Team 5 Project Proposal: Decentralized Identity and Access Management (IAM) dApp

Brijay Shetty
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States
bshetty3@asu.edu

Ishan Mandlik
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States
imandlik@asu.edu

Molly Montoya
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States
msmonto3@asu.edu

Nisarg Sanjaykumar Shah
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States
nshah132@asu.edu

Vijay Jagdish Karanjkar
School of Computing and Augmented Intelligence
Arizona State University
Tempe, United States
vkaranjk@asu.edu

***Abstract*—We propose a decentralized identity and access management (IAM) application that allows users to create, own, and control their digital identities without relying on centralized authorities. We will create the system using Ethereum smart contracts to register decentralized identifiers (DIDs), issue verifiable credentials, and maintain immutable verification and revocation records. The credential data is stored off-chain using the InterPlanetary File System (IPFS). A web-based decentralized application (dApp) built with React and Ethers.js will allow for interaction between users, issuers, and verifiers through MetaMask wallet authentication. By using blockchain and self-sovereign identity, the system improves transparency, user ownership, and security in managing digital identities.**

I. Introduction / Problem Statement

Digital identification plays a crucial role in an individual's online presence and interactions. Relying on centralized authorities, such as governments, universities, or corporations, to create, manage, and verify digital identities has several drawbacks: 1. a single point of failure, 2. limited control for individuals, and 3. inefficiencies in the verification process.

This project aims to address these issues by providing a Decentralized Identity and Access Management system that enables self-sovereign identity (SSI). This approach empowers users to create and own their individual identities, eliminating dependency on centralized authorities.

II. Motivation

This innovative system is designed to empower users with complete autonomy over their digital identities, enabling them to control, own, and share their personal information as they see fit. By leveraging blockchain technology, the system significantly enhances transparency in the issuance of credentials and verification processes. All transactions and changes to identities are recorded on a public ledger, allowing for independent audits and fostering trust among users and organizations.

To ensure robust security and privacy of sensitive personal data, the system incorporates advanced techniques such as hashing and cryptography. Hashing transforms data into a fixed-size string, making it unreadable and protecting it from unauthorized access. Together, these technologies provide users with a secure environment to manage their identities while safeguarding their privacy in an increasingly digital world.

Such a decentralized solution can be applied across various domains, including education, healthcare, and fintech.

III. Literature Review

Decentralized identity management's use in blockchain in recent years has been extensively studied. Interoperability, scalability and privacy are the main issues focused and addressed by researchers.

A. Standards and Decentralized Identity Models

Verifiable Credential standards and Decentralized Identifier (DID) were introduced by World Wide Web Consortium (W3C) that form the foundation of modern self-sovereign identity (SSI) frameworks. uPort, Sovrin, and Hyperledger Indy are some of the platforms that removes reliance on centralized providers and enable users to create and manage

their own digital identities [1]. Public-key cryptography and distributed ledgers are leveraged by this systems to provide credential issuance and secure identity verification.

B. Blockchain as a Trust Layer

Use of blockchain as a decentralized access control manager was proposed by Zyskind et al. [2], where through smart contracts users can control personal data sharing. Similarly, blockchain's distributed consensus mechanisms and immutability as demonstrated by Hardjono et al. [3] reduces dependency on centralized authorities by serving as global trust anchor for authorization and authentication.

C. Off-Chain Storage Integration

Blockchain's storage limitations led to many studies proposing hybrid architectures combining off-chain data storage solutions with on-chain records such as the InterPlanetary File System (IPFS). such systems improve efficiency and scalability while maintaining verifiability through cryptographic hashes stored on-chain as highlighted by Wang et al. [4]

D. Privacy-Preserving Mechanisms

Current research on Zero Knowledge proofs is promising and proposes a selective disclosure protocol to enhance privacy in decentralized identity system. Chaudhary et al. [5] have developed a ZK-SNARK based verification method which allows users to prove specific attributes like nationality, age etc. without disclosing private credentials. They claim to ensure General Data Protection Regulation compliance and aligns with privacy preservation principles.

E. Identified Research Gaps

There are certain challenges regarding interoperability between frameworks, usability and governance but nevertheless, blockchain based identity systems shows promise for the future. The current Self-Sovereign Identity implementations lack standardized revocation mechanism and have other limitations like complex key management system. Our project aims to address these gaps by desigining a Ethereum base IAM framework that integrates verifiable credentials, decentralized identification and IPFS storage.

IV. Proposed Architecture

A. System Overview

The proposed system is composed of three primary layers that work together to enable decentralized identity management: the on-chain layer, the off-chain layer, and the application layer.

- On-Chain Layer: Smart contract on Ethereum Blockchain to manage credential status, event logging.
- Off-Chain Layer: IPFS to store issued credentials securely and link it to the blockchain entries using hashing.
- Application Layer: A user interface to enable interaction with the system

B. On-Chain Layer

This layer is deployed on the Ethereum test network (Sepolia or Goerli) and implemented using Solidity smart contracts.

- **DID Registry Contract:** Registers and manages decentralized identifiers (DIDs), storing minimal metadata such as public keys and service endpoints. Emits events for updates or key rotations.
- **Credential Status Contract:** Maintains hashes or Merkle roots of issued credentials to support revocation, verification, and status updates.
- **Event Logging Contract:** Records credential verification and access events immutably on-chain for auditability, without storing any personal information.

C. Off-Chain Layer

The off-chain layer handles large or sensitive data that should not reside directly on the blockchain.

- **IPFS Storage:** Stores complete credential files in JSON-LD format. The on-chain smart contracts only reference the IPFS hash to ensure data integrity.
- **Optional API Server:** A lightweight Node.js or Express server may be used to cache credential metadata and expose REST or GraphQL endpoints to improve lookup performance.

D. Application Layer

The user-facing component is a web-based decentralized application (dApp) built using React and Ethers.js. It enables three main roles:

- **Holder:** Creates a DID, stores credentials, and shares proofs with verifiers.
- **Issuer:** Issues and signs verifiable credentials, uploading them to IPFS and recording hashes on-chain.
- **Verifier:** Requests credential proofs and verifies them through the smart contracts.

Users interact with the blockchain via the MetaMask wallet for key management and authentication.

E. Security and Privacy Layer

To ensure user privacy, sensitive data is hashed before being written on-chain. Public and private keys are managed locally by the user through MetaMask. Future extensions may incorporate zero-knowledge proofs to enable selective disclosure, such as proving age or citizenship without revealing full personal details.

V. Chosen Platform

We will be using the following open-source blockchain and web technologies.

1) **Ethereum**: To deploy and test smart contracts. It supports dApp's and has reliable tools for Identity and Access Management.
2) **Solidity**: For writing smart contracts and used alongside Hardhat.
3) **React.js, Ethers.js, Web3.js**: For creating frontend and integration with the Ethereum blockchain.

4) **Metamask**: For wallet authentication and transaction management.
5) **InterPlanetary File System (IPFS)**: For off chain storage of credential data, while only saving cryptographic hashes on-chain for privacy preservation.
6) **Windows OS, VSCode, Node.js, Git**: Windows with VSCode will be used as the development environment and Git will be used for version control and team collaboration.

REFERENCES

[1] W3C, "Decentralized Identifiers (DIDs) v1.0," 2022. [Online]. Available: https://www.w3.org/TR/did-core/

[2] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Security and Privacy Workshops*, 2015, pp. 180–184.

[3] T. Hardjono, A. Pentland, and D. Shrier, "Trust::Data — A New Framework for Identity and Data Sharing," MIT Connection Science, 2019.

[4] C. Wang, X. Luo, and S. Zhang, "A Blockchain-Based Decentralized Identity Management with Off-Chain Storage," *IEEE Access*, vol. 9, pp. 66535–66546, 2021.

[5] A. Chaudhary, K. Raza, and P. Rana, "Zero-Knowledge Proof Based Privacy-Preserving Identity Verification Using Blockchain," in *Proc. IEEE ICC*, 2020.

Team Contract

Team 5 — Decentralized Identity and Access Management (IAM) dApp

Roles

All team members share responsibility for planning, communication, performing literature survey, and ensuring project milestones are met.

- Blockchain Developer: Implements Solidity smart contracts for decentralized identifiers (DIDs), credential status, and event logging.
- Frontend Developer: Builds the web interface using React and Ethers.js, integrating MetaMask for wallet-based authentication.
- Backend/Integration Engineer: Connects the frontend to IPFS and manages Node.js or API services for data storage and retrieval.
- Tester/Documentation Lead: Conducts testing, quality assurance, and maintains technical documentation.

Expectations

- Every member contributes equally to development, documentation, and testing.
- Tasks will be divided fairly and tracked through GitHub.
- Members are expected to respond to messages within 24 hours and attend scheduled meetings.
- Pull requests must include clear commit messages and undergo peer review before merging.
- Team members will maintain open communication, be respectful, and help resolve conflicts constructively.
- If someone cannot meet a deadline, they will inform the team ahead of time so adjustments can be made.

Meeting Frequency

- Weekly Team Meeting: Once per week to discuss progress, assign tasks, and review goals.
- Check-Ins: Quick updates over WhatsApp during the week.
- Milestone Reviews prior to major due dates.

Name	Responsibilities	Signature	Date
Molly Montoya	Frontend development, Documentation	*Molly Montoya*	10/23/2025
Ishan Mandlik	API development, Storage Integration	*Ishan Mandlik*	10/23/2025
Vijay Jagdish Karanjkar	API development, Storage Integration	*Vijay Jagdish Karanjkar*	10/23/2025
Nisarg Shah	Chain development, wallet integration	*Nisarg Shah*	10/23/2025
Brijay Shetty	Frontend development, Testing	*Brijay Shetty*	10/23/2025