

Système de Vérification de Certificats par Blockchain

Rapport de Progression

November 16, 2025

Vue d'Ensemble du Système

Ce Que Nous Avons Fait

État Actuel

Prochaines Étapes

Technologies et Services Utilisés

Catégorie	Technologies	Utilisation dans le Projet
Services ML/IA	ResNet, LayoutLM, XLM-RoBERTa	Modèles pré-entraînés pour la classification de documents, extraction de données et analyse de compétences
API OCR	Tesseract OCR, Python Imaging Library	Extraction de texte multilingue à partir d'images de certificats
Blockchain	Ethereum, Hardhat, Solidity	Smart contracts pour l'émission et la vérification de certificats infalsifiables
Base de Données	PostgreSQL	Stockage des métadonnées, résultats ML et informations utilisateurs
Backend API	Node.js, TypeScript, Express	Orchestration des services et logique métier
Datasets	Kaggle, CASIA, Coursera	Entraînement des modèles de détection de fraude et extraction de compétences

Vue d'Ensemble du Système

Objectif

Construire un système pour vérifier les certificats académiques et professionnels en utilisant la technologie blockchain et l'intelligence artificielle.

Fonctionnalités Clés

- Stockage des certificats sur blockchain
- Vérification de documents assistée par IA
- Détection de fraude par apprentissage automatique
- Extraction de compétences des certificats
- Vérification décentralisée

Composants Principaux

- **API Backend** - Node.js avec TypeScript
- **Smart Contracts** - Blockchain Ethereum
- **Service ML** - Python avec modèles pré-entraînés
- **Frontend** - Interface web basée sur React
- **Base de Données** - PostgreSQL pour les métadonnées

Ce Que Nous Avons Fait

Complété

- Conception de l'architecture système
- Schéma de base de données avec 15 tables
- Spécifications des smart contracts
- Conception de l'API et des endpoints
- Exigences de sécurité
- Conception du pipeline ML

Phase 2: Configuration de l'Infrastructure

Base de Données

- PostgreSQL installé et configuré
- Toutes les tables créées et migrées
- Pool de connexions implémenté

Blockchain

- Nœud local Hardhat en cours d'exécution
- Trois smart contracts déployés
- Tests des contrats terminés

Backend

- Serveur API implémenté
- Intégration base de données complète

Trois Contrats Principaux

1. **CertificateRegistry**

- Émet des certificats sur la blockchain
- Vérifie l'authenticité des certificats

2. **FraudDetectionStore**

- Enregistre les résultats de détection de fraude
- Crée une piste d'audit immuable

3. **SkillValidator**

- Valide les compétences des certificats
- Lie les compétences aux enregistrements blockchain

Trois Services IA Développés

1. Classification de Documents

- Identifie si un document est un certificat
- Utilise le modèle pré-entraîné ResNet

2. Extraction de Texte OCR

- Extrait le texte des images de certificats
- Support multi-langues

3. Extraction de Compétences

- Identifie les compétences mentionnées dans les certificats
- Utilise la correspondance de similarité sémantique

Modèles Pré-entraînés Utilisés

1. ResNet (Residual Network)

- Modèle d'apprentissage profond pour l'analyse d'images
- Détecte la fraude visuelle dans les certificats
- Analyse les motifs, logos et signatures

2. LayoutLM (Layout Language Model)

- Spécialisé pour la compréhension de documents
- Extrait des données structurées des certificats
- Combine texte, mise en page et informations visuelles

3. XLM-RoBERTa (Multilingual Model)

- Compréhension linguistique pour plus de 100 langues
- Extrait les compétences du texte des certificats
- Fonctionne avec les certificats en anglais, français, arabe

Trois Datasets Principaux Sélectionnés

1. Certificate Dataset (Kaggle)

- 1000+ images de certificats
- Pour l'entraînement de la classification de documents

2. Coursera Courses Dataset

- 50,000+ cours avec informations sur les compétences
- Pour l'extraction et la cartographie des compétences

3. CASIA Image Tampering Dataset

- 12,000+ images authentiques et altérées
- Pour l'entraînement de la détection de fraude

État Actuel

Composant	État
Base de Données	Opérationnel
Nœud Blockchain	En Cours
Smart Contracts	Déployés
API Backend	En Cours
Modèles ML	Code Complet

Le backend se connecte avec succès à la base de données et à la blockchain

Prochaines Étapes

Court Terme

- Démarrer le service ML et tester les modèles
- Compléter le développement du frontend
- Intégrer tous les composants
- Tester les workflows de bout en bout

Moyen Terme

- Télécharger et préparer les datasets
- Entraîner les modèles de détection de fraude
- Construire l'interface utilisateur
- Déployer sur le réseau de test

Merci

Questions?