Self-Sovereign Identity in Semi-Permissioned Blockchain Networks Leveraging Ethereum and Hyperledger Fabric

1st Niranjan Sahi
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
niranjan.sahi@vanderbilt.edu

2nd Anda Liang
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
anda.liang@vanderbilt.edu

3rd Wyn Van Devanter
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
winslow.b.van.devanter@vanderbilt.edu

4th Konstantinos Oikonomou
Department of Computer Science
Vanderbilt University
Nashville, TN, USA
konstantinos.oikonomou@vanderbilt.edu

5th Peng Zhang, Ph.D.
Department of Computer Science and Data Science Institute
Vanderbilt University
Nashville, TN, USA
peng.zhang@vanderbilt.edu

Abstract—Patients often have their healthcare data stored in centralized systems, leading to challenges when reconciling or consolidating their data across providers due to centralized databases that store patient identities. The challenges disrupt the flow of patient care where time is sensitive for both patients and providers. Decentralized technologies have enabled a new identity model–Self-Sovereign Identity (SSI)–that grants individuals the right to freely control, access, and share their own data. This work proposes a system that achieves SSI in a semi-permissioned blockchain network using an open protocol as the certificate of authority and several guidelines for securely handling transactions in the network. Open protocols like Keccak can grant access to a permission-based network such as Hyperledger Fabric. The network architecture ensures data security and privacy through mechanisms of multi-signature transactions and guidelines for storing transactions locally, making this architecture ideal for privacy-centered use cases, such as healthcare data-sharing applications. The ultimate goal is to give patients full control over their identity and other data derived from their identity within a semi-permissioned network.

Index Terms—self-sovereign identity, semi-permissioned blockchain, healthcare interoperability, data sharing, privacy-preserving identity model, multi-signature transaction

I. Introduction

The cyber-identities of most patients today are created and managed by centralized systems owned by providers, who construct patients' identities using a set of attributes collected during the patient's registration [1]. Given the current provider-owned systems, as a patient visits different providers, the same patient information is constructed repeatedly. This centralization prevents patients from consolidating their data across systems, creating a bottleneck for the healthcare industry [2]. Decentralized technology has allowed for a new identity model: Self-Sovereign Identity (SSI), which aims to grant individuals full control over their own personal data and allow them to selectively reveal and share it with others [3].

In healthcare, traditional identity systems rely on centralized authorities to verify and issue identities, whereas SSI grants providers and patients full authority, enabling direct proof of identity claims to others [4]. Within a blockchain network, SSI offers a secure and decentralized approach for participants to prove their identities and qualifications [5], granting access to resources and transactions based on their identity or qualification document features [6].

A permissioned blockchain restricts access to network resources by issuing hierarchical credentials, allowing a network administrator to authorize participants, enable consensus, and control resource access [7]. It is commonly used in private or consortium settings, where participants are trusted and control over network and resource access is crucial [8].

In contrast, a permissionless, or public, blockchain is a type of blockchain that allows anyone to freely join the public network, participate in consensus, and view or query the data on the blockchain. This type of blockchain is typically used for applications that require a high degree of transparency and decentralization, such as cryptocurrencies [8].

In privacy-preserving applications where public blockchain storage is not feasible, a model is needed to provide open access while restricting transactions between trusted service providers and users [9]. Patient privacy is crucial in healthcare, and a permissioned blockchain can securely store and manage healthcare data, ensuring limited access to authorized parties [10]. This implementation aims to prevent data breaches, protect patient confidentiality, and facilitate secure sharing of sensitive data among authorized parties.

This work proposes a semi-permissioned model for the healthcare industry to store and transact patient information

securely. In doing so, our model makes use of

1) Open protocols for issuing and verifying credentials
2) A permissioned platform for entities to communicate and share information privately and securely
3) Multi-signature transactions for added security
4) Local storage of the ledger within nodes directly owned and managed by users and service providers
5) An Ethereum network to map a user's publicly available ID to data in the permissioned network

The following sections present relevant context and related work, components of our proposed identity model within a semi-permissioned blockchain, analysis of the framework focusing on security, privacy, and concluding remarks.

II. Background and Related Work

This section presents how a decentralized, permissioned network, using Hyperledger Fabric [11], can be implemented to give patients control over their digital identity and associated healthcare data, how a centralized authority (CA) can be modified to support it, and related work. Decentralized technology offers a solution by providing standards for achieving privacy and individual control over data, such as Hyperledger Indy [12] and Self-Sovereign Identity [13]. Open protocols such as Keccak can be utilized to implement user identities for a permissionless state while maintaining the need for and validity of permissioned components on the network.

A. Hyperledger Fabric and Certificate Authorities

Hyperledger Fabric is an open-source enterprise platform for building distributed ledger solutions. It supports flexible permissioned access to the network through a variety of modular consensus and membership algorithms and services, such as the Practical Byzantine Fault Tolerance (PBFT) algorithm [14] and the Membership Services Provider (MSP) mechanism. PBFT is a consensus algorithm that allows for efficient and secure agreement on the state of the ledger among validating nodes. At the same time, MSPs are a mechanism that allows the Fabric network to manage and validate the identity of its members. Together these algorithms allow Fabric to provide control over its network's activity.

Identities in Hyperledger Fabric are managed through a Certificate of Authority (CA), which issues digital certificates to network entities. These certificates establish and validate the identities of users, peers, and orderers, ensuring authorized access to the network [15]. Multiple signatures and hierarchical structures can be implemented in a network [16], [17], and MSP configuration allows for a subset of identities to attain scoped access to resources on the network. However, a CA's failure can disrupt network operations, hindering identity validation and certificate issuance. To mitigate this risk, a robust infrastructure and capacity for handling requests are essential. While having multiple CAs adds redundancy, each CA remains a potential point of failure. Therefore, self-healing and duplication capabilities provided by a container orchestrator are necessary for network reliability [18].

B. Open Protocols as Alternative to CAs

This work proposes using Keccak, an open protocol, as the foundation of public key infrastructure in a Hyperledger Fabric network. In permissioned networks, CAs are hidden, but using an open protocol like Keccak can provide transparency to a user's identity. Keccak256 is a cryptographic hash function accepting variable-length input and generating 256-bit fixed-length [19] output commonly used in blockchain networks, including Ethereum [20], to generate unique cryptographic keys for users and entities.

Using Keccak256 in the public key infrastructure enables users to generate cryptographic keys for network access [21]. Similar to permissionless networks, anyone can participate without central authority permission. However, Hyperledger Fabric for blockchain transactions requires permission to access and interact with the ledger. Participants gain authorized access by requesting permission from an MSP or meeting specific requirements like background checks or proof of identity [22]. Once granted, users can utilize their keys to interact with the ledger based on their permissions. In healthcare, patients can gain network access by verifying their identity with healthcare providers and associating it with the generated public key. Ultimately, the network would be a hybrid of permissioned and permissionless elements, making it a semi-permissioned blockchain network.

C. Decentralized Identity Management Frameworks

Hyperledger Indy and Serto are digital identity management systems that leverage blockchain technology to create and validate decentralized identities (DIDs) that can be cryptographically validated by authorities [12], [23]. They operate based on the principles of self-sovereign identity, giving users control over their data management. In traditional SSI systems, individuals receive a unique DID linked to a blockchain wallet containing verified identification information [13]. These systems eliminate the need for centralized databases or authorities to create, store, and share personal information, improving data privacy and security. In healthcare, self-sovereign identity enables patients to independently validate, control, and transfer their data, replacing the trust model with verifiable credentials. Previous work has proposed a multi-signature model where the public keys of all signers are aggregated to form a compact conjoined statement, enhancing data transmission efficiency, ensuring signer participation, and providing security against rogue key attacks through the Decisional Diffie-Hellman assumption [24].

Differentiating proposed work from related work:
Existing SSI frameworks often lack ownership features within a permissioned blockchain network. Our work presents a framework combining SSI benefits with a semi-permissioned network, detailing recovery, identity verification, data transfer, and user relationships across both permissionless and permissioned blockchains.

III. Proposed Architecture

We propose a new decentralized framework with a set of rules to dictate the flow of healthcare information. Our framework uses the Ethereum network and a Hyperledger Fabric network to securely build an interface for Hospitals and Patients to interact with each other. The Ethereum network is responsible for identity management and the Fabric network for communication between entities. Using the open protocol Keccak, the public identity will reside on the Ethereum network and have smart contracts to proxy validation and verification of identities for the Fabric network.

A. Ethereum Network

The Ethereum Network can be used to provide identities for entities through certificates, creating a relationship between data and the entity. This network will host several Smart Contracts to enable identity management. One Smart Contract will provide CA functionality by using an open protocol that accepts an input, applies the algorithm, and provides an output. In our framework, we leverage the Keccak256 open protocol. Now that our CA is represented by a Smart Contract, we will need to create three more Smart Contracts responsible for Creating, Reading, and Validating an identity. Creating will alter the state of the blockchain by appending a new transaction with a new DID for the user, while Reading and Validating will perform a lookup operation on the blockchain.

B. Hyperledger Fabric Network

The purpose of the Hyperledger Fabric Network is to facilitate communication between different verified entities, including healthcare organizations, patients, and delegates. There will be two overarching MSPs to validate identity and provide finer control for organizations.

1) Structure for Healthcare Organizations: One MSP (belonging to the Fabric Network) will be responsible for the validation of Healthcare Organizations (HO), which are established as independent Fabric organizations. These independent organizations will have their own MSP, which can communicate with the open protocol on the Ethereum chain, for finer control over healthcare data. For example, an organization may want sub-clinics to have different identity requirements, allowing patients to share this information with specific sub-clinics only. Moreover, every independent organization will also contain at least one Anchor Peer Node which allows for cross-organizational communication (enabling patient communication) via a Fabric Gossip Channel.

2) Structure for Patient and Delegate Organizations: A single MSP will be established for both the patient and the delegate organizations, as these organizations are comprised of individuals as users on the network. Within the organizations, each individual identity (i.e., a single patient) will have a corresponding Anchor Peer Node for establishing Gossip Channels with the providers and between each other (for shared access).

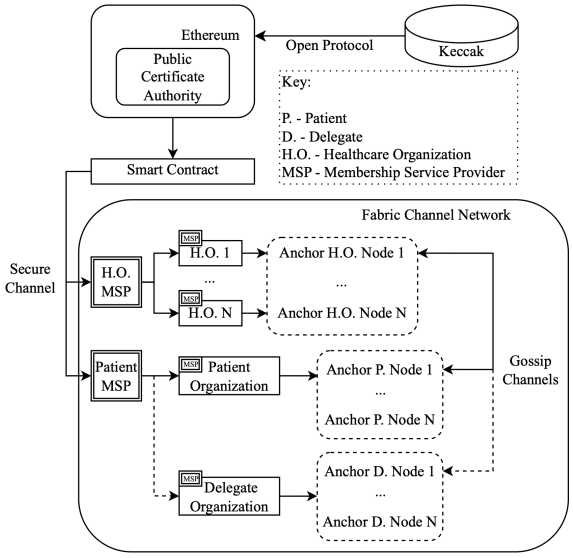

Fig. 1. In our model, the CA (on the Ethereum Network) is responsible for verifying identity and validating permission/access for both individuals and organizations. When a patient (user) attempts to access the network, they must be verified through the public CA first, as each user would have a unique digital certificate issued on first contact that is derived from the open protocol (Keccak) by the CA. On the other hand, organizations (providers) must also be verified and validated to access users' information. Providers and users are connected in the Fabric Channel Network through Gossip Channels, which can be established by and between Anchor Peer Nodes (i.e., H.O nodes, Patient nodes, and Delegate Nodes).

C. Multi-Signature Transactions

In a typical blockchain, each transaction is signed with a single private key to prove the authenticity and authorization of the transaction. When a transaction is signed with a private key, it is cryptographically secured and can only be validated by a user with access to the corresponding public key.

In addition to signing transactions with a user's private key, it is also possible to sign transactions with the private key of an organization. This is known as a multi-signature (multi-sig) transaction, and it can provide additional security and accountability for the transaction. By requiring the organization to sign the transaction in addition to the user, it can provide an additional level of assurance that the transaction is legitimate and authorized by both parties. This is particularly useful in applications that require a high level of security and privacy, such as those in the healthcare and finance sectors.

D. Storing Transaction Data

In Hyperledger Fabric, channels partition and isolate data within sub-ledgers for privacy. Each channel has its own ledger copy, visible only to member nodes. This allows private and secure data sharing between parties without exposing it to the entire network. For instance, a healthcare organization can use a channel to share medical records with a specific hospital while maintaining confidentiality from others. Data within a channel is stored on member nodes and replicated for consistency and reliability.

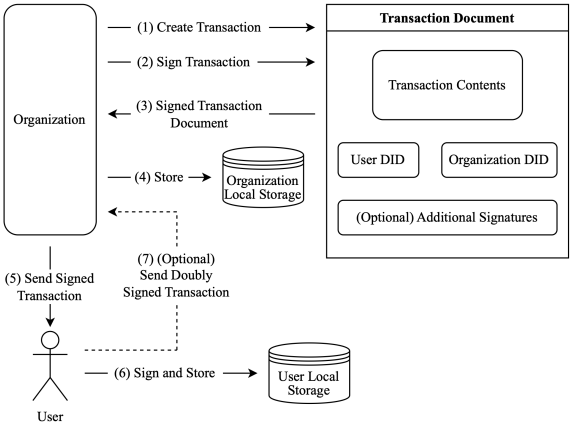

Fig. 2. The interaction between an organization and a user begins with the organization curating the information of a transaction into a document (1). The organization then signs this document (2) to validate the document's origin. The signed document is then returned to the organization where the organization will store this document, in their desired local storage choice, for recording purposes (3) and (4). The organization then sends this signed transaction to the user (5). The user will doubly sign this document validating that the information in the document pertains to them, and will store the document in their storage of choice (6). The user then has the optional choice to send this doubly signed transaction back to the organization where the organization can store the finalized record of the transaction (7).

The proposed architecture involves the utilization of two main Fabric channels. One channel is used per organization to ensure all nodes in the organization have access to the same ledger. This organization can represent any administrative entity that users transact with on the network. The second channel includes all organizations available on the network and contains a registry of the public keys associated with each organization. The information in this registry can be accessed by other organizations to verify user transactions that have been carried out externally. Then, after each transaction between a user and an organization, the transaction data will be stored locally on both the organization channel's ledger and the user device's file system. This local storage approach safeguards the data from unauthorized access or tampering, which is especially important when the transaction data contains sensitive or confidential information.

Furthermore, storing transactions locally on both the organization's and the user's file systems can facilitate easier data access and retrieval for later use. This local storage approach allows the user or anyone in the organization to access the data more quickly and easily when compared to remote storage. Overall, our proposed storage approach is especially useful for sensitive transactions that are accessed or processed regularly and for transactions that are part of a larger workflow..

E. Scalability

Fabric nodes can be run in pods managed by Kubernetes, a container orchestrator that can templatize operations of the network (e.g., creating a new organization) while keeping the critical configurations in sync. We propose that each organization manages its own Kubernetes cluster with Fabric nodes, integrating into a larger network of other Hyperledger Fabric clusters. A custom API will facilitate communication between organizations for easier node deployment and management.

1) Standard Platform: Fabric is a complex network to operate largely due to its multitude of components and multiple organizations maintaining a distributed ledger. Kubernetes can ease the operational burden by providing automation for configuring organization nodes. It also provides distributed system functionality to each organization's cluster, including mechanisms for auto-scaling, resiliency, and duplication [18].

2) Distributed System Capabilities: Kubernetes can be configured to autoscale the Fabric peer nodes if transaction metrics hit a certain threshold. The same can be done for storage based on utilization using Persistent Volumes [18] to store the ledger data. Kubernetes also provides resiliency by making the nodes self-healing, monitoring them for failures, and restarting them as necessary. It can replicate the peer nodes based on metrics such as transaction count and load-balance of incoming traffic, providing availability and fault tolerance. [25].

3) Decentralized configuration: Configuration for the multi-cluster network will be decentralized on a private ledger shared with participating organizations. This way, off-the-shelf tooling can be configured with agreement and co-ownership from the organizations in terms of how Kubernetes clusters run on the network. Policies in Fabric can be used to require a consensus among the organizations to change this underlying configuration. This prevents any single organization from owning the network alone and allows each organization to access and propose maintenance on the particular configuration.

IV. Discussion

In this section, the proposed architecture for achieving SSI in a semi-permissioned network will be analyzed for security, privacy, and various use cases. Detailed analysis of multi-signature transactions, an individual's recovery process, delegate association, and potential issues will also be outlined.

A. Multi-Signature Strategy

Signing a transaction with both the user's and the organization's certificate enhances accountability and provides a verifiable trail for auditing purposes. This approach improves security by mitigating identity theft, transaction forgery, and errors. In a single-signature system, imposters can post seemingly legitimate transactions if a user's key pair is compromised. However, compromising the accounts of all required signatures is necessary for such transactions in a multi-signature model. Multi-signature can also reduce the number of mistakes that typically occur as another party or reviewer will approve the transaction and sign off on it.

However, there are also potential disadvantages to signing transactions with both the user's and the organization's certificate. One potential disadvantage is that it can add complexity and overhead to the transaction process, as it requires multiple parties to coordinate and sign the transaction. This can slow down the transaction process for users to access the network and interact with the ledger.

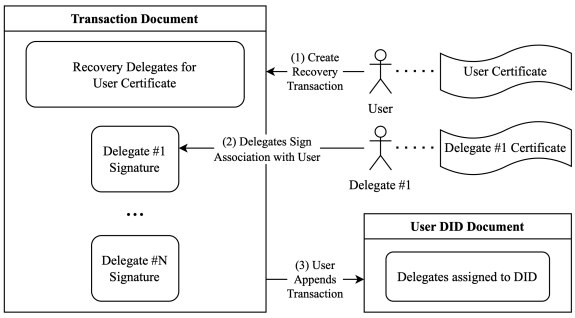

Fig. 3. The recovery setup begins with the User being registered on the network with an assigned certificate corresponding to their identity. The user will create a recovery transaction to be appended to their DID document, which lives on the Ethereum network (1). In the recovery transaction document, the user's content specifies delegates to their certificate or identity. The user will then ask the delegates to sign the document with their identity (2). Once all signatures are assigned to the transaction, the user will append this finalized version to their DID document (3). If the user loses their device, the delegates can create a new transaction with the user's new certificate that contains a majority of said assigned delegate signatures. Thus a trail of trust is created.

Additionally, signing transactions with both the user's and the organization's certificate can also create potential points of failure or vulnerability. For example, if the user's private key is lost or stolen, they may not be able to sign transactions on the network. Similarly, if the organization's private key is compromised, it could potentially disrupt the operation of the network and impact the ability of users to access the network. In mitigation for the loss of a user's key pair, a user can delegate a number of other users to help with identity recovery, in which case, a new key pair can be issued, and delegates that were originally assigned to the old key pair can create a majority signed attestation to their DID [26] to inform users on the network of their new public key.

B. Recovery Process

One of the key requirements of modern-day EMR software is the ability to reissue credentials if a patient loses certain identifying information, given the healthcare organization is the CA [27], [28]. This is fairly straightforward technically since healthcare providers run centrally, but it is a complex problem in decentralized systems with CAs [29].

While cryptographic certificates are highly secure, they're infeasible to reissue. This means that if a device is lost, it is typically impossible to recover the certificate. The case of disclosing a certificate to third parties, no matter how trustworthy, poses a security risk and potential for identity theft. Nonetheless, a recovery process is viable to associate patient information with a new device and certificate, update their network identity, and continue service access.

The process begins with an arbitrary Patient K, prior to losing their original device, creating Delegate Transaction(s) (i.e., designating delegates) and appending it to their DID store on the Ethereum Network, as detailed in Figure 3. Thus, when a device loss happens, the trusted users (delegates) can initiate a vote with a clear transactional trail, to update Patient K's identity with their new device DID and thereby new certificate. To begin the recovery process, Patient K will set up a new identity on both Ethereum and Fabric networks with a new device. Patient K will contact majority delegates and inform them to create a Recovery Linking Transaction to link Patient K's old DID to Patient K's new DID, effectively linking the old certificate to the new certificate. The Recovery Linking Transaction will be posted to the Ethereum Network. Following the posting of the transaction, the delegates will contact the healthcare organizations to verify the transaction if it contains majority delegate signatures and if Patient K's old DID exists in their system. Because Patient K registered themselves on both networks with their new device, the healthcare organization will create a Gossip Channel on the Fabric network with Patient K's new device's Anchor Peer Node and forward all healthcare information via the channel. Patient K will then aggregate and store all incoming data on their new device's local storage, see Figure 4.

Alternatively, another recovery process that could be implemented involves Patient K adding delegates to all gossip channels with healthcare organizations - making the delegate a spectator. The recovery would require Patient K to contact the delegates and have them gather the data in the gossip channels and provide it to Patient K's new device.

The first recovery mechanism is beneficial when a user has a lot of delegates and is arguably more secure as no single person can steal the user's healthcare information, whereas the second recovery mechanism hinges on having fewer delegates and exposes the possibility of individual delegates stealing the user's healthcare information. The first recovery mechanism is liable to both delegates and healthcare organizations to perform work on Patient K's behalf, while the second recovery mechanism is liable to only the delegates. The main drawback of the first recovery mechanism is the lack of efficiency, as gathering a majority delegate signature could be a difficult process. In rare cases, when some delegates can no longer provide signatures, there exists the possibility that a majority delegate signature is not attainable. On the other hand, the risk of the second recovery mechanism is the possibility of a delegate stealing a patient's healthcare information because the delegate is essentially granted access to the patient's healthcare information as a spectator of the gossip channel between the healthcare organization and the patient. However, it is arguable that no recovery process can be done without some degree of risk or inefficiency [30]. Overall, both of these recovery methods will enable Patient K to obtain all of their healthcare data on a new device and restore their SSI with a new identity.

C. Complex Transaction Trail

A key drawback to the multi-signature approach is that for transactions that reference other transactions or have several signatures, it becomes non-trivial and time-consuming to verify each signature or subsequent transaction. This is especially frustrating where the environment of seeking information needs to be swift and with little training in understanding the system at hand. An additional challenge comes with privacy: by following a transaction trail, signatures of private parties,

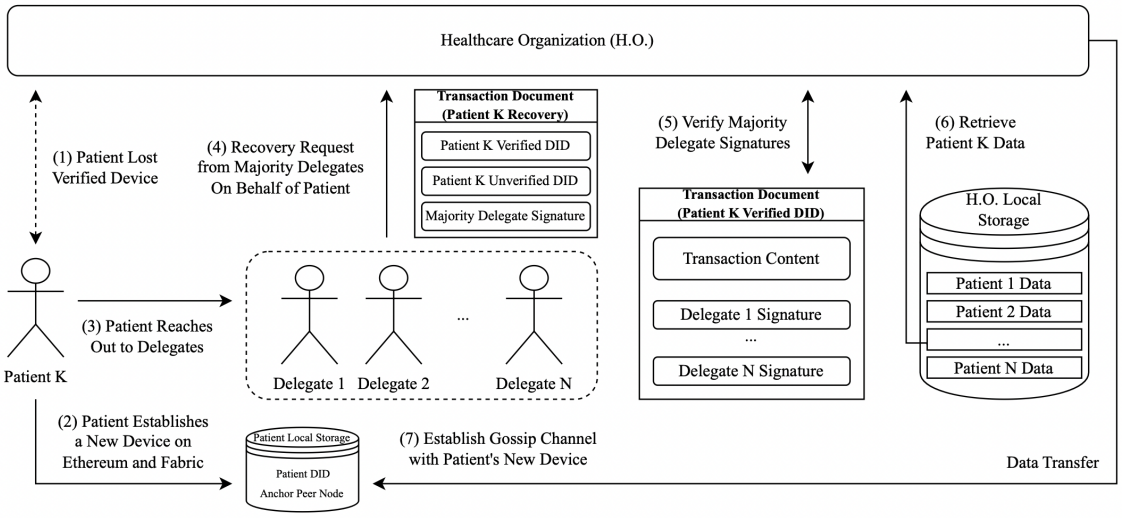

Fig. 4. The entire recovery process begins with a patient losing their verified (original) device (1). The patient would receive a new device and set it up on the Ethereum and Hyperledger Fabric Network. In turn, the user will receive a new DID and Anchor Peer Node (2). The patient would contact their delegates to create a new transaction to be posted to the Ethereum network and signed with a majority of delegates where the contents contain their old DID and their new DID (4). The Healthcare Organization(s) would verify the document linking the patient's old DID and new DID and fetch the data if the old DID existed (5) (6). The Healthcare Organization would establish a gossip channel with the patient's new device and send the related healthcare information (7).

who may wish to stay anonymous, can be identified. This poses a problem for certain individuals given the sensitivity in the healthcare industry. A potential mitigation would be to aggregate transactions over some predefined period into a separate node that indicates the user's transaction history for a given time frame. By componentizing a time period to delimit all transactions into a single one, the sheer volume of transactions a user would be sorting through would be reduced.

D. Challenges Associated with Local Storage

Saving transactions on local file systems can present storage and maintenance overhead drawbacks. Users and organizations must manage storage capacity, backups, and data protection, which can be time-consuming and costly, particularly for high transaction volumes. However, our proposed architecture addresses this by leveraging Kubernetes pods, enabling automatic scaling of storage capacity based on usage.

Moreover, storing transactions locally introduces potential points of failure and vulnerability. If local file systems are corrupted or lost, accessing transaction data becomes impossible, potentially disrupting network operations and user interaction with the ledger. Some strategies for mitigation would include redundant storage across nodes/devices or via distributed storage, implementing backup strategies as well as and recovery strategies as illustrated in Figures 3 and 4, and working monitoring and fault tolerance into the systems.

V. Conclusion

In conclusion, self-sovereign identity is a digital identity system that allows individuals or organizations to have control over their own personal data and share it selectively with others. This is in contrast to traditional identity systems, which rely on centralized authorities to verify and issue identities. The proposed SSI mechanism offers a secure, decentralized method for identity and qualification verification within a semi-permissioned blockchain network, enabling authorized transactions and resource access.

The proposed model replaces the Certificate Authority with open protocols for issuing or validating credentials, employs a multi-signature approach for identity verification, and stores ledgers locally to address privacy concerns. Specifically, this model uses Hyperledger Fabric, an open-source enterprise platform, for flexible, permissioned network access using various consensus and membership algorithms – with the caveat of using an open protocol CA. Unlike a fully permissioned network, our model proposes an open protocol with distributed control among individuals and the governing entity.

The semi-permissioned model proposed in this paper balances open access and trusted transactions, ideal for privacy-preserving applications like healthcare. Additionally, this model supports scalability through Kubernetes, a container-orchestrated deployment network ensuring privacy, security, and data ownership. Overall, the semi-permissioned model presented in this paper provides a secure, efficient solution for managing sensitive data on a blockchain, with access limited to authorized parties.

Acknowledgments

This research is supported by NSF's CISE-CRII program (Project No. 2153232). The authors would like to thank Dr. Kelly Aldrich from Vanderbilt University and COMBINED-Brain (combinedbrain.org) for providing domain expertise to this research.

REFERENCES

[1] A. Sonya and G. Kavitha, "An effective blockchain-based smart contract system for securing electronic medical data in smart healthcare application," Concurrency and Computation: Practice and Experience, vol. 34, no. 28, p. e7363, 2022.

[2] P. Zhang, J. White, D. C. Schmidt, G. Lenz, and S. T. Rosenbloom, "Fhirchain: applying blockchain to securely and scalably share clinical data," Computational and structural biotechnology journal, vol. 16, pp. 267–278, 2018.

[3] A. Giannopoulou and F. Wang, "Self-sovereign identity," Internet Policy Rev., vol. 10, 2021.

[4] M. Shuaib, S. Alam, M. S. Alam, and M. S. Nasir, "Self-sovereign identity for healthcare using blockchain," Materials Today: Proceedings, 2021.

[5] J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen, "Digital identities and verifiable credentials," Business & Information Systems Engineering, vol. 63, no. 5, pp. 603–613, 2021.

[6] Q. Stokkink and J. Pouwelse, "Deployment of a blockchain-based self-sovereign identity," in 2018 IEEE international conference on Internet of Things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData). IEEE, 2018, pp. 1336–1342.

[7] M. J. Amiri, D. Agrawal, and A. E. Abbadi, "Permissioned blockchains: Properties, techniques and applications," Proceedings of the 2021 International Conference on Management of Data, 2021.

[8] E. Strehle, "Public versus private blockchains," BRL working paper, Blockchain Research Lab, Tech. Rep., 2020.

[9] V. Malamas, G. Palaiologos, P. Kotzanikolaou, M. Burmester, and D. Glynos, "Janus: Hierarchical multi-blockchain-based access control (hmbac) for multi-authority and multi-domain environments," Applied Sciences, vol. 13, no. 1, p. 566, 2022.

[10] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz, "Metrics for assessing blockchain-based healthcare decentralized apps," in 2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom). IEEE, 2017, pp. 1–4.

[11] 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.

[12] A. Banerjee, B. Dutta, T. Mandal, R. Chakraborty, and R. Mondal, "Blockchain in iot and beyond: Case studies on interoperability and privacy," in Blockchain based Internet of Things. Springer, 2022, pp. 113–138.

[13] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," The Sovrin Foundation, vol. 29, no. 2016, p. 18, 2016.

[14] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric)," in 2017 IEEE 36th symposium on reliable distributed systems (SRDS). IEEE, 2017, pp. 253–255.

[15] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla et al., "Let's encrypt: an automated certificate authority to encrypt the entire web," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2473–2487.

[16] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," IEE Proceedings-Computers and Digital Techniques, vol. 141, no. 5, pp. 307–313, 1994.

[17] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," IEEE Access, vol. 7, pp. 103 059–103 079, 2019.

[18] T.-T. Nguyen, Y.-J. Yeom, T. Kim, D.-H. Park, and S. Kim, "Horizon-tal pod autoscaling in kubernetes for elastic container orchestration," Sensors, vol. 20, no. 16, p. 4621, 2020.

[19] M. Dworkin, "Sha-3 standard: Permutation-based hash and extendable-output functions," 2015-08-04 2015.

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

[21] C. Adams and S. Lloyd, Understanding public-key infrastructure: concepts, standards, and deployment considerations. Sams Publishing, 1999.

[22] T. Saleem, M. U. Janjua, M. Hassan, T. Ahmad, F. Tariq, K. Hafeez, M. A. Salal, and M. D. Bilal, "Proofchain: An x. 509-compatible blockchain-based pki framework with decentralized trust," Computer Networks, vol. 213, p. 109069, 2022.

[23] https://www.serto.id/, "Serto," 2021.

[24] D.-P. Le, G. Yang, and A. Ghorbani, "A new multisignature scheme with public key aggregation for blockchain," in 2019 17th International Conference on Privacy, Security and Trust (PST). IEEE, 2019, pp. 1–7.

[25] A. Poniszewska-Marańda and E. Czechowska, "Kubernetes cluster for automating software production environment," Sensors, vol. 21, no. 5, p. 1910, 2021.

[26] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, and M. Sporny, "Decentralized identity: Where did it come from and where is it going?" IEEE Communications Standards Magazine, vol. 3, no. 4, pp. 10–13, 2019.

[27] B. Middleton, W. E. Hammond, P. F. Brennan, and G. F. Cooper, "Accelerating us ehr adoption: How to get there from here. recommendations based on the 2004 acmi retreat," Journal of the American Medical Informatics Association, vol. 12, no. 1, pp. 13–19, 2005.

[28] A. A. Elngar, K. Sagayam, A. Elngar, and A. Elngar, "Augmenting security for electronic patient health record (ephr) monitoring system using cryptographic key management schemes," Fusion: Practice and Applications, vol. 5, no. 2, pp. 42–52, 2021.

[29] M. T, K. Makkithaya, and N. V G, "A blockchain based decentralized identifiers for entity authentication in electronic health records," Cogent Engineering, vol. 9, no. 1, p. 9, 2022.

[30] P. Mayer, "Data recovery: Choosing the right technologies," Datalink White Paper, 2003.