WEB3ID

Senior Design

Edward Alkrire, Abhisek Kumar Jha

Abstract

Create and share "digital identities" with selective disclosure of Personally Identifiable Information (PII). This includes support for attestations verified by the service and associated with the user using cryptographic signature verification and immutable public blockchain history.

Background

The problem with treasury management and salary agreements in the prevailing financial system is the lack of transparency and the accompanying trust assumptions. Consider a non-profit organization like Wikipedia, which misleads users into believing they wouldn't exist without regular donations. Their operational costs are relatively small, and they have hundreds of millions of dollars in assets, enough to operate indefinitely without additional donations. More credible charities could benefit from transparent treasury management, attracting donations by proving their need and history of appropriate fund management.

Salary agreements are another area where trust is abused in the current financial system. Employees must trust their employer's brand to compensate them as expected. This becomes problematic when companies mismanage funds or go bankrupt, leaving employees without due compensation. Relying on reputation is also a challenge for startups trying to attract talent. Even if a startup offers a competitive salary, prospective employees are asked to take a risk on a company with no established reputation or trust that the startup will have the funds to pay them long-term.

Traditional salary agreements are prone to human error and emotional biases, potentially leading to disputes and legal intervention. For example, the average wrongful termination settlement is $40,000. Traditional salary agreements are burdened by unpredictability and unnecessary costs, whether legally related or due to human nature.

That is why we are building a system to create and share digital identities with selective disclosure of PII, including support for attestations verified by the service to be associated with the user using cryptographic signature verification and immutable public blockchain history

Key Requirements

Functional Requirements:

- **User Interface**: Intuitive and user-friendly design for ease of access.
- **Account Management:** Support account creation and secure login for users.
- **Financial Data:** Real-time aggregation & historical insights of financial data.
- **Search and Sharing:** Publicly searchable accounts with options for sharing.

Technical Requirements:

- **Frameworks:** Used Next.js
- **Security:** Implemented encryption and access control

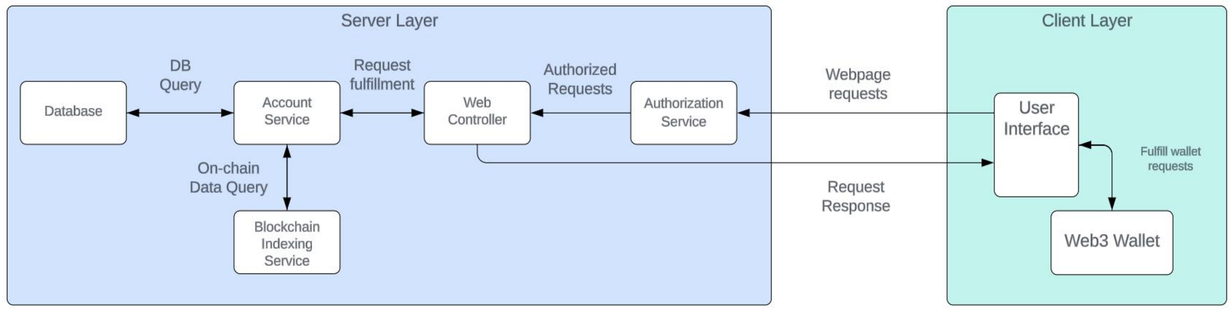
Non-Functional Requirements:

- **Performance:** Ensure scalability, responsiveness, and high availability.
- **Usability:** It complies with accessibility standards.
- **Compliance:** It adheres to the GDPR

Additional Considerations:

- **Proper Documentation:** It provides the user with all the user and technical documentation.
- **Support:** It offers customer support and future updates.

Detailed Design Phase



System Overview

Client Layer:

- **User Interface:**
  This layer provides the interface for users to interact with the Smart Pay application. It includes web pages and forms for creating and managing smart contracts, viewing organization information, checking on-chain events, and accessing user-specific data.
- **Web3 Wallet:**
  Users can connect their Web3 wallet to the application, allowing them to securely manage their cryptocurrency funds and interact with the blockchain.

Server Layer:

- **Database:**
  Manages user account authentication, user profile storage, payment history, and other session-related data. It stores information that is not suitable for storage on the blockchain.
- **Account Service:**
  Responsible for managing user accounts, including user registration, login, and profile management.
- **Web Controller:**
  Handles incoming requests from the user interface, processes them, and interacts with other services and the blockchain layer as necessary.
- **Authorization Service:**
  Ensures that users have the necessary permissions to access certain features and data within the application.
- **Blockchain Indexing Service:**
  Interacts with the blockchain layer to index and retrieve data from smart contracts and on-chain events. It improves application performance by storing and processing blockchain data in a format that is easily accessible to other components.

Conclusion and Future Works

- We plan to establish standards and APIs for cross-platform and jurisdiction integration.
- We will design intuitive interfaces and educational resources for better user understanding and management.
- We will perform security audits and continuous monitoring for vulnerabilities and their mitigation.

References

•**Decentralized Identifiers (DIDs) v1.0,** Core architecture, data model, and representations

•Verifiable Credentials Data Model v2.0, W3C Candidate Recommendation Draft 24 July 2024