PrIde - Private Identity

Aishwarya Rastogi, Ronak Shah, Samkit Sheth and Siddhant Sribhashyam
Department of Software Engineering, San José State University
San José, CA
aishwarya.rastogi@sjsu.edu, ronak.shah@sjsu.edu, samkitraseshbhai.sheth@sjsu.edu, siddhant.sribhashyam@sjsu.edu

***Abstract*— In today's world where data privacy is an everyday concern, the app aims to provide granular control of identity information to any organization. The issuing organization will store a signed identity schema in the decentralized blockchain and will issue a identity certificate to the user. The user can then present the certificate to a verifying organization as proof. Whenever an organization requests some identity information like age, place of birth, etc., using zero-knowledge proof the user can share only the required information with the verifier. The verifier can then validate the given proof from the decentralized blockchain thus being sure of the authenticity of the issuer and information. This would help reveal only the data that is required to the concerned party and keep all the other data hidden, while also maintaining data integrity and authenticity.**

I. Introduction

PrIde is a platform for organizations to be used for issuing identity credentials and verifying them. PrIde also provides granular control over the identity information to the individuals.

II. Architecture

There are 5 major components of the project. HyperLedger Indy, Blockchain Server, PrIde API, User Database, Frontend Application.

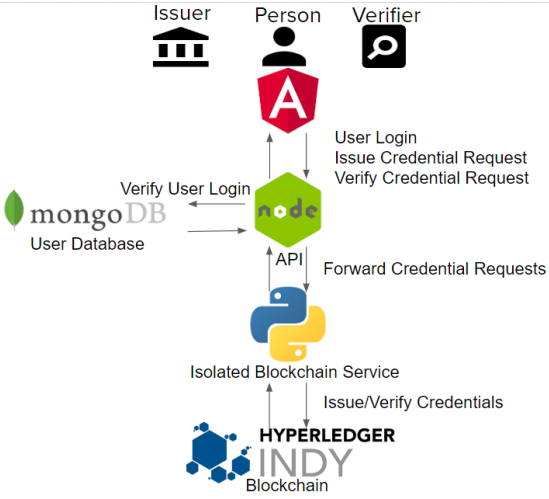

Fig. 1. Project Architecture

III. HyperLedger Indy

Hyperledger Indy is a distributed ledger, purpose-built for decentralized identity. Hyperledger Indy provides tools, libraries, and reusable components for providing digital identities rooted on blockchain. Hyperledger Indy can be used to implement the most crucial concepts in private identity management - Zero Knowledge Proofs and Self Sovereign Identity.

A. Zero Knowledge Proofs

A Zero-Knowledge Proof consists of using cryptography for authentication in such a way that allows one entity to prove to another entity that they know a certain information or meet a certain requirement without having to disclose any of the actual information that supports that proof. The entity that verifies the proof has thus "zero knowledge" about the information supporting the proof but is "convinced" of its validity. This is useful when and where the prover entity does not trust the verifying entity but still has to prove to them that he knows a specific information. For example, one could prove that they are over 21, without showing their exact date of birth.

B. Self Sovereign Identity

Self-sovereign identity (SSI) is a term used to describe the digital movement that recognizes an individual should own and control their identity without the intervening administrative authorities. SSI allows people to interact in the digital world with the same freedom and capacity for trust as they do in the offline world.

A blockchain consisting of multiple nodes was created and managed using the Indy framework. These nodes were connected to make a Verifiable Organizations Network(VON) which processes all the identity transactions in the system.

IV. Blockchain Server

A server that acts as a wrapper for the blockchain was developed using Python for creating, transmitting and storing verifiable digital credentials on the blockchain. An isolated application programming interface for the blockchain was developed which provides various functionalities like - issuing credentials, verifying credentials, creating credential schemas, etc.

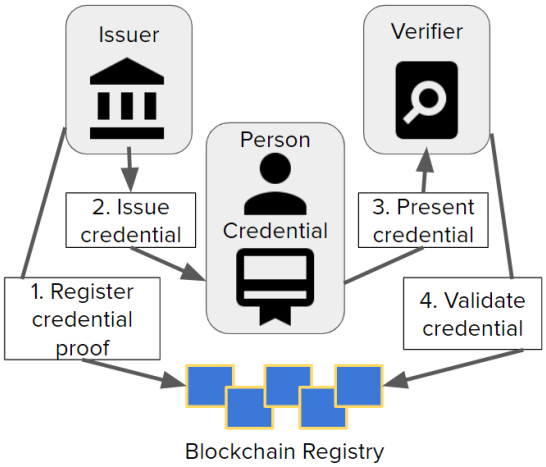

Fig. 2. Workflow Diagram

V. PrIde API

An application programming interface was developed using NodeJS to bridge the gap between frontend applications and the blockchain server. The decision of isolating the blockchain server and frontend application was taken due to security concerns. The functionalities provided by this API are user management, session management and middleware for the blockchain server. This API server is connected with the User Database for authenticating user access to the frontend.

VI. User Database

A NoSQL database was used for maintaining user access to the system. MongoDB was used because of its cloud based database as a service offering. As the User data is not relational, a NoSQL database was preferred over the traditional relational database.

VII. Frontend Application

Frontend Application was developed using the latest version of Angular framework. Material Design template was used throughout the application to give it a consistent look. Application was integrated with the Node based REST API.

VIII. Application Workflows

A. Issuing Organization Workflow

Admin users of Issuing Organizations like DMV, Universities, etc can login to manage the issuance of credentials. Organizations can create new schemas for credentials like Driver License, Transcript, SSN, etc. These schemas can be used to issue encrypted identity credentials on the blockchain.

B. Individual Workflow

Individuals can login to the system to view their credential wallet. The credential wallet contains all the identity credentials that were issued to the user by different organizations on the Verifiable Organizations Network(VON). All of the credential attributes are visible to the user and the user has granular control over the attributes to be sent to the verifier. A unique code is generated which contains the encrypted credentials selected by the user. This code is sent to the verifier to be decrypted and verified.

C. Verifier Workflow

Verifiers can request specific attributes of identity information from individuals. Verifying Organization can enter the code sent by the user and the system will verify the authenticity of the user credentials. Verifiers can also view only those decrypted attributes of the identity credentials that were selected to be sent by the individual.

IX. Conclusion

We got to learn many concepts from this project like Blockchain, Zero Knowledge Proofs, Docker Containerization, NoSQL databases, etc. We also got hands on experience on various technologies and methodologies like Node, Angular, Python, Hyperledger. Blockchain based identity management system could be a feasible replacement for traditional identity management systems. The major drawback of using a decentralized database is its low transaction throughput.

A. Future Improvements

Different view can be created for a central credential schema issuing authority. Right now any of the issuing/verifying organizations can create schemas. Mobile Application can be created in which QR code will be generated by individual for easy scanning and verification by verifier. Create a user authentication system which fetches credentials from the blockchain, without relying on any centralized authentication service.

X. Deliverables

A. GitHub Repository

`https://github.com/SJSUFall2019-CMPE272/PrIde`