Case studies for the necessity of blockchain in Self-Sovereign Identity

Youngwoo Choi, University of Twente, The Netherlands

Identity management technology has gradually developed and has now evolved into a self-sovereign identity in that the user has all control over his/her authentication information. Blockchain is one of the main technologies to realize SSI, and various SSI services using blockchain have been studied and created. However, it is still ambiguous whether using blockchain for SSI is the best option. Therefore, it is necessary to evaluate whether blockchain is required in SSI and figure out whether there is a better alternative. In this paper, we will analyze and compare two representative blockchain-based SSI systems, uPort[7] and Sovrin[8], investigate the need for blockchain in those services, and suggest alternatives if necessary.

Additional Key Words and Phrases: blockchain, SSI, Self-Sovereign Identity, uPort, Sovrin

1 INTRODUCTION

As the times and technologies evolve, various services are running online, and identity authentication has been becoming necessary accordingly. In the past, single or multiple authorities have been in charge of issuing and authenticating, but there have been attempts to gradually move control from authorities to users. As gradually developing, the stage of 'self-sovereign identity' that people want to achieve is progressing. Self-Sovereign identity (SSI) is an identity system that ensures 10 principles provided by Allen[1] and one of the technologies that made SSI possible is blockchain.

As the features of blockchain, such as decentralization and immutability, allow everyone to share the tasks that a single authority has taken on, identity systems using blockchain have been created, such as uPort [7] and Sovrin [9].

Obviously, it is true that SSI can be created using blockchain, but there are ways to create SSI without blockchain [5]. To determine which implementation would be better, it should be inspected first if the blockchain is the necessary technology for the service.

In this research, we investigate crucial properties of blockchain which enables SSI, SSI requirements and conditions, compare the underlying design of uPort and Sovrin as case studies.

This paper initially scrutinizes what is the pre-requisite of SSI services and which properties of blockchain fulfill the requirements.

TScIT 37, July 8, 2022, Enschede, The Netherlands

After that, to evaluate the necessity of the blockchain, Wüst and Gervais's methodology [10] will be applied. Wüst and Gervais [10] found that blockchain is often used even it is not really needed, so that they introduced a framework to evaluate the necessity of blockchain for a service. This methodology can provide the criteria of the evaluation for the blockchain in SSI. The framework helps to analyze specific conditions for the service and draw which system suits the most for the service, among permissioned blockchain, permissionless blockchain and neither.

Applying the framework to uPort and Sovrin, it will be able to determine what they need respectively, among a permissioned blockchain, permissionless blockchain or neither. Furthermore, an alternative solution will be discussed if the blockchain is not the best option for the services or they are not using the recommended blockchain, which is permissioned or permissionless.

This research will be achieved solving the following research questions.

RQ1 : What are the features of using blockchain for an application? Which properties meet or support the requirements of SSI?
RQ2 : What is the difference between the ways of implementation with blockchain in uPort and Sovrin?
RQ3 : What is the result of the evaluation of the necessity of the blockchain for uPort and Sovrin? What could be an alternative design for such system without BC based on our evaluation?

The methodologies to answer the research questions are descried below. Answering RQ1 : By referring to the blockchain and SSI documentations, the properties of blockchain and requirements of SSI will be compared and analyzed which properties are in the intersection. Answering RQ2 : By referring to the technical documentations of uPort and Sovrin, the usage of design and usage of blockchain in them will be explained and compared. Answering RQ3 : Wuster and Gervais's methodology to evaluate the necessity of blockchain in SSI will be applied and an alternative solution will be proposed if needed.

The results will show that blockchain can supplement SSI features and uPort and Sovrin use different type of blockchains. In addition, an alternative design with central database for Sovrin will be proposed.

The rest of the paper is arranged as follows. Section 2 shows the related works and section 3 describes the features of SSI and blockchain, the designs of uPort and Sovrin and the framework which will be used for the evaluation. Section 4 shows the results of the evaluation of the uPort and Sovrin and provides a discussion based on the result from the evaluation. Finally, section 5 presents conclusion and summarization.

2 RELATED WORK

In this section, related works in SSI and blockchain will be shown.

The origin of the term SSI starts from a blog post written by Devon Loffreto[6]. In the post, he argued that the sovereignty of an individual's identity was being violated by administration systems.

Researches on SSI has been carried out, and Allen provided a definition and 10 principles for SSI[1] in 2016, when there was no precise consensus on SSI's definition.

Referring to the Allen's proposal, Ferdous and his colleagues [3] released a formal definition for SSI in 2019. Geovane and his colleagues [2] proposed in their paper in 2020 about different approaches of SSI implementation in IoT and their comparisons.

Samia and colleagues [4] analyze Sovrin, uPort and ShoCard services and describe their differences. Although they have compared the advantages and disadvantages of each service, but an alternative solution is not proposed.

3 BACKGROUNDS

3.1 SSI

SSI is a system in which users can manage/supervise their identity information, and no one else can verify their identity without the user's consent. For example, if you sign up for a general web service, all registered information is stored in the database of the web service. Therefore, the user does not have direct access to this information but can access and modify it only through the corresponding web service. On the other hand, SSI can be viewed as a system that allows users to verify their identity by storing identity in their own storage other than the entity's database and showing it only when desired, rather than the entity owns the identity of the user. Through some development, this identity system appeared, and exactly what principles SSI has will be presented in the following section.

3.1.1 Evolution of Digital Identity

There are 4 phases of evolution of digital identity[1,5].

1) central identity

Centralized identity is the oldest identity management model. A single entity issues and manages identity. This makes it easy for a single authority to manage users, but the problem is that users must be issued an ID every time they sign up for a new service or site. Moreover, there is a risk that the authority may deny its identity or even confirm the wrong identity. Also, users do not own their own identity records, and they can be passed on to anyone at any time. In other words, a single authority, not a user, has authority on all identities.

2) Federal identity.

Federation is common in large businesses and makes multiple internal services available through a single sign-on method. For example, you can access multiple related multiple sites through your Microsoft account. This reduces the burden that users have to remember and manage various identities, but this federation can also be viewed as a huge centralized authority like Microsoft, and has the disadvantage that if one account is lost, other sites cannot be used together. In addition, in the end, it can be seen that the sovereignty over identity still does not belong to the user.

3) User centric

It appeared to solve the problems of central identity and federal identity and tried to place the user in the middle of the identity process as the term indicates 'user-centric.' For example, by connecting to various SNS and websites through a facebook account, you can register and log in without providing additional information to the services. This is only possible if the user has allowed it.

However, relying parties have to connect with many providers (like facebook or twitter) to gain a wide range of customers, which results in complex and time-consuming integration. Also, in the end, the user's credential and identity data are ultimately owned by the identity provider, so it can be seen as the same as the central authority again.

4) Self-Sovereign

SSI is an identity management model that is in the final stage of the evolution of this identity, and each individual fully owns his or her identity and is not stored in any other silo. And no third party can claim that it provides an individual's identity, and an individual is ultimately independent from any single organization.

3.1.2 ***Principles of self-sovereign identity***

In order to satisfy these conditions of SSI, the ten principles of SSI were defined by Allen. Among those ten principles, 4 of them are introduced which are related with the blockchain's properties. Other 6 principles are more related to an application's design, rather than blockchain.

1) control

Users must control their identities, which means that they are able to refer, update and even hide their identities. It also indicates that a user is the authority itself of their identity. Rather users have to manage and control all of the claims about their identity, others make claims about a user and they can accept or deny it by themselves, without intervention of 3rd parties.

2) Access

Users must have access to their own data. It should be visible to the owner without any hidden information, and there should be no gatekeeper. Of course, you should only be able to access your own data and not be able to access the data of other users.

3) Transparency

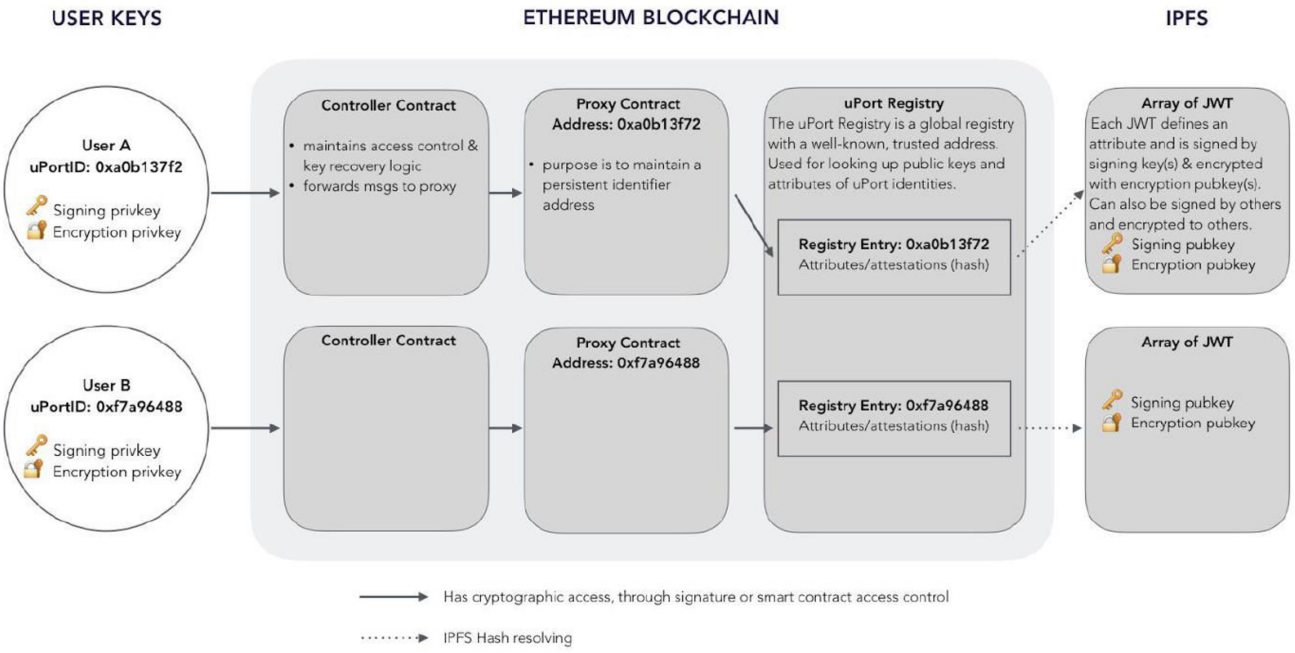

Figure 1. components of uPort System [7]

systems and algorithms must be transparent. That is, it must be clear how the system is managed, and identity works, and how it is updated must also be disclosed. The algorithm is open source, so anyone can check it, it is well-known, and it should not belong to a specific architecture as much as possible.

4) Persistence

Identites must be long-lived. Ideally, identites should last forever as long as the user wants them to. The private key to guarantee the user can be changed, but even so, the user's information itself must be maintained.

In the following section, it is checked if those features can be satisfied with blockchain's feature.

3.2 Blockchain

Blockchain technology is a database mechanism that enables the transparent sharing of information within business networks[10]. A blockchain database stores data in blocks chained together by a cryptographic hash. A block can store a list of transactions in which peers can not only send and receive digital assets directly between peers without any other intermediate steps, but also execute code called smart contracts. Once a block is appended, modification or deletion is impossible due to hashing, and because of the nature of the distributed ledger, all users can verify the blockchain, so it can be viewed as a transparent system.

In the blockchain network, users do transactions in a peer-to-peer(P2P) network, where no central server exists, and peers communicate with a direct connection. To authenticate and verify a peer, a public / private key system is used [8]. A *private key* is a randomly generated 256-length secret number and a *public key* is a number computed using the private key which is shared with other peers. With these keys, encryption and decryption are available. For example, if something is encrypted with a private key, then it can be decrypted with a public key and vice versa. When two peers do transactions, a sender can send a transaction encrypted with the receiver's public key so that only the receiver can decrypt and read. A *signature* is a number generated using a hash function with the private key so that only the owner of the private key can generate it and others can verify with the user's public key, which can be used to verify the writer. Attaching a signature inside the transaction, a receiver can verify the writer.

Since this blockchain is a kind of database, writers and readers exist. A *writer* collects these transactions and processes them into blocks and appends them to the blockchain, and a *reader* is a participant who is either in transaction creation process, simply reading and analyzing or auditing the blockchain. The blockchain can be classified according to users' accessability as readers and writers.

1) permissionless Blockchain

Permissionless blockchain does not have any restrictions for the users who are qualified to append the blocks of transactions. Any peer can join and leave the network as reader and writer at any time. Accordingly, there is not any central organization which manage members in the system. Bitcoin and

Ethereum are the representatives of the permissionless blockchain.

2) permissioned Blockchain

There are the predefined members who are qualified process the transactions and append blocks on blockchain. Therefore, there is a central entity which manage membership and can ban malicious users.

3.2.1 Analyzation with SSI principles

In the blockchain, since there is no central database, the entity cannot store the user's credentials in their own storage are only stored in the user's wallet. Therefore, no one else can access, and only the users can access their identity credential. Therefore, *access* is protected. Also, due to the nature of the blockchain, communication proceeds peer-to-peer, so when claiming data about a user, only the user should be requested. As a result, the *control* is maintained. Due to the nature of the blockchain where all data is public and the code is maintained as an open source, the *transparency* of SSI is maintained, and it is considered *persistent* due to the nature of the blockchain that cannot be erased once it is uploaded to the block.
In conclusion, the principles of SSI shown in the previous section are enabled with blockchain.

3.3 Current SSI with blockchain

3.3.1 uPort

UPort is an SSI system based on the public permissionless blockchain called Etherium[7]. Basically, it operates using smart contracts, which are Controller Contract, Proxy Contract and uPort Registry as described in Figure 1[7], which are explained in the following section.

Using uPort, users can safely manage and publish their identities. Organizations and other resources as well as individuals are also able to join and create their identity through uPort. All credentials are managed by the user himself without any third party's intervention, they own identity-related keys in their personal wallet.

The smart contracts have different functionalities, respectively.

1) Controller Contract

The Controller Contract has features that access the Proxy Contract, and helps the user authenticate himself to the Proxy Contract. In addition, there is a function that allows the user to recover control of the proxy contract if the private key is lost or needs to be replaced.

2) Proxy Contract

A proxy contract contains a user's permanent identifier. The private key is stored in the user's mobile device, and by putting a permanent identifier rather than this private key as the core of the identity, you can change the private key while maintaining your identity persistently. Accordingly, when a user changes or loses a mobile device, it allows the user to replace the private key while maintaining their uPort identity. The permanent identifier is the address of the Ethereum smart contract, which carries transactions on the blockchain.

3) Registry Contract

Registry Contract cryptographically connects uPort identifier and off-chain data such as InterPlanetary Files System (IPFS) in the Figure 1. IPFS is a decentralized object storage system where storing and retrieval of data is possible. The linked data are stored in the form of Json Web Token (JWT) and can be encrypted using the user's encryption key. Furthermore, it can be encrypted with the public encryption key of other identities to be shared so that only them can see it.

Through the above three contracts, users can own and manage not only their personal identity and data, but also cryptocurrencies and digital assets connected to the blockchain, access various digital services without passwords. They also can sign to identity claims and blockchain transaction.

3.3.2 Sovrin

Sovrin is a public permissioned blockchain based identity [9]. As it is based on permissioned blockchain, there are entities that manage membership. The qualified members who can write on ledgers are called stewards, trustworthy individuals or institutions elected by the Sovrin foundation. In addition, trusted anchors elected through stewards as trusted members are able to provide credential to 'members', who are general users, and help with verification. In other words, Stewards are selected to write transactions and append blocks to the blockchain used by Sovrin, whereas members can only read it.

Trusted anchors are trusted individuals or institutions such as banks, schools and corporations. They can offer credential to members of their institutions, and also claim credential of users who need it. Each trusted anchors requires different identity attributes, and registers and uses a schema for them in the blockchain. Therefore, when a user creates a new credential, he writes attributes that match the schema and credential definition defined by the trusted anchor, and the created credential containing the trusted anchor's signature is stored in the user's personal wallet. In this process, only the user's decentralized identifier and public key are stored in the blockchain, and the credentials and transaction records of the user are stored only in the user's wallet. Therefore, other users cannot find any personal information of the user with only the DID in the blockchain. DID is a decentralized identifiers which is used in a decentralized system, referring any subject such as a person, organization, thing, data model or abstract entity as determined by the owner of the DID.

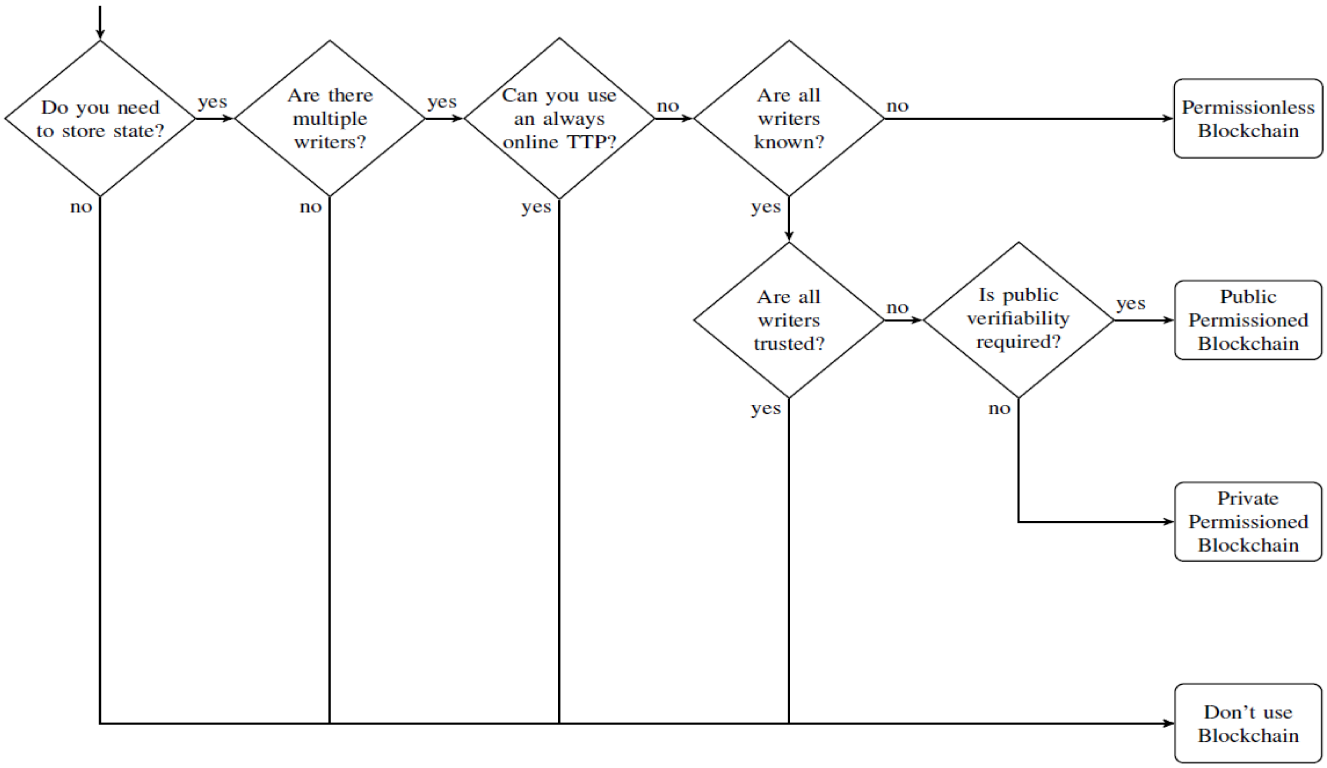

Figure 2. Flow chart offered by Wüster and Gervais to evaluate the necessity of blockchain for a specific system [10]

3.3.3 Comparison

The biggest difference is that the types of blockchain they are based on. In the case of uPort, based on public permissionless blockchain, anyone can easily join, and anyone can leave transactions and blocks. That is, there are no restrictions on becoming a writer, which means that it is not known exactly which writers there are. On the other hand, in the case of Sovrin, based on public permissioned blockchain, only authorized writers can leave a transaction. Also, since only stewards have a role as writers, all writers are known and trusted. In the case of uPort, it can be said that it is focused on easy to use in that anyone can easily join, whereas Sovrin is safer in terms of security in that only selected writers can create blocks.

Let's look at the data stored in the blockchain. As mentioned earlier, in the case of uPort, each user's controller contract, proxy contract, and registry contract will be stored. Proxy contracts contain a permanent identity, so each contract will contain its own address. User identity is stored in IPFS or off-chain storage, which is a storage outside of blockchain, whose address is stored in the registry contract. On the other hand, Sovrin stores schema and credential definitions written by trust anchor, public DID and public key for verifying issuer, and related documents.

3.4 Wüster and Gervais's framework

Through the methodology mentioned above, it is possible to investigate the need for blockchain in a specific system and further determine which type of blockchain is suitable.

Referring Table 1, it is noticeable that in the order of permissionless, permissioned, and central database, they have better throughput and latency. In other words, it is necessary to check the required database according to the system and determine the most suitable type to prevent unnecessary performance decrease.

Depending on the conditions and goals of the system, one can decide whether to use the central database or blockchain, if they use a blockchain, whether to use a permissioned blockchain or a permissionless blockchain. Figure 2 is a flow chart from Wüster and Gervais's research [10], which consists of 6 questions to help make a decision. The questions will be briefly explained in the following sections.

1. Do you need to store state?

If there is no need to store information, the blockchain, which can be viewed as a distributed database, will also be unnecessary because there is no need for a database at all. Therefore, it is necessary to check whether and what information the system needs to store. If a system needs to store state, then we can move on to the next question.

table 1. Evaluation of the databases for some criteria [10]

	Permissionless Blockchain	Permissioned Blockchain	Central Database
Throughput	Low	High	Very High
Latency	Slow	Medium	Fast
Number of readers	High	High	High
Number of writers	High	Low	High
Number of untrusted writers	High	Low	0
Centrally managed	No	Yes	Yes

2. Are there multiple writers?
One of the benefits of using a blockchain is integrity. That is, it is impossible to make unauthorized modifications to the data once uploaded to the chain, but if there is only one writer, an advantage obtained when using the blockchain is unnecessary. So, if there is only one writer in the system, it would be wiser to use a central database.

3. Can you use an always online TTP?
If a trusted third party (TTP) exists, there are two options. First, if the TTP is always online, it can be exercised as a delegator for write operations and as a verifier for state transitions. In this case, no other writers except the TTP exist, which is similar situation of a sole writer, so that there is no need for a blockchain. Second, if the TTP is not always online, the TTP can assume the role of a certificate authority within the permissioned blockchain.

4. Are all writers known?
If writers are not known, the permissioned blockchain cannot be used, because permissioned blockchain is only available to authorized writers. Therefore, in this case, permissionless blockchain should be used.

5. Are all writers trusted?
If all writers are trusted, the central database can be used. The best solution would be to use a shared central database where not malicious and allowed writers can access only. However, if not all writers are trusted, then move on to the next question.

6. Is public verifiability required?
In this question, public verifiability means whether all participants can verify the correctness of the state of the system. That is, if only some limited users are able to check and verify the state, the private permissioned blockchain should be used, and if all observers should have verifiability, the public permissioned blockchain should be used.

4 RESULTS

This section includes the evaluation of the SSIs investigated in section 3 and further discussion from the result of the evaluation.

4.1 Evaluation

The characteristics of Sovrin and uPort that we checked earlier will be applied to the flowchart. One thing to note is that, since Sovrin and uPort already use blockchain, the characteristics of blockchains are excluded in the evaluation and only their design and goal are considered.

4.1.1 uPort

1. Do you need to store state?
In uPort, controller contract, proxy contract and application contract are needed to be stored in a database. Even if uPort is not based on blockchain and the states were not smart contract in a blockchain, similar data or states which contain the same information and functionality should be stored in a database. Therefore, it is ‘yes’.

2. Are there multiple writers?
In uPort, writers are users who are identity holders. That is, the user with the identity is both a writer and a reader. Since all users using uPort are writers, multiple writers exist.

3. Can you use an always online TTP?
Since SSI is designed to eliminate interference from third parties, the answer to this question is ‘no’.

4. Are all writers known?
In uPort, every reader is also a writer. In other words, it is impossible to confirm who the writer is, and the number of writers will continue to change as new users register. Therefore, the answer to this question is no. It reaches the ‘Permissionless blockchain’ block.

According to the flow chart, uPort must use permissionless blockchain. Because uPort is already using Etherium, which is permissionless blockchain, it can be said that uPort is properly using blockchain system.

4.1.2 Sovrin

1. Do you need to store state?

In the case of Sovrin, the DID and identity schema are stored in the database so that users' credentials can be identified with did. Therefore, the DID must be stored, so the answer to this question is yes.

2. Are there multiple writers?

In Sovrin, writers are Stewards and their certified trust anchors. Therefore, it is yes because multiple writers exist.

3. Can you use an always online TTP?

Like uPort, Sovrin is also SSI, which means an interference of external TTP should be excluded. Therefore, the answer to this question is no.

4. Are all writers known? 5. Are all writers trusted?

In Sovrin, there are two types of writers, Stewards and trust anchors. Stewards are selected by the Sovrin foundation, and trust anchors are also selected organizations or individuals who can prove trustworthiness and accountability through sufficient public evidence. Therefore, all writers can be considered known and trusted and the answer is yes.

It already reached the final block, so question 6 is not needed to be investigated more. It is able to be seen that Sovrin can be implemented without using blockchain. How to implement without blockchain will be proposed in the following Discussion Section.

4.2 Discussion

As mentioned earlier, in the case of Sovrin, it was concluded that the service could be provided without using the blockchain. To support this, we would like to propose a design that can provide the same function without blockchain.

As mentioned in the previous section 7.3, public DID and related DID documents, schemas and credential definitions are stored in the blockchain of Sovrin. If blockchain is not used, a centralized database must be provided to replace blockchain, and any user data and credentials should not be stored in the database. Moreover, ensure that only trust anchors and stewards have write access to this centralized database.

Each user will store their credential, signing key sets, and data exchange records in a storage in a personal device, such as wallets in the mobile application.

Since only the prover and verifier should participate in identity issuance or verification without the intervention of a third party, all processes should proceed in a peer-to-peer manner even if proceeding without blockchain. Still, for user verification through DID, you must use only the key pairs of the verifier and the provider so that others cannot peek.

Additionally, among the ten principles mentioned above, all operating methods and algorithms should be disclosed as open source due to transparency.

For example, suppose that corporation B claims Bob to have a degree of A university. A university and B corporation have already been designated as trusted anchors by stewards, and each of them would have uploaded the schema in which the necessary information they require are described. For example, a schema of A university will store name, degree, status, year, BSN, and score, and it of B corporation will store name, salary, employee status, experience, etc. Bob would have been provided with a credential with his name, degree, graduation status, year, credit, BSN, offered from A university and stored it in his wallet. Since university A does not store Bob's credential, corporation B cannot request the proof of Bob's degree from university A. Therefore, it claims the identity directly to Bob, and as proof of this, Bob authenticates the identity by showing the credential received from A university.

When Bob connects with a university or corporation, he will use a different DID with each organization, respectively. That is, the number of DIDs stored in Bob's wallet will be the same as the number of organizations that Bob communicated with, and when they communicate, a secure channel for peer-to-peer communication will be created and proceeded. In each channel, peers use own verification key sets to prevent hacking, and only the verification key that matches the user's DID is stored in the blockchain, so any private information, credentials, and even name of the user are not recorded.

If it is replaced with a central database, likewise B corporation and A university have already been designated as trusted writers, and they would have stored their schema in the database. When they write something on the central database, they have to attach a digital signature so that the others cannot modify it. The claim and verification process should remain the same as before, and each organization does not store any credentials in the central database.

When a user and an entity communicate, like the blockchain, a secure channel where only they can communicate should be created from the server. When this secure channel is created, a user's unique DID and a verification key set are created. As before, only the user's DID and verification key are stored in the central database, and no other credentials will be stored.

The database type has been changed from blockchain to centralized database, and all operating principles will remain the same. One thing to note is that, unlike distributed ledger, since it has been converted to a centralized database, it is very fatal if the original data is deleted by an attacker. In a distributed ledger, even if one ledger is attacked, it is not changed in others, so it is easy to find out and recover. Therefore, it is necessary to store backup files in several storages to be prepared from the attack in centralized database case.

In this way, when using a centralized database instead of blockchain, the basic design and operation principle are all kept together. Also, as indicated in Table 1, high performance can be expected through higher throughput and lower latency due to the characteristics of the centralized database. However, it can be vulnerable to the above-mentioned security problem, and because of the nature of the identity system, write actions will not generate as tremendous transactions as in the banking system, so it should be taken account to choose a more important value by comparing the trade-off between performance and security.

5 CONCLUSION

Through this research, I investigated the characteristics of SSI and blockchain, and evaluated two different SSIs using Wüster and Gevais's framework to confirm the necessity of blockchain. As a result, it was confirmed that permissionless blockchain must be used in uPort, and in Sovrin, it can be implemented without blockchain, but most of the design must be maintained and the security that can be obtained through blockchain is lost. In other words, it is possible to create without a blockchain, but it has been concluded that it would be better to use a blockchain because the benefits obtained through the blockchain are larger. The limitation was that only two SSIs could be investigated due to time reasons. In the future, this research could be supplemented by analyzing more SSIs and simply implementing the idea suggested in the discussion section, you will be able to check more problems that may occur in the design.

REFERENCE

[1] Allen, Christopher. "The Path to Self-Sovereign Identity." *Lifewithalacrity.com*, 26 Apr. 2016, www.lifewithalacrity.com/2016/04/the-path-toself-sovereign-identity.html.

[2] Fedrecheski, Geovane, et al. "Self-Sovereign Identity for IoT Environments: A Perspective." *IEEE Xplore*, 1 June 2020, ieeexplore.ieee.org/abstract/document/9119664/citations#citations. Accessed 8 May 2022.

[3] Ferdous, Md Sadek, et al. "In Search of Self-Sovereign Identity Leveraging Blockchain Technology." *IEEE Access*, vol. 7, 2019, pp.103059–103079, 10.1109/access.2019.2931173.Accessed 13 Nov. 2019.

[4] Haddouti, Samia El, and M. Dafir Ech-Cherif El Kettani. "Analysis of Identity Management Systems Using Blockchain Technology." *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, Apr. 2019, 10.1109/commnet.2019.8742375. Accessed 10 May 2021.

[5] Jelle, Nauta, and Joosten Rieks. *Self-Sovereign Identity: A Comparison of IRMA and Sovrin*. 2019, 10.13140/RG.2.2.19755.18721. Accessed 8 May 2022.

[6] Loffreto, Devon. *What Is "Sovereign Source Authority"?* 2012, www.moxytongue.com/2012/02/what-issovereign-source-authority.html. Accessed 8 May 2022.

[7] Lundkvist, Christian, et al. *UPORT: A PLATFORM for SELF-SOVEREIGN IDENTITY*. Feb. 2017.

[8] Panda, S. K., Jena, A. K., Swain, S. K., & Satapathy, S. C. (Eds.). (2021). Blockchain technology : applications and cha llenges (Ser. Intelligent systems reference library, volume 2 03). Springer. https://doi.org/10.1007/978-3-030-69395-4

[9] Tobin, Andrew, and Reed. *The Inevitable Rise of Self-Sovereign Identity a White Paper from the Sovrin Foundation*. 2017.

[10] Wüst, Karl, and Arthur Gervais. "Do You Need a Blockchain?" *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, June 2018, ieeexplore.ieee.org/abstract/document/8525392, 10.1109/cvcbt.2018.00