

## LIVRABLE JOUR 1

# Analyse des préliminaires

## Projet A - *FinanceManager Pro*

Le projet **FinanceManager Pro** vise à répondre au besoin du cabinet comptable *Expertise & Conseil* de moderniser les outils de gestion financière utilisés par ses clients PME. Actuellement, ces entreprises travaillent avec des outils dispersés (Excel, logiciels obsolètes), ce qui entraîne une perte de temps, des erreurs et un manque de visibilité sur la trésorerie.

L'objectif du projet est de concevoir une **application web SaaS centralisée** permettant de gérer la **facturation, la comptabilité et la trésorerie** au sein d'une seule plateforme. Cette solution doit offrir aux PME une vision financière claire et en temps réel, tout en permettant au cabinet comptable de suivre efficacement l'ensemble de ses clients depuis un dashboard administrateur dédié.

Le périmètre fonctionnel comprend plusieurs modules : la gestion comptable (écritures, plan comptable, bilan automatique), la facturation (devis, factures, relances et export PDF), la trésorerie (rapprochement bancaire simulé, prévisionnel) et le reporting (tableaux de bord, export Excel et alertes). La plateforme doit également gérer plusieurs profils utilisateurs avec des droits différenciés (gérant, comptable, collaborateur).

Sur le plan technique, l'application doit être **sécurisée, performante et évolutive**. Elle s'appuie sur une architecture web utilisant **Python (Django)** et **PostgreSQL**, avec une authentification renforcée (2FA), un système d'audit des actions et le chiffrement des données sensibles. L'application doit être compatible avec un déploiement cloud, intégrant des mécanismes de sauvegarde automatique et de monitoring.

Le projet se poursuivra par la définition détaillée de l'architecture, de la base de données et par le développement d'un prototype fonctionnel.

# Architecture technique

## 1. Architecture globale

L'application adopte une architecture **client-serveur en trois couches** :

- **Couche présentation** : interface web accessible via un navigateur (desktop prioritaire).

- **Couche applicative** : backend assurant la logique métier, la sécurité et les traitements.
- **Couche données** : base de données relationnelle centralisant les informations financières.

Cette séparation permet une meilleure maintenabilité, une montée en charge maîtrisée et une sécurité renforcée.

## 2. Couche présentation (Frontend)

Le frontend est développé en **HTML, CSS et JavaScript**, avec une interface responsive adaptée principalement aux écrans desktop.

Il permet aux utilisateurs d'interagir avec l'application via des formulaires, tableaux de bord et graphiques.

Les requêtes du frontend transitent exclusivement par le backend via des routes sécurisées, garantissant que les données sensibles ne sont jamais accessibles directement côté client.

## 3. Couche applicative (Backend)

Le backend est développé en **Python avec le Framework Django**, choisi pour :

- Sa robustesse,
- Sa gestion native des utilisateurs et des permissions,
- Ses mécanismes de sécurité intégrés.

Le backend centralise :

- La logique métier (facturation, comptabilité, trésorerie),
- La gestion des utilisateurs et des rôles,
- L'authentification avec **double facteur (2FA)**,
- La traçabilité des actions (audit trail),
- Les échanges avec des services externes simulés (API bancaire, Active Directory).

## 4. Couche données (Base de données)

Les données sont stockées dans une base **PostgreSQL**, adaptée aux volumes de données financières et aux besoins de performance.

La base est conçue selon une logique **multi-entreprises**, chaque donnée étant associée à une entreprise cliente afin de garantir l'isolation des informations.

L'accès à la base est exclusivement réalisé via l'ORM Django, limitant les risques d'injection SQL et améliorant la maintenabilité du code.

## 5. Sécurité et conformité

La sécurité est intégrée à tous les niveaux de l'architecture :

- Authentification sécurisée avec 2FA.
- Gestion des droits basée sur les rôles utilisateurs.

- Chiffrement des données sensibles en base.
- Journalisation des actions utilisateurs (audit trail).
- Communication sécurisée via HTTPS.

Ces mécanismes répondent aux exigences de confidentialité et de traçabilité du cahier des charges.

## 6. Exploitation et déploiement

L'application est conçue pour un déploiement **cloud-ready** sur un environnement Linux.

Des **scripts Bash** assurent :

- Les sauvegardes automatiques de la base de données,
- Les tâches de maintenance,
- La supervision basique du système.

Un système de monitoring permet de suivre l'état de l'application et d'anticiper les incidents.

## Conclusion

Cette architecture technique répond aux contraintes du cahier des charges en proposant une solution **robuste, sécurisée et évolutive**.