- **Faciliter l'Intégration** : Fournir les bases documentaires nécessaires pour les phases d'intégration et pour les développeurs qui rejoindraient le projet ultérieurement.

## 1.2 Périmètre et Alignement avec le Cahier des Charges (CdC)

Ce référentiel technique couvre la totalité des composants de la solution, conformément aux exigences du CdC :

- **Module 1 : Application de Dépôt / Gestion** (Webapp - React / Prisma)
- **Module 2 : Application de Gestion** (Client Lourd - Windev)
- **Module 3 : Service IA et Traçabilité** (API Python/Flask)
- **Base de Données Centrale** (MySQL)

**Alignement Critique avec le CdC** : Le choix des outils et de leurs versions, détaillé dans ce document, est directement motivé par la nécessité d'honorer les contraintes légales et techniques du Cahier des Charges.

Référence CdC	Exigence	Impact de la Stack Technique
C.8.3.1	Chiffrement Fort (AES-256)	L'architecture doit intégrer des mécanismes cryptographiques pour le chiffrement des données sensibles (géré au niveau applicatif avant stockage).

C.8.3.4	Conformité Juridique (Sapin 2/RGPD)	Le choix de <b>MySQL</b> doit supporter les fonctions d'audit et de traçabilité nécessaires à la conformité.
C.8.3.2	Exigence OS Client Lourd	Le choix de <b>Windev</b> est lié à l'environnement d'exécution <b>Windows 10 Pro ou supérieur</b> exigé.

Les technologies sélectionnées et non négociables pour ce projet sont : **React/Prisma** (Webapp), **Python/Flask** (IA), **MySQL** (Base de données) et **Windev** (Client Lourd).

### 1.3 Règles de Mise à Jour du Document

Le maintien de ce référentiel est crucial pour la qualité du produit final. Les règles de gouvernance sont les suivantes :

- **Responsables de la Validation** : La validation finale du document et de toute modification est la responsabilité de **Baptiste et Maxime** (l'équipe MOE).
- **Déclenchement de la Mise à Jour** : Une révision sera obligatoire lors de conflits de versions ou d'incident technique majeur.
- **Procédure en Cas de Déviation** : Toute solution alternative à un outil spécifié doit faire l'objet d'une mise à jour formelle de ce document avant déploiement.

## 2. Architecture de la Solution et Flux de Données

### 2.1 Schéma Architectural Global

L'architecture est structurée pour garantir la performance de l'application web via un accès direct optimisé aux données, tout en déléguant l'intelligence artificielle à un service dédié.

Module	Rôle Architectural	Technologie(s)
<b>Tiers 1 : Webapp (Module 1)</b>	Interface Utilisateur et Gestion des Données. Interagit directement avec la BDD via son propre Layer ORM.	<b>React</b> (Frontend) et <b>Prisma</b> (DB Layer / Backend Node)

<b>Tiers 1 :</b> <b>Client</b> <b>Lourd</b> <b>(Module 2)</b>	Gestion locale (Offline) et synchronisation.	<b>Windev</b> (Client Lourd)
<b>Tiers 2 :</b> <b>Service IA</b> <b>(Module 3)</b>	<b>Service Spécialisé IA.</b> Dédié à la communication entre l'Intelligence Artificielle et la Base de Données (Classification, Analyse).	<b>Python / Flask</b> (avec SQLAlchemy)
<b>Tiers 3 :</b> <b>Données</b>	Stockage persistant et sécurisé.	<b>MySQL</b> (Données) et <b>Stockage Objet/S3</b> (Fichiers)

**Principe Fondamental Révisé :** La Webapp (Module 1) possède désormais son propre accès à la base de données via **Prisma**, ne dépendant plus de l'API Python pour les opérations courantes (CRUD). Le Module 3 (API Python/Flask) se concentre exclusivement sur les tâches d'IA et de classification en arrière-plan.

## 2.2 Zones de Sécurité et Confidentialité

- **Webapp (React/Prisma) :** Cette zone gère l'authentification des utilisateurs et l'enregistrement des signalements directement en base de données de manière sécurisée. Elle est responsable de la validation des données avant insertion.
- **Service IA (Python/Flask) :** Cette zone accède à la base de données (via SQLAlchemy) uniquement pour récupérer les nouveaux signalements à analyser et pour mettre à jour les métadonnées de classification (IA).
- **Base de Données (MySQL) :** Point de convergence des données, recevant les écritures de la Webapp (Dépôts) et du Service IA (Classifications).

## 2.3 Flux d'Information entre les Modules

### Flux 1 : Dépôt d'un Signalement Anonyme (F.1.4)

1. **Client (Webapp) :** Saisie des données par le déposant.
2. **Webapp → BDD (via Prisma) :** Enregistrement direct du signalement dans MySQL.
3. **Service IA (Module 3) :** Détecte le nouveau signalement (via polling ou mécanisme de trigger).

4. **Traitement IA** : Analyse du contenu et génération de la `category_detectee`.
5. **Service IA → BDD** : Mise à jour du signalement avec les résultats de classification.

### **Flux 2 : Consultation/Modification (RH/Juriste)**

1. **Client (Webapp)** : Authentification de l'utilisateur.
2. **Webapp → BDD (via Prisma)** : Requête de lecture des dossiers via l'ORM Prisma.
3. **Retour** : Affichage des données à l'utilisateur.

### **Flux 3 : Gestion Offline (F.2.7 - Client Lourd)**

1. **Client Lourd (Windev)** : Enregistrement local (buffer) en mode hors ligne.
2. **Client Lourd → BDD** : Synchronisation des données dès la reconnexion (accès direct ou via API de synchronisation dédiée).

## **3. Stack de Développement**

Cette section liste l'ensemble des technologies retenues pour le développement des trois modules.

### **3.1 Module 1 : Application de Dépôt (Webapp)**

Ce module gère désormais l'intégralité du flux utilisateur et l'accès aux données.

Domaine	Technologie	Version Cible	Rôle et Justification
<b>Frontend Framework</b>	<b>React</b>	18.x	Framework leader, assurant l'évolutivité.
<b>Langage</b>	<b>TypeScript</b>	5.x	Typage fort pour la réduction des erreurs et la maintenance.
<b>ORM / DB Layer</b>	<b>Prisma</b>	5.8.1	<b>Nouveau standard.</b> Assure la liaison directe entre la Webapp et MySQL. Simplifie les requêtes et la gestion des migrations de schéma.

<b>Validation</b>	<b>Zod</b>	Dernière stable	Validation de la conformité des données des formulaires.
<b>Design</b>	<b>Tailwind CSS</b>	Dernière stable	Framework utilitaire pour une interface responsive.
<b>Requêtes HTTP</b>	<b>Axios</b>	Dernière stable	Client HTTP pour les appels externes si nécessaire.

### 3.2 Module 2 : Application de Gestion (Client Lourd - Windev)

Ce module correspond au client lourd utilisé par les employés, avec capacité offline.

Domaine	Technologie	Version Cible	Rôle et Justification
<b>Environnement</b>	<b>Windev</b>	Version 28/29	Choix imposé pour l'application de bureau Windows.
<b>Langage</b>	<b>WLanguage</b>	N/A	Langage propriétaire pour la logique métier locale.
<b>OS Cible</b>	<b>Windows 10 Pro</b>	N/A	Répond à la contrainte C.8.3.2.

### 3.3 Module 3 : Service IA et Traçabilité (API Core)

Ce module est recentré sur sa fonction d'intelligence artificielle et de traitement asynchrone.

Domaine	Technologie	Version Cible	Rôle et Justification
<b>Langage</b>	<b>Python</b>	3.10 ou supérieur	Langage polyvalent, excellent pour l'IA et le traitement de données.

<b>Framework API</b>	<b>Flask</b>	2.x	Sert uniquement d'interface de contrôle pour le moteur d'IA et la communication technique interne.
<b>ORM / DB Connector</b>	<b>SQLAlchemy</b>	2.x	Permet au script Python de lire/écrire les résultats de l'IA dans MySQL (seul accès DB pour ce module).
<b>Logique IA</b>	<b>Ollama / Libs Natives</b>	N/A	Module de <b>Pseudo-IA (F.3.3)</b> pour la classification sans dépendance cloud payante.
<b>Modèle IA</b>	<b>BERT / XLM-Roberta</b>	N/A	Modèles performants pour le tri multilingue.

## 4. Stack de Données et Services Critiques

### 4.1 Base de Données Relationnelle (SGBD)

Domaine	Technologie	Justification
<b>Moteur SGBD</b>	<b>MySQL</b>	Moteur open-source robuste.
<b>Hébergement</b>	<b>Aiven.io</b>	Solution DBaaS Managée (Haute disponibilité, RTO & PRA).
<b>Accès Données</b>	<b>Prisma &amp; SQLAlchemy</b>	La base est accédée par <b>Prisma</b> (pour la Webapp) et <b>SQLAlchemy</b> (pour l'IA).
<b>Modélisation</b>	<b>dbdiagram.io</b>	Documentation du schéma DBML.

**Rôle Sécuritaire :** La base de données supporte la traçabilité (audit logs) et stocke les données structurées.

### 4.2 Stockage de Fichiers Chiffrés (Pièces Jointes)

Les pièces jointes sont stockées sur un service de stockage objet.

Domaine	Technologie	Justification
<b>Service Stockage</b>	<b>Cloudflare R2</b>	Recommandé (Compatible S3, pas de frais de sortie). Alternative : AWS S3.
<b>Connexion</b>	<b>AWS SDK / Boto3</b>	Les modules accèdent au stockage via les SDK appropriés (JS pour Webapp, Boto3 pour Python si besoin d'analyse IA sur les fichiers).

## 5. Outils et Méthodologie

### 5.1 Gestion de Version (Git)

- **Plateforme :** GitHub
- **Stratégie :** Gitflow Simplifié (main, develop, feature/\*)

### 5.3 Environnement de Développement

Domaine	Technologie	Module(s) Concerné(s)	Justification
<b>IDE Principal</b>	<b>VS Code</b>	Webapp & IA	Support natif TypeScript (Prisma/React) et Python.
<b>IDE Spécifique</b>	<b>Windev</b>	Client Lourd	Requis pour le WLangage.
<b>Gestionnaire Paquets</b>	<b>npm</b>	Webapp	Gestion des paquets JS/TS et Prisma.
<b>Gestionnaire Paquets</b>	<b>pip</b>	Service IA	Gestion des dépendances Python (Flask, SQLAlchemy).

## 6. Récapitulatif des Versions et des Licences

### 6.1 Tableau Récapitulatif des Technologies et Versions Stables

Composant	Module	Technologie	Version Cible	Rôle Critique / Justification
Webapp ORM	Module 1	Prisma	5.8.1	Layer d'accès aux données (CRUD) pour l'application Web.
Webapp Frontend	Module 1	React	18.x	Interface Utilisateur.
Langage Web	Module 1	TypeScript	5.x	Typage fort (Front & Back).
Validation	Module 1	Zod	Stable	Validation des schémas de données.
Design/CSS	Module 1	Tailwind CSS	Stable	Design rapide et homogène.
Service IA	Module 3	Python	3.11.x	Logique de classification IA.
Framework API	Module 3	Flask	2.x	Interface interne pour le service IA.
IA Connector	Module 3	SQLAlchemy	2.x	Accès DB pour le module Python uniquement.
Pseudo-IA	Module 3	Ollama	Stable	Classification automatique.
Client Lourd	Module 2	Windev	29	Client Offline Windows (C.8.3.2).
Moteur DB	Tiers 3	MySQL	8.0.35	SGBD central.

<b>Hébergement DB</b>	Tiers 3	Aiven.io	N/A	Fournisseur DBaaS.
<b>Stockage Fichiers</b>	Tiers 3	Cloudflare R2	N/A	Stockage Objet compatible S3.
<b>Gestion Version</b>	MOE	GitHub	N/A	Hébergement du code source.
<b>IDE</b>	MOE	VS Code	Stable	IDE principal (React/Python).

## 6.2 Contrainte de Licences (C.8.3.5)

*Inchangé : La MOA est responsable des licences Windev.*

## 6.3 Synthèse des Outils Open Source

Les outils suivants sont Open Source : Python, Flask, SQLAlchemy, **Prisma**, React, TypeScript, Zod, Axios, Tailwind CSS, MySQL, GitHub.