Blockchain-Based Academic Credential Verification System

CS9053 – Introduction to Java
Spring 2025

Mukesh Durga
md6256@nyu.edu

Aditya Nagdekar
an4744@nyu.edu

Tejaswini Ojha
to2226@nyu.edu

April 24, 2025

Abstract

Academic institutions worldwide struggle with issuing and verifying credentials in a secure, efficient, and scalable manner. Traditional systems are centralized, prone to forgery, and require third-party verification agencies. This project introduces a blockchain-based solution where academic certificates are issued, stored, and verified using decentralized technologies. Leveraging Ethereum smart contracts and IPFS, we ensure immutability, transparency, and global accessibility of credentials, implemented using Java-based technologies.

Introduction

In the current digital age, verifying academic credentials remains a slow and manual process. Employers often depend on third-party agencies or direct communication with institutions to validate degrees and transcripts. This introduces inefficiencies, security risks, and high costs.

Blockchain, as a decentralized ledger technology, offers a transformative solution. By recording certificate data as immutable transactions and storing certificate files on a distributed file system (IPFS), we can eliminate fraud and allow instant, trustless verification. Java serves as the primary development environment for our backend system, utilizing Spring Boot for REST APIs, Web3j for blockchain integration, and IPFS clients for distributed file storage. The project emphasizes secure multi-role interaction between universities, students, and employers in a trustless academic ecosystem.

System Architecture

Below is the architecture of our proposed system:

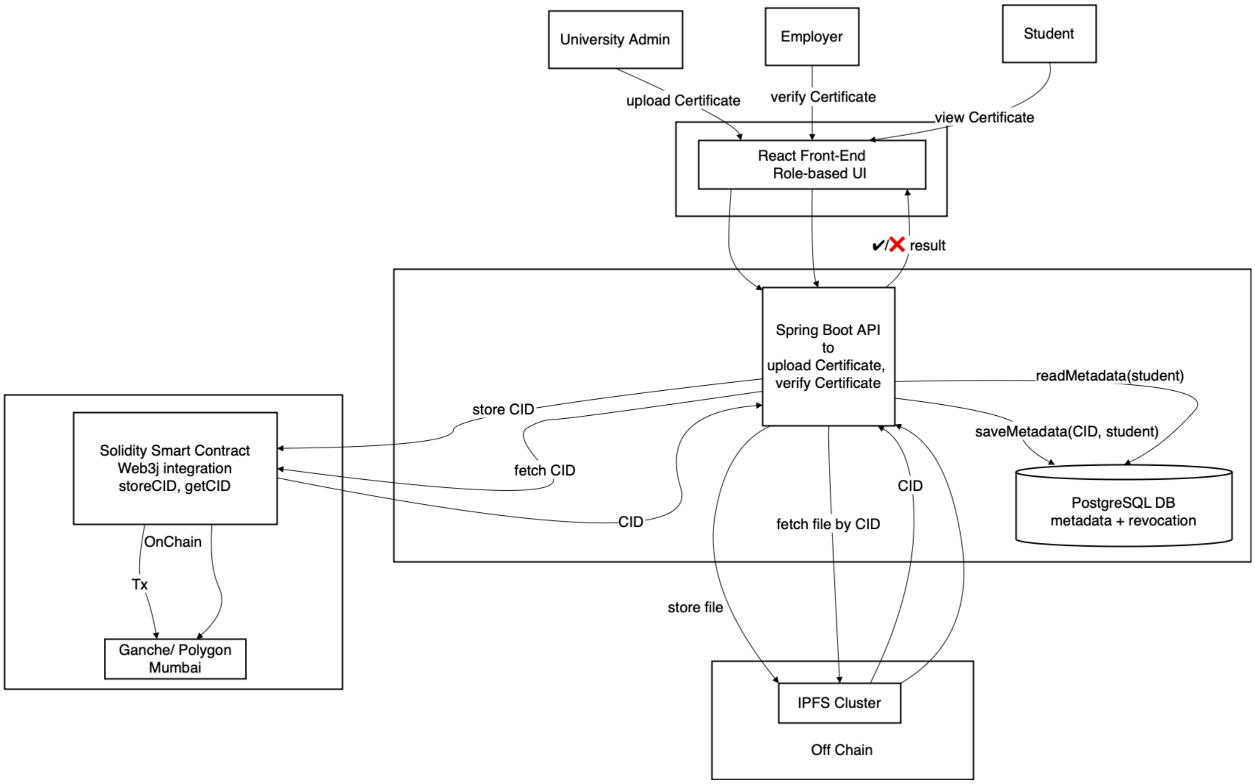

Figure 1: Blockchain-based Academic Credential Verification System

Key Features

- Admin uploads certificate PDF, which is hashed and uploaded to IPFS.
- The returned CID is recorded on the blockchain using Web3j and a smart contract.
- Employers can upload certificates for verification. Hashes are compared with CID on-chain.
- Role-based access: Admin (issue), Student (view), Employer (verify).
- Optional: Maintain logs and analytics via a local database.

Technologies Used

Layer	Tools
Backend	Java Spring Boot, Web3j
Blockchain	Ganache / Polygon Mumbai
Smart Contract	Solidity (Remix/Hardhat)
Storage	IPFS (Pinata/web3.storage)
Frontend	React.js / Postman / JavaFX
Database	PostgreSQL / H2 (Optional)
Wallet	Metamask
Build Tools	Maven / Gradle

Advanced Java Concepts

- **Multithreading**: Handle multiple uploads/verifications.
- **Networking**: RESTful APIs and blockchain interaction.
- **Custom Data Structures**: Manage certificate and user metadata.
- **File Handling**: SHA-256 hashing, uploads, IPFS interactions.
- **Spring Security**: Role-based session management.
- **Library Integration**: Web3j and IPFS libraries.

Team Roles

- **Mukesh Durga**: Develops backend REST APIs, integrates third-party services, and manages file handling routines.
- **Aditya Nagdekar**: Bridges frontend and backend systems—implements API endpoints and integrates React components with Spring Boot services.
- **Tejaswini Ojha**: Designs and deploys smart contracts, orchestrates Web3j integration, and administers IPFS storage workflows.

Outcome & Benefits

- Fast, tamper-proof academic credential verification.
- Eliminates the need for third-party verifiers.
- Self-sovereign identity for students.
- Real-world blockchain use case using Java.

Future Scope

- Add QR code verification support.
- Implement certificate expiration and renewal.
- Analytics dashboard for admin users.

References

- Solidity on Remix IDE
- Web3j Documentation
- IPFS Docs
- Ganache by Truffle Suite