Coffre fort numérique

1 - Description

Cette application web constitue un système de stockage sécurisé de documents sensibles destiné aux organisations nécessitant une confidentialité maximale.

Elle propose des fonctionnalités de chiffrement end-to-end, partage temporaire sécurisé, et authentification forte (2FA), sans la complexité ni les coûts des solutions commerciales

1.1 - Périmètre Fonctionnel

Les parcours utilisateurs distinguent trois profils principaux avec des responsabilités clairement définies.

- **Administrateur**
  - Gère les utilisateurs de la plateforme avec possibilité d'ajout, modification et suppression de comptes.
  - Configure les politiques de sécurité globales (durée minimale des mots de passe, activation 2FA obligatoire, durée maximale des liens de partage).
  - Supervise l'activité via un tableau de bord affichant les statistiques d'usage (nombre de documents stockés, utilisateurs actifs, partages en cours).
  - Ne peut jamais accéder au contenu des documents chiffrés grâce à l'architecture Zero-Knowledge.
- **Utilisateur Standard**
  - Dépose des documents sensibles via l'interface web avec chiffrement automatique côté navigateur avant envoi.
  - Organise ses documents dans une arborescence de dossiers personnelle.
  - Consulte ses documents avec déchiffrement transparent côté client.
  - Crée des liens de partage temporaires avec contrôle granulaire (lecture seule, durée de validité, protection par mot de passe optionnel).
  - Active l'authentification à deux facteurs (2FA) via application mobile

TOTP.

- Consulte l'historique complet des accès à ses documents pour détecter toute activité suspecte.

1.2 - Licence

MIT License. Ce choix facilite l'adoption par les organisations en permettant une utilisation commerciale libre tout en encourageant les contributions communautaires. La licence MIT supprime les barrières juridiques qui pourraient freiner le déploiement dans des environnements d'entreprise.

1.3 - Publics cibles

Cabinets d'avocats, études notariales, cabinets d'expertise comptable, centres de recherche, administrations publiques soucieuses de souveraineté numérique, entreprises dans les secteurs régulés (santé, finance), PME souhaitant protéger leur propriété intellectuelle, et particuliers exigeants en matière de vie privée.

2 - Architecture

2.1 - Architecture technique

L'architecture s'appuie sur un modèle Zero-Knowledge où le serveur ne possède jamais les clés de déchiffrement des contenus utilisateur, garantissant une confidentialité maximale.
Le frontend implémente toute la cryptographie côté client via les APIs Web Crypto natives du navigateur, assurant que les documents sensibles sont chiffrés avant transmission et déchiffrés uniquement dans l'environnement contrôlé par l'utilisateur.
Interface web responsive développée en React, Vue.js ou Flask avec intégration de la librairie CryptoJS ou Web Crypto API pour les opérations de chiffrement AES-256-GCM côté navigateur.

Le backend expose une API REST documentée suivant les standards OpenAPI 3.0.
Cette couche orchestre la logique métier essentielle :

- gestion des utilisateurs avec authentification forte (JWT + 2FA TOTP),
- stockage des fichiers chiffrés sur système de fichiers ou stockage objet S3-compatible (MinIO),
- gestion des permissions d'accès avec contrôle granulaire,
- création et gestion des liens de partage temporaires avec expiration automatique,
- API, génération des logs d'audit inaltérables avec horodatage,
- scan antimalware automatique des uploads via VirusTotal API,

Aucune donnée métier en clair ne transite ou n'est stockée côté serveur, garantissant l'architecture Zero-Knowledge.
La couche persistance centralise les données dans une base MySQL relationnelle.

3 - Scénarios d'usage

3.1 - Scénario 1 : Inscription et activation 2FA

Un nouvel utilisateur crée son compte en saisissant ses informations (nom, prénom, email professionnel).
Il choisit un mot de passe robuste respectant la politique de sécurité (minimum 16 caractères, majuscules, minuscules, chiffres, caractères spéciaux).
Le système génère un email de validation contenant un lien d'activation.
Après activation, l'utilisateur accède à son espace personnel et active l'authentification à deux facteurs en scannant un QR code avec une application TOTP (Google Authenticator, Authy).
Il teste la connexion 2FA avec succès et reçoit des codes de récupération d'urgence à conserver précieusement.
Son compte est maintenant protégé par double authentification obligatoire à chaque connexion.
Une paire de clés RSA est générée.
La clé privée stocké dans le navigateur (IndexedDB)
La clé publique est envoyée au serveur.

3.2 - Scénario 2 : Dépôt Sécurisé d'un document confidentiel

Un avocat se connecte avec son mot de passe et son code 2FA.
Il accède à son dossier « Contrats M&A 2024 » et clique sur « Déposer un document ».
Il sélectionne un fichier PDF de 5 Mo contenant un contrat de fusion confidentiel.

Le fichier est envoyé pour vérification, depuis le client via l'API VirusTotal, garantissant qu'aucune menace n'est détectée.

Si le scan est valide, alors le navigateur génère localement une clé de chiffrement symétrique AES-256 appelée DEK (Data Encryption Key).
Le document est chiffré intégralement côté client via la Web Crypto API, en utilisant cette DEK et un vecteur d'initialisation aléatoire (IV).
Une fois le chiffrement terminé :

1. Le navigateur chiffre la DEK avec la clé publique RSA de l'utilisateur (stockée côté serveur lors de la création du compte).
   → Cela produit une clé DEK chiffrée (wrapped key) que le serveur peut conserver sans jamais connaître la clé réelle.
2. Le fichier chiffré, accompagné du IV, de la DEK chiffrée, et de l'empreinte SHA-256 du document original, est alors transmis au serveur.
3. Le fichier chiffré est stocké définitivement dans l'espace sécurisé de l'utilisateur, accompagné de sa DEK chiffrée.

Lors d'un futur téléchargement, le navigateur :

- télécharge le fichier chiffré et la DEK chiffrée,
- déchiffre la DEK avec la clé privée RSA locale de l'avocat,
- puis déchiffre le document grâce à la DEK récupérée.

Ainsi, seul l'avocat — détenteur de sa clé privée RSA — peut déchiffrer le document.
Le serveur n'a jamais accès à la clé AES, ni au contenu du fichier original : aucune donnée en clair ne transite ni n'est stockée.

3.3 - Scénario 3 : Partage temporaire avec partenaire externe

Un expert-comptable souhaite partager un bilan financier confidentiel avec son client pour validation.
Il sélectionne le document « Bilan 2024 - Confidentiel.pdf » dans son coffre-fort numérique et clique sur « Créer un lien de partage ».

L'expert configure les paramètres :

- Durée de validité : 48 heures
- Accès : lecture seule (pas de téléchargement ni capture directe)
- Protection : mot de passe partagé séparément — ici, Client2024!

Le système génère :

- un lien unique du type https://coffre.example.com/share/a3f9d2e8b1c4
- une clé symétrique AES-256 spécifique à ce partage, appelée SEK (Share Encryption Key),
- et un vecteur d'initialisation (IV) aléatoire.

Le document est ensuite re-chiffré côté serveur à l'aide de cette SEK avant toute mise à disposition.
La SEK est elle-même chiffrée avec une clé dérivée du mot de passe de partage (Client2024!), via PBKDF2 (ou Argon2) et un sel aléatoire.

Le serveur conserve :

- le fichier chiffré (AES-256-GCM),
- la SEK chiffrée,
- l'IV, le sel PBKDF2, et les métadonnées du partage (durée, politique d'accès, empreinte SHA-256 du document original).

Aucune clé en clair n'est stockée.

3.4 - Scénario 4 : Accès par le client

Le client reçoit le lien par e-mail et le mot de passe par SMS.

1. Il clique sur le lien, et le navigateur charge la page de visualisation sécurisée.
2. Il saisit le mot de passe Client2024!.
3. Le navigateur dérive localement la clé SEK à partir du mot de passe et du sel PBKDF2 fourni.
4. Le navigateur déchiffre la SEK, puis déchiffre le fichier localement avec AES-256-GCM à l'aide de cette SEK et de l'IV.
5. Le document est affiché directement dans le navigateur en mode lecture seule (rendu PDF dans un conteneur sans bouton de téléchargement, ni impression).

3.5 - Scénario 5 : Expiration et traçabilité

- Le serveur contrôle la validité du lien à chaque requête.
- Après 48 heures, le lien devient totalement invalide : la SEK, le sel, et les métadonnées sont supprimés.
- Les logs d'accès (horodatage, adresse IP, navigateur, identifiant de session) sont archivés pour traçabilité.

3.6 - Scénario 6 : Export sécurisé de clé privée utilisateur

Un avocat souhaite sauvegarder sa clé privée afin de pouvoir la restaurer ultérieurement en cas de changement d'ordinateur ou de perte du navigateur.

Depuis les paramètres de sécurité de son coffre-fort, il clique sur « Sauvegarder ma clé privée ».
Le système lui demande de saisir une passphrase de protection (exemple : MaCle2024!) et lui rappelle que la perte de cette passphrase rendra la clé irrécupérable.

Côté navigateur, les opérations suivantes sont effectuées localement :

- La clé privée RSA (format PKCS#8) est exportée depuis le stockage sécurisé du navigateur (WebCrypto API).
- Une clé de chiffrement dérivée (KEK) est générée à partir de la passphrase via Argon2id (ou PBKDF2 si indisponible), avec un sel aléatoire de 16 octets.
- Un vecteur d'initialisation (IV) aléatoire de 12 octets est produit pour le chiffrement.
- La clé privée est ensuite chiffrée en AES-256-GCM à l'aide de la KEK et de l'IV.
- Le résultat (clé chiffrée) est converti en Base64 pour intégration dans un fichier de sauvegarde au format JSON.

Le fichier final contient :

- la clé privée chiffrée (wrappedPrivateKey),
- le sel de dérivation,
- l'IV,
- les paramètres Argon2id (temps, mémoire, parallélisme),
- l'empreinte SHA-256 de la clé publique associée,
- la date et la version du format.

4 - APIs externes

VirusTotal API : Intégration critique pour la sécurité de la plateforme lors des uploads de documents. Avant l'envoie d'un fichier chiffré dans le coffre-fort, le client envoie une copie temporaire à l'API VirusTotal pour scan antimalware complet. Si le scan détecte une menace potentielle (score de détection supérieur à 3 moteurs), le fichier n'est ni chiffré ni envoyé au serveur et l'utilisateur reçoit un message d'erreur explicite lui demandant de vérifier son fichier. Cette protection empêche la diffusion involontaire de malware via les fonctionnalités de partage.

Les résultats de scan sont conservés en base de données pour traçabilité et audit de sécurité.

Have I Been Pwned API (optionnel) : Vérification de la compromission de l'email utilisateur lors de l'inscription pour renforcer la sécurité du compte.

5.1 - Application Fonctionnelle

5.1.1 - Interface web complète

Application web responsive permettant l'inscription utilisateur avec activation 2FA obligatoire, le dépôt sécurisé de documents avec chiffrement côté client transparent, l'organisation en dossiers avec arborescence personnelle, la consultation avec déchiffrement automatique, la gestion des versions avec historique complet et restauration, la création de liens de partage temporaires avec configuration fine et l'accès aux logs d'audit de sécurité.

L'interface doit être fonctionnelle sur desktop et mobile avec design moderne mettant en valeur les fonctionnalités de sécurité (indicateurs de chiffrement, badges de sécurité, alertes).

5.1.2 - Démonstration opérationnelle

Base de données pré-remplie avec 3 comptes utilisateurs de test (administrateur, 2 utilisateurs standards) pour permettre des démonstrations immédiatement exploitables. Au moins 5 documents exemples déjà déposés avec différents types (PDF, DOCX, images) pour illustrer les fonctionnalités de chiffrement et partage.

5.2 - Documentation

5.2.1 - README complet

Fichier README.md détaillé avec présentation du projet et architecture Zero-Knowledge, prérequis techniques, instructions d'installation locale étape par étape incluant la configuration du stockage objet, configuration des variables d'environnement VirusTotal, SMTP pour emails), configuration du chiffrement côté client (génération des paires de clés RSA), commandes de lancement, et section troubleshooting pour les problèmes courants (erreurs de chiffrement, échecs de scan VirusTotal).

5.2.2 - Documentation API

Documentation complète des endpoints API principaux avec exemples de requêtes et réponses.

Générée automatiquement via Swagger/OpenAPI ou rédigée manuellement en Markdown.

Couvrir au minimum les endpoints de gestion des utilisateurs et 2FA, upload et download de documents chiffrés, gestion des permissions et partages et consultation des logs d'audit.

Inclure des explications détaillées sur l'architecture cryptographique pour faciliter les audits de sécurité externes.

5.2.3 - Guide utilisateur basique

Guide illustré au format PDF ou HTML expliquant les fonctionnalités principales pour chaque profil. Pour les utilisateurs, détailler le processus complet depuis l'inscription avec 2FA jusqu'à la création d'un partage temporaire sécurisé. Pour les gestionnaires, expliquer comment lancer un workflow de signature multi-parties. Pour les administrateurs, documenter la génération de rapports d'audit et de conformité RGPD. Inclure une section "Bonnes pratiques de sécurité" expliquant l'importance de la robustesse des mots de passe, la protection des codes 2FA, et la vigilance face aux tentatives de phishing.

5.3 - Tests et qualité

5.3.1 - Tests unitaires

Tests unitaires sur les fonctions principales avec couverture minimale de 50% mesurée avec pytest et pytest-cov. Priorité aux tests des fonctions critiques de sécurité : chiffrement et déchiffrement AES-256 côté serveur (pour tests uniquement, le chiffrement réel se fait côté client), génération et validation des tokens JWT, validation des codes 2FA TOTP, gestion des liens de partage temporaires avec expiration automatique, intégration DocuSign API et VirusTotal API (avec mocks pour éviter dépendances réseau), et génération des rapports PDF. Tests de sécurité spécifiques pour vérifier la robustesse du hashing Argon2, la génération de tokens cryptographiquement sûrs, et l'impossibilité d'accès aux données chiffrées sans clés valides.

5.3.2 - Repository Git

Code source versionné sur Git avec historique de commits clair et descriptif. Structure de branches simple (main pour la production, develop pour le développement). Code documenté avec docstrings Python conformes PEP 257 pour les fonctions et classes principales, particulier celles impliquant des opérations cryptographiques. Fichier requirements.txt ou pyproject.toml listant toutes les dépendances avec versions exactes. Fichier docker-compose.yml pour faciliter le déploiement avec tous les services (application Python, MySQL, MinIO pour stockage S3, Redis pour cache). Documentation de sécurité SECURITY.md expliquant l'architecture Zero-Knowledge, les algorithmes cryptographiques utilisés (AES-256-GCM, RSA-4096, Argon2id), et les procédures de signalement de vulnérabilités.