EduBlocks - A Decentralized Identity Management Platform

Akhilesh Anand
Computer Engineering Department, San Jose State University
San Jose, CA, USA
akhilesh.anand@sjsu.edu

Jasmine Dhunna
Computer Engineering Department, San Jose State University
San Jose, CA, USA
jasmine.dhunna@sjsu.edu

Sanjay Nag
Computer Engineering Department, San Jose State University
San Jose, CA, USA
sanjaynag.bangaloreravishankar@sjsu.edu

Vishwanath Manvi
Computer Engineering Department, San Jose State University
San Jose, CA, USA
Vishwanath.manvi@sjsu.edu

Abstract — International students applying to a foreign university for higher education go through a long and painful process of getting all the required credentials from various issuing agencies/intermediaries that the university expects

1. School to obtain transcripts,
2. Recommendation letters, etc.
3. ETS (for GRE/TOEFL tests),
4. WES credential (for vetting transcripts), etc.

This process takes months and students have to produce their personal identifiable information (PII) among other data. Also, these authorities additionally request students to provide proof of credentials to provide the service. The student's identity is replicated across the agencies centralized systems. This data is now not only vulnerable to potential information threats, but also potential misuse by the agencies themselves to further their own business interests. With EduBlocks, a decentralized blockchain platform, there's no need for any of the businesses/agencies to collect and store personal and credential data. Students will own their identity in the form of credentials and will only provide access to only the minimum required data that each of the agencies require to provide their service. Once approved, the credential is then added to the student’s wallet. Since credentials would be cryptographically verified through blockchain they can be trusted by all authorities. It also makes data more secured and transparent to all parties involved. The student can then present the verified credentials along with application form to the university thereby enabling them to make a faster decision.

Keywords—EduBlocks, Blockchain, hyperledger Indy, DID, decentralization

I. Introduction to Blockchain

Blockchain basically refers to a block of data that has been recorded over some time and is grouped and cryptographically linked to a previous set of data forming a chain of events. These computers agree on what happened over a time period and then each of them represents that data instead of having one centralized entity which is doing so. All of these events which took place on the blockchain, are recorded on a public ledger and all committed transactions are stored in a list of blocks. This chain grows as new blocks are added to it continuously. The blockchain gives the users the ability to authenticate digital information and generally has key characteristics of decentralization, persistency, anonymity and auditability. With these traits, this technology is safe, transparent and can help save the cost and improve the efficiency.

II. About Hyperledger Indy

Hyperledger Indy is a distributed ledger, purpose-built for decentralized identity. This framework provides tools, libraries, and reusable components for creating and using independent digital identities working on blockchains or other distributed ledgers so that they can be operated across administrative domains, applications. It is important that use cases for ledger-based identity carefully consider foundational components, including performance, scale, trust model, and privacy. Also, privacy by Design and privacy-preserving technologies are important for a public identity ledger where correlation can take place on a global scale.

For all these reasons, Hyperledger Indy has developed some specifications, terminology, and design patterns for decentralized identity along with an implementation of these concepts that can be leveraged and consumed.

To use Hyperledger, Indy SDK is the key. This is the official SDK for Hyperledger Indy, which provides a distributed-ledger-based foundation for self-sovereign identity. Indy provides a software ecosystem for private, secure, and powerful identity, and the Indy SDK enables clients for it. The major artifact of the SDK is a c-callable library; there are also convenience wrappers for various programming languages and Indy CLI tool.

A. Identity Concepts

Decentralized Identity: DID identities are independent from any centralized registry, identity provider, or certificate authority.

Self-sovereign identity: This identity is the concept that people and businesses can store their own identity data on their own devices, also provide it efficiently to those who need to validate it, without relying on a central repository of identity data.

Our persona Akhilesh Anand's has just passed his bachelor's and wishes to take admission to SJSU. His story can be described using following key concepts:

Role	Description	Examples
Holder	An entity that controls one or more verifiable credentials.	A person, organization or connected device can be a holder.
Issuer	An entity that creates a verifiable credential and associates itself with a holder.	They include corporations, government. In EduBlocks, Issuers are Amrita college and ETS
Inspector-Verifier	An entity that is related to processing the verified credentials.	In EduBlocks, SJSU is the inspector Verifier.
Identifier-Registry	Subject identifier's verification is mediated by this registry.	In EduBlocks, we use a distributed ledger which has a pool of indy nodes.

Table 1: Key concepts

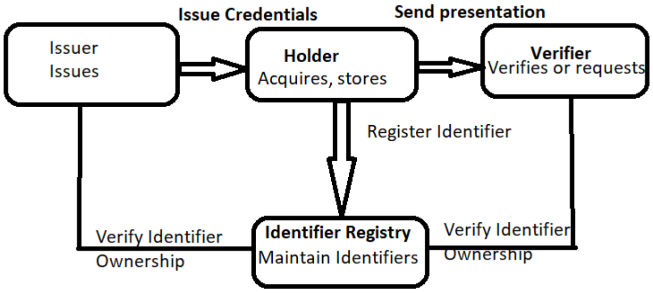

Fig 1: Key players

III. EDUBLOCKS PLATFORM

EduBlocks is a blockchain based education platform that governments, companies, and agencies can use for issuing and verifying credentials with your consent and more importantly students can use to view their wallet that has all the credentials issued. Requesting for credentials, issuing and requesting for proof request happens through the platform with the student's consent.

A. Architecture

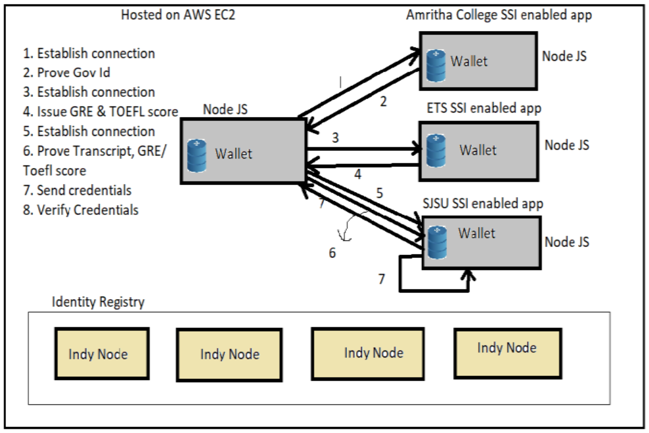

Fig 2: EduBlocks Architecture

EduBlocks has the following SSI enabled node JS apps:

-Akhilesh Anand (the student)
-Amrita College (His alma-mater)
-ETS (Educational testing services)
-SJSU (the university Akhilesh is applying to)

Each one of these Issuers creates a verifiable credential for an individual that is given by a credential schema definition. Before issuing a credential, a proof request should be filled out using the individual's current credentials and verified by the institution.

Additionally, the platform also runs Indy-node ledger (4 nodes), that uses practical byzantine fault tolerance algorithm for consensus.

B. Usage Instructions

Accessing the platform:

Akhilesh Anand (Student) - http://ec2-34-238-154-119.compute-1.amazonaws.com:3000/

Amritha College - http://ec2-34-238-154-119.compute-1.amazonaws.com:3002/

ETS - http://ec2-34-238-154-119.compute-1.amazonaws.com:3003/

SJSU - http://ec2-34-238-154-119.compute-1.amazonaws.com:3004/

Running locally:

Docker and docker compose are used to build and run the demo.

- Install docker and docker compose on the system.

- To check the version of docker and docker compose on the system, use the commands:

```
$ docker --version
$ docker-compose --version
```

To run EduBlocks, following steps were followed:

- Run below commands in nodejs folder
- use the commands,

```
   ./manage build  - Builds the images

   ./manage up - Runs the containers

   ./manage down - Shuts down the containers
```

Once it starts, you can access the below web apps:

Akhilesh Anand (Student)- http://localhost:3000/login

Amritha College - http://localhost:3002/login

ETS - http://localhost:3003/login

SJSU - http://localhost:3004/login

C. Technologies Used

- NodeJS
- EJS
- Indy-node (ledger)
- Indy-SDK
- Docker/Docker Compose
- Hosted on AWS EC2

D. Platform explanation:

For demo purposes, we login into all the agent systems, i.e. Akhilesh, Amrita College, ETS and SJSU which run on different instances and work on different ports.

- First step is to make a connection between the student i.e. Akhilesh here and his college (Amrita college). This connection is made using endpoint DID. Amrita's endpoint DID is used to make a new connection and this request is submitted.

- After this, a message pops up in the message tab of both the agents and if they accept the message a new connection is formed and Amrita has a proof request from Akhilesh with which it can validate Akhilesh's credentials.

- To issue a transcript, Amrita creates a schema for his transcript in issuing section and sends a credential offer.

- Once, Akhilesh accepts the transcript he can see it in his wallet.

- ETS issues GRE credential to Akhilesh

- ETS issues TOEFL credential to Akhilesh

- Akhilesh at the end has all the required credentials in his wallet, i.e. the transcripts, the GRE and TOEFL scores.

- In the end, a connection is established with the university that he is applying to, in this case, SJSU.

- SJSU submits a proof request to Akhilesh, which is accepted by both the agents. This proof request asks Akhilesh for his GRE, TOEFL and college transcript. On accepting this, Akhilesh's details are sent to SJSU, which can further be validated.

EduBlocks is a platform which makes the admission process hassle free and speeds up the same.

E. Benefits of EduBlocks

- Makes the entire application process more transparent and secure through strong cryptography enabled blockchain.

- Credentials are owned by students who can prove them to verifiers who can trust it.

- Decentralized Identifiers (DIDs) created and controlled by the owning entity

- The use of DIDs for each relationship, preventing cross-service correlation.

- Peer-to-peer, end-to-end encryption from message creator to receiver.

- Verifiable Credentials (VC) held by their owner and used only when necessary.

- Selective disclosure of VC data - exposing only the data necessary

- The use of VCs without the need to contact the issuer.

F. Challenges faced

- Understanding the Blockchain technology along with Hyperledger Indy concepts.

- Faced issues hosting indy on AWS EC2 that were resolved.

IV. References

1. Hyperledger Indy - https://github.com/hyperledger/indy-node#about-indy-node
2. IBM Indy world implementation - https://github.com/IBM-Blockchain-Identity/indy-ssivc-tutorial
3. Spencer Holman and Matthew Hailstone of Brigham Young University via EdX materials - https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS171x+3T2018/course/