Decentralized Identity Management Using Blockchain

Atharva Thorve
Dept. of Computer Engg. and IT
Veermata Jijabai Technological Institute
Mumbai, India
ajthorve_b18@it.vjti.ac.in

Mahesh Shirole
Dept. of Computer Engg. and IT
Veermata Jijabai Technological Institute
Mumbai, India
mrshirole@it.vjti.ac.in

Pratik Jain
Dept. of Computer Engg. and IT
Veermata Jijabai Technological Institute
Mumbai, India
phjain_b18@it.vjti.ac.in

Crehan Santhumayor
Dept. of Computer Engg. and IT
Veermata Jijabai Technological Institute
Mumbai, India
cdsanthumayor_b18@it.vjti.ac.in

Soham Sarode
Dept. of Computer Engg. and IT
Veermata Jijabai Technological Institute
Mumbai, India
svsarode_b18@it.vjti.ac.in

***Abstract*—Identity management is the process of setting and organizing the roles and access privileges of a user's identity. The current identity management system is centralized and is controlled by a single entity. Users' privacy concerns are not in their best interest. Users have very little to no control over their data. The centralized system becomes a single point of failure which is prone to attack that leads to users losing their data privacy if these centralized systems are breached. Therefore we propose a Blockchain-based decentralized Identity Management System that makes use of self-sovereign identity, decentralized identifiers, and verifiable credentials. It also gives users the ability to choose from a very large number of identity providers instead of just a select few corporations. The main advantages of the proposed solution include the elimination of the need for a central authority for identity verification and identity data management, the reduction of time spent on identity verification, the ability to share data with permission, and the ability to verify the origin of the data while sharing.**

I. Introduction

In today's internet age most websites provide single sign-on (SSO) solutions to end-users using Facebook and Google. Although this may be convenient for users it comes at a cost of their privacy. Moreover organizations like Facebook and Google who provide these services may not have their users best interest at heart. Thus user's data may end up being used for various different purposes without their knowledge and approval. Users can also make use of SSO solutions for the convenience of not needing to register at each and every website and keep their privacy intact at the same time. But there are still many more issues regarding the security of existing authentication system that are in place. There have been many real life situations involving tech giants such as Facebook and Google wherein a single breach in Data has left millions of users without any control over their shared data.

As a result, we offer a decentralized Identity Management System based on Blockchain that employs self-sovereign identity, decentralized identifiers, and verified credentials. The main advantages of the proposed solution include the elimination of the need for a central authority for identity verification and identity data management, the reduction of time spent on identity verification, the ability to share data with permission, and the ability to verify the origin of the data while sharing. Furthermore, biometric authentication will be required to access the verifiable credentials of the user. This is much safer than using regular email and password as the user might forget his or her password or use the same password for different applications thus making it much weaker [1]. The suggested system's goal is to provide an identity management system that is remote-friendly, scalable and by design provides both privacy and security.

A. Lacunae of previous research:

- **Personal Privacy Leakage:** To create a virtual online identity a user might be required to provide private or sensitive information such as medical information, employment information etc. This leads to exposure of a user's information which may be prone to security breaches as mentioned in [2].
- **Inconvenient Management:** A user may have to individually register themselves on various websites, which involves giving out commonly required information. Moreover the user provides information to websites which are independent of each other which adds a layer of difficulty for the users as they have to remember and manage various online identities [2].
- **Direct usage of DIDs:** DIDs are a very long string of characters. It becomes pretty cumbersome for humans to directly deal with DIDs, as they have to memorize and type in DIDs when any transaction has to be made [3]. Existing research papers don't provide a way to deal with this problem.
- **No easy way to onboard new/existing applications:** It is fairly cheaper for companies to make changes to

existing applications rather than creating an entire application around new technology. The solutions provided in existing research papers don't provide a way for existing applications to easily adopt new technologies for decentralized identity management. This proves to be a hurdle for many existing organizations that would like to make the transition to using a decentralized identity management system.

B. Unique Features of Proposed Solution:

- **Biometric:** Biometric Identification is used to secure the user's wallet and all the credentials inside it [1]. Without Biometric verification users cannot get new credentials or send the existing credentials.
- **DID to name mapping:** Since DID is a very long string of characters, identifying an entity, like a user or an organization, just by DID is not feasible. Having a mapping of DID to name eliminates this issue as users will only have to input the name of the organization with whom the transactions are being made. Therefore in our application a user or an organization can specify their name so as to make identification easier. All the mappings DID to name are stored on the Ethereum blockchain.
- **API endpoints:** We provide multiple API endpoints to make sure the onboarding process of new applications, organizations, and users is as smooth as possible. These APIs interact with the Ethereum blockchain and IPFS. These endpoints can be used by organizations to release new verifiable credentials and used by user, via the mobile wallet app, to get, give, or revoke access to these verifiable credentials.
- **Flexible access control:** The user can anytime revoke access to any identity they have submitted hence providing complete control to the user
- **Data portability:** We have added a feature so that the user can any time switch his mobile device and can recover the current state of the wallet and credentials without any issues.
- **Distributed data storage:** In the whole process of our application data is never stored in one central location. All of the users' data is added to IPFS (Interplanetary File System). All the metadata is stored on the Ethereum blockchain. A user's data can be accessed only through the given API endpoints after the user has given access to it.

The rest of the paper is organized as follows. Section II presents literature survey. Theoretical foundations are presented in Section III. Proposed system architecture and implementation is given in Section IV. Result and discussion is presented in Section V. Finally, Section VI concludes the paper.

II. Related Work

In this section, we present an overview of some existing research on Identity management and previous works concerned with enhancing the security and privacy of users using Identity management in Blockchain, followed by overview of existing work in DID and VC. [3] mentions how the distributed ledger technology has given a new perspective on identity management systems, and new approaches transpire, aiming to enhance decentralization, user control, and transparency. [4] mentions how we can use Blockchain Identity as a service. [5] gives an overview over six of the most promising identity management systems. [2] and [3] mentions about diifferent attacks on identity systems. [4] tells about VANET is a mobile network formed by vehicles, road side units, and other infrastructures that enable communication between the nodes to improve road safety and traffic control but also about how if messages exchanged in VANET wireless communication carry inferable Personally Identifiable Information (PII), it can introduce several privacy threats that could limit the adoption of VANET. [2] tells about Inconvenient management and Personal privacy leakage in centralized identity management systems.

The delicate balance between privacy and accessibility of electronic health records can be achieved with blockchain technology as mentioned in [6].Access control with permission delegation mechanism allows fine granular access to secure resources. Existing architectures for permission delegation and access control are either event-based or query-based.Consent manager which is a proof of a concept receipt is a cryptographic hash of the receipt, which is stored on the ledger. This process generates evidences that acts as a proof of sharing of verifiable credential and they can't be easily tampered with [7].

[8] mentions about a finger-print-based authentication framework.It also about other biometric authentication solutions available on the smartphone, e.g. retina scanning, voice, and face recognition. [9] describes multiple decentralized identifiers (DID) which are a part of SSI, each one created for every relation between identity owners. [7] mentions that Verifiable credentials are machine-readable, privacy respecting, cryptographically secure digital credentials of identity owners. Verifiable credentials support self-sovereign identity, such that identity owners accumulate credentials into an identity account and use the credentials to prove who they are.

III. Theoretical Foundations

For decentralized identity management using blockchain technology, various fundamental technologies needed are discussed below.

A. Blockchain

A blockchain is a decentralized database that is common among computer network nodes. A blockchain can store digital information and can act as a database. Blockchains are especially known for the integral function of keeping information tamper-free and decentralized and a record of transactions in cryptocurrency systems like Bitcoin [10]. The blockchain's innovation is that it generates trust without a trusted third party and ensures the accuracy and security

of a data record. The three pillars of the blockchain are Transparency, Decentralization and Immutability

B. Biometric Authentication

Authentication is the process of determining whether a person is who he or she claims to be. Passwords are sometimes too difficult to remember and store, instead we can leverage biometric authentication such as fingerprint which is unique to every user and can be used to authenticate him/her making it easy for users to access resources without having to remember password. [8] tells us about a finger-print-based authentication framework that includes a secure enclave (Trust Zone), sensors, and controller. [1] tells us about how we can use biometrics for mentally ill patients who cannot remember passwords.

C. Decentralized Identifiers

The Decentralized Identifiers (DIDs) defined in this specification are a new type of globally unique identifier. They are designed to enable individuals and organizations to generate their own identifiers using systems they trust [9]. Individuals, organizations, and things are given a standard, cryptographically verifiable, globally unique, and permanent identity via a decentralized identifier (DID). DIDs are totally controlled by the identity owner and are not reliant on central authority. DID makes use of public-key cryptography wherein each DID has a public and a private key, together forming an asymmetric key pair. The DID's private key is used to administer the DID's control. DIDs allow one identity owner to communicate with another identity owner through an encrypted private channel for the rest of their lives.

D. Verifiable Credentials

Credentials are digital documents that are used to confirm the identity of a particular individual, giving them the right to access documents pertaining to a particular field. These are documents that the individual makes use of on a day-to-day basis. They include driver's license, college transcripts, Aadhaar cards. Moreover, verifiable credentials are those credentials that are Machine-readable, privacy-preserving and cryptographically secure.

Verifiable credentials support Self-sovereign identity, in which identity owners can collect and store credentials in an identity account or wallet and make use of them to prove who they claim to be. Verifiable credentials are mostly verified by a third party organization, but they can also be self-attested. Attestation is achieved by making use of the concept of digital signatures. The individual first signs his/her records using their private key, which is then converted into a verifiable credential by an individual having a DID [9]. A verifier can then verify a person's credential using the issuer's/attester's public key. The verifier makes sure to trust only those credentials that are signed by a reliable issuer.

IV. Implementation

This proposed section covers the architectural overview of the application. It also goes through the workflow of the application to provide in-depth information about the workings of the application. It also covers different features provided by the application.

The architecture diagram shown in the figure shows various components and entities in the application.

- **Client App**: Client App is a digital wallet application which uses biometric functions to authenticate a user. All user's public keys and DIDs can be accessed using this client app.
- **User**: User entity consists of a DID and public keys. DID is the unique user ID. Public keys and policies are used to access the user Data. An API client is required to connect to the API. Client app is used to interact with all these processes.
- **Organization**: Organization entity is similar to user entity, except there is not a client app. Organization directly integrates with the API.
- **API**: The API connects the user, via Client App, and the Organization to the Ethereum Blockchain and IPFS. With help of API, onboarding into application ecosystem is made simple and easy.
- **Ethereum Blockchain**: The Ledger entity consists of all the transactions made by a particular user. The Ledger records information about events where users share their identity with others. It also stores the metadata of and the IPFS hexcode for credentials.
- **IPFS**: Interplanetary File System (IPFS), is used to store the actual Credential Data and Credential Schema Data. For every bit of data that is stored on IPFS an hexcode generated which is in turn stored as a part of mapping on the Ethereum Blockchain.

A. Workflow

In this section, the workflow for the sample use case is covered in detail. The use case covers a user signing up on our mobile application. The receiving a verifiable credential from its college organization. Finally, the users shares the received credential using the mobile app.

a) Wallet User:

Every user will be using the app and will be identified by their DID.The app have various sections like View Did document, get credential, sent credential and export data.

The user first has to create their DID, which is done by typing in their name as shown in the figure below. This data is sent to the main API service. Upon creating the API, a message containing the user is then displayed on the mobile app

1) Issuing Credentials:

1) **Scan QR Code and Gets Schema DID**

The first step to issuing credentials is for the user to

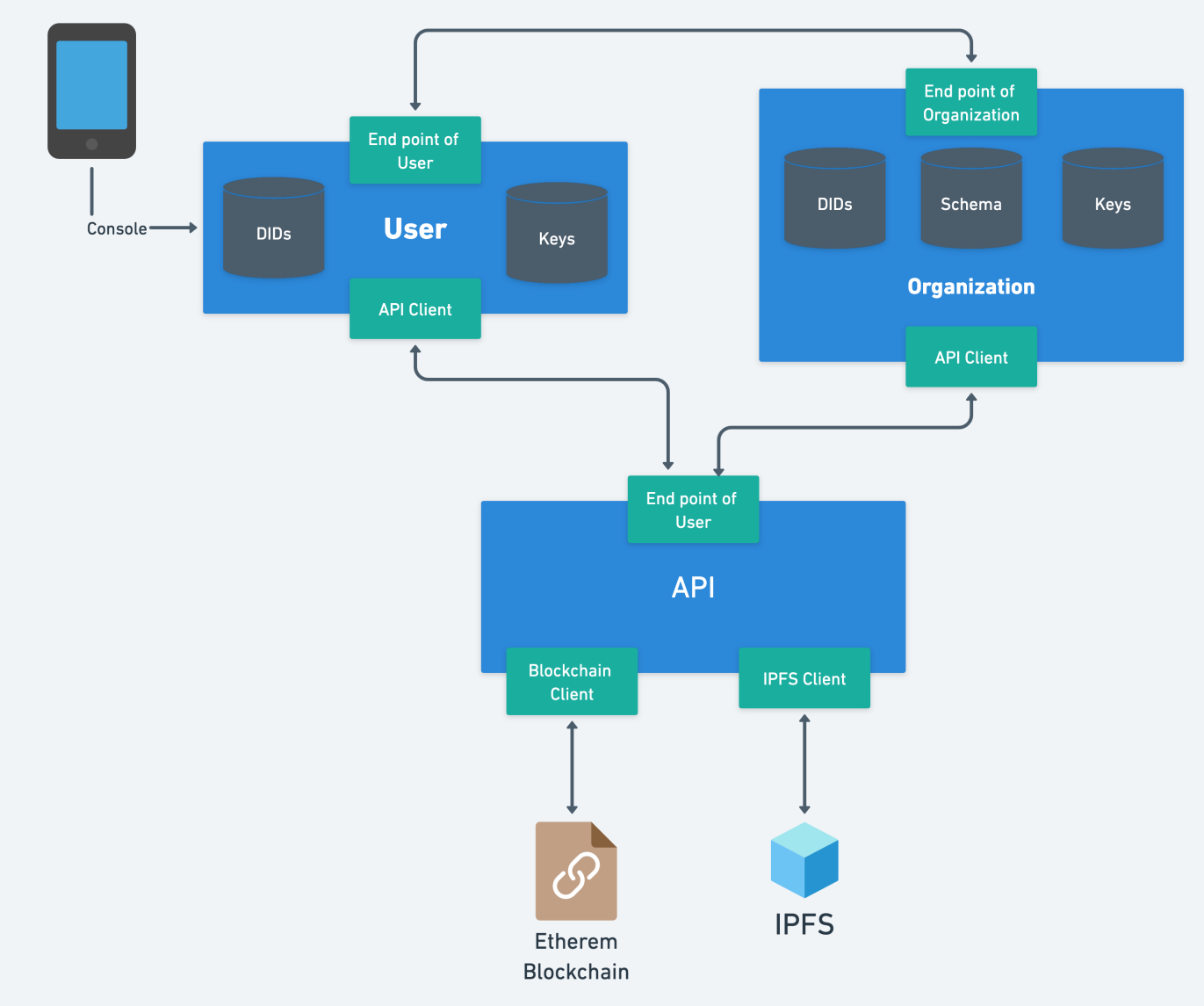

Fig. 1. Application Architecture

scan the QR code presented on the issuer website.This is done by *handleBarCodeScanned(type,data)* in the wallet application The QR code in the figure below encodes the following data: **URL**: It is the URL to which the mobile app sends the required data. **Schema DID**: This is the Schema using which the credential is to be issued. **User unique ID**: This is the unique ID of the User's account on the Issuer website. This helps to fetch the user data from Issuer's database

2) **Sends User's DID and Schema's DID**

   Once the User Scans the QR. The mobile app makes a request to the URL encoded in the QR code, containing the User's DID and Schema DID.

3) **Sending User's Credentials along with Proof**

   Upon receiving a request from the mobile app for creating the User's credentials, the issuer creates the Credential following the schema of the Specified Schema DID. The Issuer then sends the Credential along with its proof to the */addCredential* POST route of the Main API.

4) **Store Credentials on IPFS and getting Its Hexcode**

   The Main API firsts verifies the proof, confirming the identity of the issuer. Then sends the Credentials to IPFS to store it, which in turn sends the Hexcode corresponding to the Credential.The API internally calls the *createCredential(ownerDID, issuerDID, hash)* function of the smart contract to create the credential.

5) **Create DID of Credential**

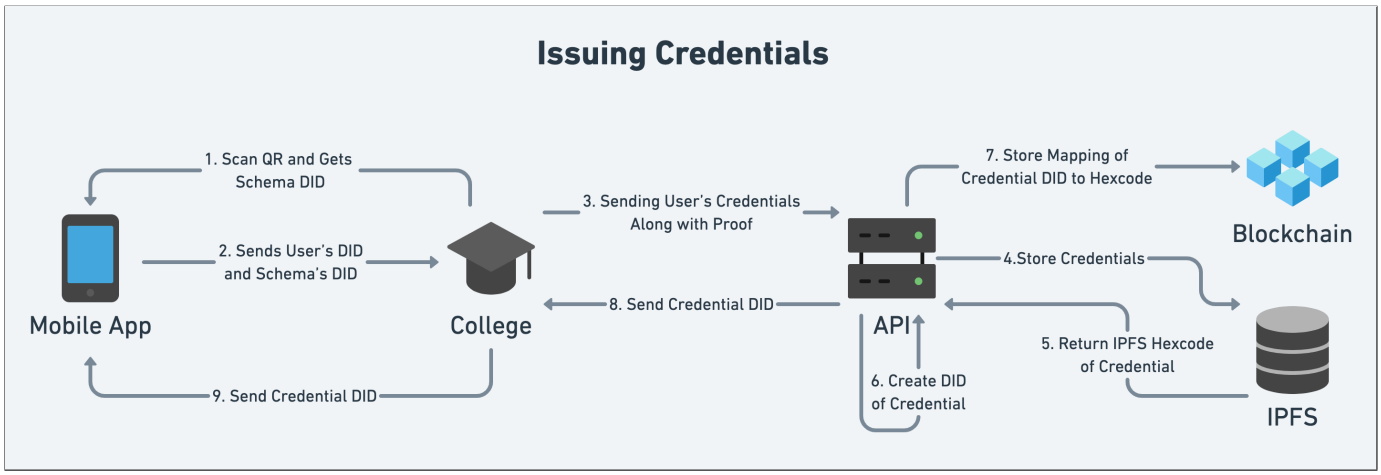

Fig. 2. Issuing Credentials Workflow

The Main API creates the DID for the Credential it has generated. This operation is performed within the *createCredential* function of the smart contract.

6) **Store Mapping of Credential DID to Hex Code**

The Mapping of Credential DID to its Hexcode is stored on the Blockchain. This is done so that the API is able to retrieve the Credentials from IPFS for a particular Credential DID for future situations. The user is also set as the owner of the credential, giving the user complete control over who has access to the credential. This Mapping is stored onto the *credentialStore* object in the smart contract.

7) **Send Credential DID from API to Mobile App**

Now that the credential is stored and its DID generated, the *createCredential* function returns the DID to the API which in turn sends it to the Issuer as a part of a JSON response along with a success message. The Issuer then passes on this information back to the user via the mobile app. Now, the user can use the credential DID to view the entire credential on the mobile app by just sending a request to the main API containing the credential's DID.

2) Sharing Credentials:

1) **Scan QR and Get Receiver's DID:**

The user scans the QR code shown on the website of the receiver, in this case GitHub. By scanning the QR code, the user gets the **Receiver's DID** and a **URL** on which the user will send the required data.

2) **Grants Permission to Access User's Credentials:**

Data to grant permission to access User's Credentials are sent on the URL received after scanning the QR code. The data that is sent includes hash, sign of the hash, User's DID, and Credential's DID. All this data is automatically sent to the Receiver organization after the **user selects** which document to share as shown in figure 3.

3) **Sending Received Information:**

The organization receiving the user credential just forwards all this information along with its own DID to the API service. The request is sent to */getCredential* POST request with credential DID, user's DID, receiver's DID, hash, and signed value of the hash in request body.

4) **Verify Signature:**

The API service first verifies the signature of the user to determine whether the user has given permission to access the credential. The verification of the signature is performed by using the function *verifySig* which internally uses the verify function provided by *@noble/secp256k1* npm package.

5) **Send Credential DID:**

After the signature has been successfully verified, the Credential DID is sent to the blockchain along with the Receiver DID, and User DID using the function *getCredential* exposed by the contract. After the owner of the credential is verified, as the given user, then this adds the Receiver's DID to a list of DIDs that have access to the given Credential DID.

6) **Receive Credential Hexcode:**

The blockchain stores the mapping from Credential DID to Credential Hexcode, which is generated when

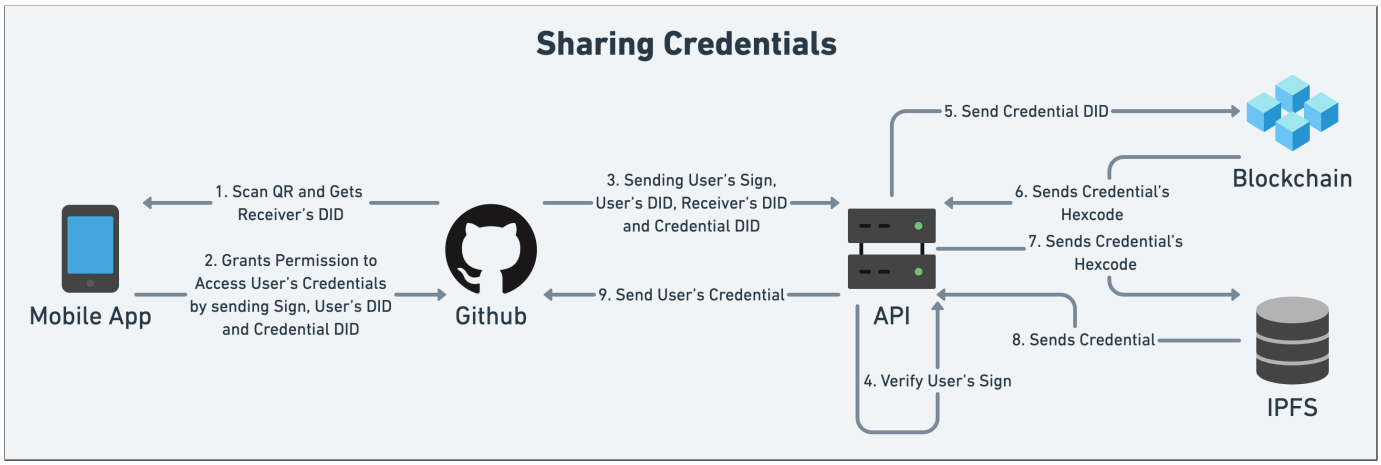

Fig. 3. Sharing Credentials Workflow

the credential is added to IPFS. This Hexcode is sent back to the API service as a return value for the function *getCredential* called in the previous step.

7) **Send Hexcode to IPFS and Receive Credential:**

After receiving the Hexcode from the blockchain, this Hexcode is used to fetch data from IPFS. IPFS has a function *cat* which returns a stream of data when provided with the hexcode. This function is used to fetch the data. IPFS returns the data corresponding to the Hexcode back to the service, which is then converted to a JSON object.

8) **Send Credential:**

The created JSON object is sent back to the receiver as a response to the API call *getCredential* made in step number three. The user is notified on successful sharing of the credential.

V. Result and Discussion

We developed an application prototype for decentralized identity management and verifiable credentials using blockchain. Important features such as issuing, sharing, and revoking access to credentials were achieved using IPFS and blockchain. We implemented a three tier architecture to provide greater accessibility and usability.

All entities, user or organization, can easily on board onto our application. For a user to onboard, the user only needs to install our application. For an organization, they'll need to integrate and use the APIs provided in our application. The organization has complete control over which APIs to use and integrate, but for all the features, every API needs to be assimilated.

The three tier architecture contains the mobile application that we developed, for facilitating easy onboarding of user to our ecosystem. The APIs used to interact with blockchain and IPFS. It also allows easy integration and onboarding of organizations on our application. The third and final tier refers to the Ethereum blockchain and IPFS, which is used to hold user and credential information.

Blockchain itself cannot be considered for storing complete credential data, as the gas fees for storing data on blockchain is very high and only a limited amount of data can be stored in a particular block on the Ethereum blockchain. Therefore, IPFS is used along with blockchain. On IPFS the actual credential data is stored, while on the blockchain only the metadata is stored. Even though blockchain improves transparency, there should exist trust between entities.

VI. Conclusion and Future Scope

A. Conclusion

Our work explored the concepts and implementation of decentralized identity and verifiable credentials using the Ethereum blockchain. Current Identity management systems are centralized and there are concerns with respect to users privacy and the question of security of users personal data. Blockchain provides a solution to this problem in the way of decentralized identity. The user has complete control over their credentials and identity. Since blockchain is decentralized and immutable, the control of the data of a particular user is only in their hand.

An organization can easily issue a verifiable credential for a user which is stored on IPFS, a decentralized file system. The credential can only be accessed by the user, and the entities the user gives permission to access it. The access permissions to a credential are stored on the blockchain and can only be updated by the user. This way decentralized identity with complete control over the credential data is achieved using blockchain.

B. Future Work

Future work for this involves integration of zero knowledge proofs. This way, only the necessary data stored in a credential needs to be shared. This will offer even more control over

the data to the user. Also, monetization of APIs need to be conducted. But this monetization will be only be applied to the APIs related to issuing and sharing credentials. Furthermore, fuzzy extraction from biometric data will also be implemented so as to make the more accessible to people with disabilities.

REFERENCES

[1] C. Adams, "A privacy-preserving blockchain with fine-grained access control," *Security and Privacy*, vol. 3, no. 2, p. e97, 2020.

[2] Z. Zhao and Y. Liu, "A blockchain based identity management system considering reputation," in *2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE)*. IEEE, 2019, pp. 32–36.

[3] J. Alsayed Kassem, S. Sayeed, H. Marco-Gisbert, Z. Pervez, and K. Dahal, "Dns-idm: A blockchain identity management system to secure personal data sharing in a network," *Applied Sciences*, vol. 9, no. 15, p. 2953, 2019.

[4] J.-H. Lee, "Bidaas: Blockchain based id as a service," *IEEE Access*, vol. 6, pp. 2274–2278, 2017.

[5] J. Roos, "Identity management on the blockchain," *Network*, vol. 105, 2018.

[6] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology," *Sustainable cities and society*, vol. 39, pp. 283–297, 2018.

[7] Z. András Lux, D. Thatmann, S. Zickau, and F. Beierle, "Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials," *arXiv e-prints*, pp. arXiv–2006, 2020.

[8] Z. Gao, L. Xu, G. Turner, B. Patel, N. Diallo, L. Chen, and W. Shi, "Blockchain-based identity management with mobile device," in *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, 2018, pp. 66–70.

[9] M. Aydar, S. Ayvaz, and S. C. Cetin, "Towards a blockchain based digital identity verification, record attestation and record sharing system," *arXiv preprint arXiv:1906.09791*, 2019.

[10] L. Conway, "Blockchain explained," https://www.investopedia.com/terms/b/blockchain.asp, Nov. 2021, accessed: 2021-12-03.

[11] R. C. Merkle, "Protocols for public key cryptosystems," in *1980 IEEE Symposium on Security and Privacy*. IEEE, 1980, pp. 122–122.

[12] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[13] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[14] Tykn, "Blockchain identity management: The definitive guide (2021 update)," https://tykn.tech/identity-management-blockchain/, May 2021, accessed: 2021-12-03.

[15] P. J. Windley, "Sovrin: An identity metasystem for self-sovereign identity," *Frontiers in Blockchain*, p. 30, 2021.

[16] S. Manu, L. Dave, S. Markus, R. Drummond, S. Orie, and A. Christopher, "Decentralized identifiers (dids) v1.0," https://www.w3.org/TR/did-core/acknowledgements, Aug. 2021, accessed: 2021-12-03.

[17] S. Manu, L. Dave, and C. David, "Verifiable credentials data model v1.1," https://www.w3.org/TR/vc-data-model/, Nov. 2021, accessed: 2021-12-03.

[18] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE, 2017, pp. 1–5.

[19] B. Faber, G. C. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, "Bpdims: A blockchain-based personal data and identity management system," 2019.