Digital Identification System using Blockchain Technology

KLE Technological University
Creating Value Leveraging Knowledge

Anurag C[1] Rohan K[2] Revati B [3] Mrunali B [4]

[1]Project Guide: Prof Asharani Patil

Abstract

This project introduces a Blockchain-Based Digital Identity System for university-level student identity management and document verification. Built on Ethereum, it ensures data integrity, security, and transparency using smart contracts for student registrations, document uploads, and verification. MongoDB manages non-blockchain data, while Ganache and MetaMask enable secure interactions. The system features user-friendly interfaces for students and administrators, promoting decentralized identifiers (DIDs) and verifiable credentials to enhance privacy and scalability, addressing limitations in traditional verification methods.

Problem Statement

Traditional student identity management and document verification systems are centralized, inefficient, and prone to tampering. They lack transparency, scalability, and data security, leading to challenges in verifying the authenticity of academic credentials. A decentralized and secure solution is needed to streamline these processes and ensure data integrity.

Objectives of the project

1. **Develop a Decentralized System:** Create a blockchain-based platform to manage student identities and academic document verification, ensuring data integrity and transparency.
2. **Ensure Data Security and Privacy:** Utilize decentralized identifiers (DIDs) and verifiable credentials to protect sensitive student information and academic records.
3. **Streamline Verification Processes:** Automate student registration, document upload, and verification workflows using Solidity smart contracts.
4. **Enhance Scalability:** Design a solution capable of handling a growing volume of academic records efficiently.
5. **Facilitate User-Friendly Interaction:** Build intuitive web interfaces for students and administrators to perform their roles seamlessly.
6. **Promote Trust and Authenticity:** Leverage Ethereum blockchain's immutability to guarantee the authenticity of verified credentials.

Project Requirements

Functional requirements:

1. For Students:
   - **Secure Registration and Login:** Students must register using university-provided credentials, such as email or unique student IDs. A one-time authentication mechanism and optional two-factor authentication (2FA) ensure secure access.
   - **Certificate Upload:** Students should be able to upload scanned or digital certificates in secure formats like PDF or JPEG. The system should validate file types and sizes to prevent invalid or malicious uploads.
   - **Track Verification Status:** A dashboard must provide real-time tracking of certificate verification, displaying statuses like "Pending," "In Review," and "Verified." Students should receive notifications or alerts for any updates.
   - **Download Verified Certificates:** Students should be able to download certificates authenticated via blockchain. Each certificate must have a unique identifier for easy validation and authenticity checks.
   - **View Dashboard:** A personalized dashboard should summarize uploaded certificates, their current status, and verified certificates available for download.
2. For Admins:
   - **Admin Login:** Admins must log in securely using designated credentials to access the admin panel.
   - **Review and Verification Tools:** Admins must have tools to view and manage certificate requests. They should be able to filter requests by criteria like date, certificate type, or student ID and perform bulk actions for efficiency.
   - **Certificate Approval:** Admins should have the ability to verify uploaded certificates, approve or reject them, and provide feedback or request resubmission for incomplete or incorrect details.
   - **Manage Blockchain Records:** Admins must ensure verified certificates are securely stored on the blockchain using cryptographic hashes. Metadata such as student details and verification timestamps should be stored off-chain in MongoDB.
3. System Features:
   - **Immutable Storage:** All verified certificates must be stored on the Ethereum blockchain to ensure they cannot be tampered with or deleted.
   - **Unique Transaction IDs:** Each verified certificate should have a unique transaction ID and timestamp for authenticity and easy traceability.
   - **Decentralized Identifiers (DIDs):** The system must assign a DID to each student, enabling self-sovereign identity management.
   - **Transparency and Auditability:** Blockchain records must provide a transparent, tamper-proof log of all verification actions, ensuring accountability for both students and administrators.

Design

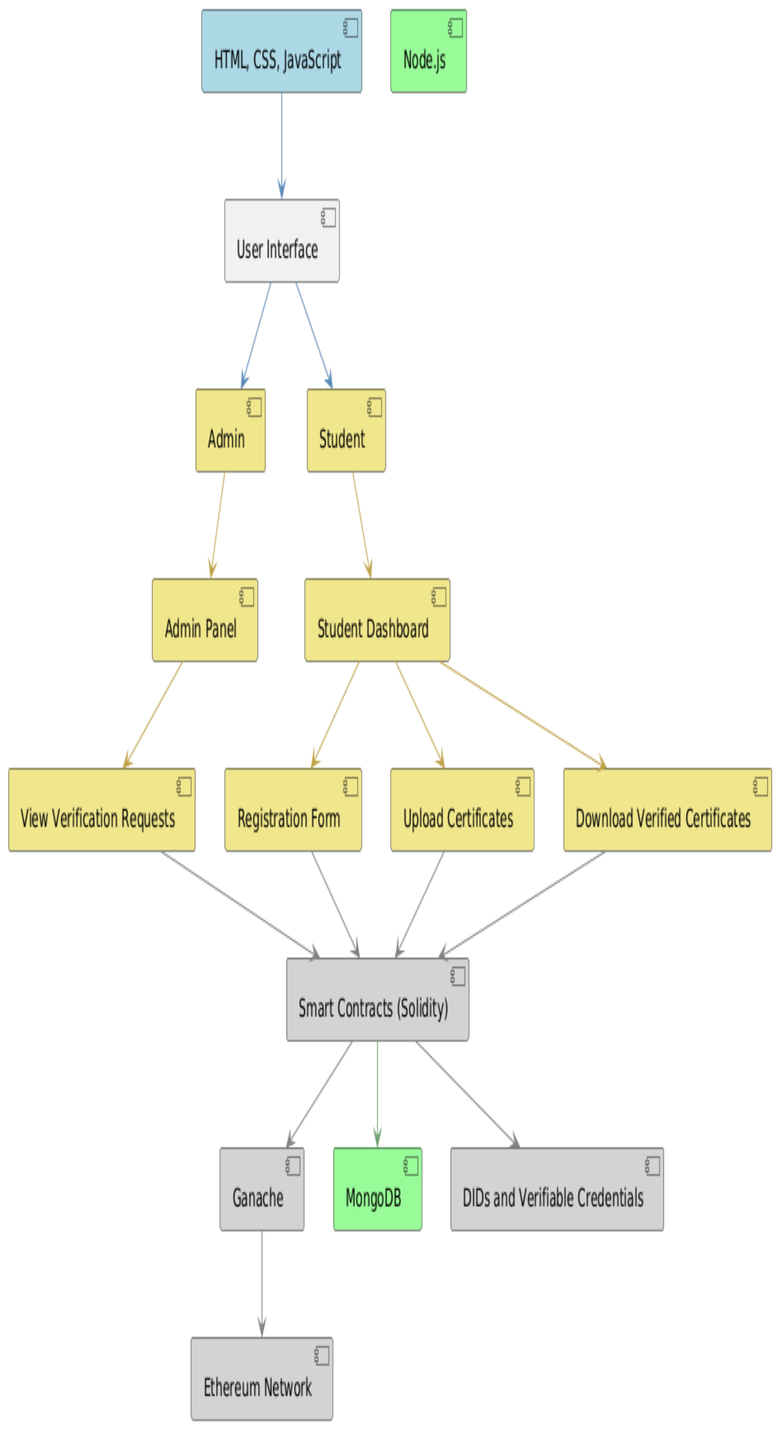

Figure 1. Blockchain-Based Digital Identity System Design Diagram

Implementation and Testing

Implementation Strategy:

- Smart Contract (Solidity):

The smart contract handles certificate uploads and verification. It stores certificates with a unique identifier, their hash, and their verification status on the Ethereum blockchain. Functions include uploading certificates, verifying them, and rejecting them.

- Backend (Node.js & Express):

A RESTful API is created to interact with the Ethereum smart contract and MongoDB database. `web3.js` is used to communicate with the Ethereum blockchain. MongoDB stores additional data, such as student details and certificate metadata.

- Frontend (HTML, CSS, JavaScript):

Students can upload certificates through a simple, user-friendly interface. The frontend interacts with the backend API to upload certificates and display their verification status. File hashes are generated and used to ensure that the certificate content is unique and tamper-proof.

Testing Strategy:

- Backend Testing:

- **Unit Testing:** Used to test the REST API endpoints.
- **Example:** Verify that certificates can be uploaded and retrieved from MongoDB.
- **Testing Tools:** Postman is used to write test cases.
- **Sample test:** Ensures that uploading a certificate returns the correct status.

- Frontend Testing:

- **Manual Testing:**
  - Verify the certificate upload process works smoothly, including hash generation and API communication.
  - Ensure UI/UX flow for students and admins is functional.

- System Testing:

- **Ganache:** Local Ethereum blockchain for smart contract deployment and testing.
- **Truffle:** A development framework for Ethereum to deploy and test smart contracts.
- **Sample test:** Verify that the smart contract stores certificate data correctly on the blockchain.

Results and Discussion

Category	Result
Performance	Fast uploads
Smart Contracts	Efficient certificate handling
Blockchain Storage	Secure and tamper-proof

Table 1. Key Findings

Challenge	Issue
Blockchain Delays	Network congestion delays
Gas Fees	Increased transaction costs
UI/UX Issues	Upload process difficulties

Table 2. Challenges Encountered

System Type	Key Feature
Blockchain-based	Fast and secure verification
Traditional	Time-consuming and prone to errors

Table 3. Comparison with Traditional Systems

Limitation	Impact
Network Constraints	Delays and high costs
Smart Contract Limits	Scalability and complexity issues

Table 4. System Limitations

Enhancement	Details
Scalability	Layer 2 solutions
Improved Security	More robust cryptographic methods
Automated Verification	AI-driven verification processes

Table 5. Future Enhancements

The analysis of the project outcomes reveals the effectiveness and limitations of the blockchain-based certificate verification system. The following key insights were derived:

- **Performance and Security:** The system demonstrated fast certificate uploads and verification processes, ensuring efficiency in handling user data. The tamper-proof nature of blockchain ensures that certificates remain secure and immutable, offering a significant advantage over traditional systems.

- **Challenges Encountered:** Despite its advantages, the project faced challenges such as blockchain network delays and high gas fees during peak usage times. These factors highlighted the need for optimization strategies to enhance performance and reduce costs.

- **Comparison with Traditional Systems:** The blockchain-based system provides faster, more secure, and error-free verification compared to traditional paper-based or centralized digital methods. However, it is necessary to address scalability concerns to support wider adoption.

- **System Limitations:** Network constraints and limitations in smart contract design posed scalability and complexity issues. These constraints emphasize the importance of incorporating advanced solutions like Layer 2 scaling or sidechains.

- **Future Enhancements:** The proposed improvements include Layer 2 solutions for scalability, AI-driven verification, and stronger cryptographic methods, aiming to enhance the system for broader networks. In conclusion, the results demonstrate the feasibility of using blockchain for certificate verification, with potential for ongoing enhancements to overcome current limitations.

Conclusion

The project demonstrated a secure and decentralized blockchain-based certificate verification system. Smart contracts ensured transparency, while a user-friendly interface enhanced accessibility. Testing revealed the need for improved scalability and optimized execution, emphasizing blockchain's potential for efficient document management.

References

[1] Amal Abid, Saoussen Cheikhrouhou, Slim Kallel, and Mohamed Jmaiel. Novidchain: Blockchain-based privacy-preserving platform for covid-19 test/vaccine certificates. *Software: Practice and Experience*, 52(4):841–867, 2022.

[2] Haifa Alanzi and Mohammad Alkhatib. Towards improving privacy and security of identity management systems using blockchain technology: A systematic review. *Applied Sciences*, 12(23):12415, 2022.

[3] May Htet, Phyo Thet Yee, and Jay R Rajasekera. Blockchain based digital identity management system: A case study of myanmar. In *2020 International Conference on Advanced Information Technologies (ICAIT)*, pages 42–47. IEEE, 2020.

[4] Ibrahim Tariq Javed, Fares Alharbi, Badr Bellaj, Tiziana Margaria, Noel Crespi, and Kashif Naseer Qureshi. Health-id: A blockchain-based decentralized identity management for remote healthcare. In *Healthcare*, volume 9, page 712. MDPI, 2021.

[5] Chia-Hung Liao, Xue-Qin Guan, Jen-Hao Cheng, and Shyan-Ming Yuan. Blockchain-based identity management and access control framework for open banking ecosystem. *Future Generation Computer Systems*, 135:450–466, 2022.

[6] Andreea-Elena Panait, Ruxandra F Olimid, and Alin Stefanescu. Identity management on blockchain–privacy and security aspects. *arXiv preprint arXiv:2004.13107*, 2020.

[7] Xiaoyang Zhu and Youakim Badr. Identity management systems for the internet of things: a survey towards blockchain solutions. *Sensors*, 18(12):4215, 2018.

[8] Andrej J Zwitter, Oskar J Gstrein, and Evan Yap. Digital identity and the blockchain: universal identity management and the concept of the "self-sovereign" individual. *Frontiers in Blockchain*, 3:26, 2020.

02fe22bcs055@kletech.ac.in